Deliverable Number: 2.10

Deliverable Title: Stata program to compute calibrated survey weights

Work Package: 2 - Representing the population

Deliverable type: Other

Dissemination status: Public

Submitted by: SHARE ERIC

Authors: Giuseppe De Luca, Claudio Rossetti.

Date Submitted: March 2018

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 654221.
SERISS (Synergies for Europe’s Research Infrastructures in the Social Sciences) aims to exploit synergies, foster collaboration and develop shared standards between Europe’s social science infrastructures in order to better equip these infrastructures to play a major role in addressing Europe’s grand societal challenges and ensure that European policymaking is built on a solid base of the highest-quality socio-economic evidence. The four year project (2015-19) is a collaboration between the three leading European Research Infrastructures in the social sciences – the European Social Survey (ESS ERIC), the Survey of Health Ageing and Retirement in Europe (SHARE ERIC) and the Consortium of European Social Science Data Archives (CESSDA AS) – and organisations representing the Generations and Gender Programme (GGP), European Values Study (EVS) and the WageIndicator Survey.

Work focuses on three key areas: Addressing key challenges for cross-national data collection, breaking down barriers between social science infrastructures and embracing the future of the social sciences.

Please cite this deliverable as: De Luca, G and Rossetti C (2018) Stata program to produce calibrated survey weights. Deliverable 2.10 of the SERISS project funded under the European Union’s Horizon 2020 research and innovation programme GA No: 654221. Available at: www.seriss.eu/resources/deliverables
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

March 29, 2018

Abstract
This report describes the Stata programs available to create calibrated weights from scientific usefile based on data from the first wave of Survey on Health, Ageing and Retirement in Europe (SHARE). Since the basic units of analysis can be either individuals or households, we illustrate the computation of both calibrated cross-sectional individual weights for inference to the target population of individuals and calibrated cross-sectional household weights for inference to the target population of households.

Contents
1 Introduction .................................................. 2
2 Cross-sectional SHARE data .................................. 2
   2.1 Individual data .................................................. 3
   2.2 Household data .................................................. 4
3 Calibration margins ........................................... 6
4 Calibrated cross-sectional individual weights ..................... 9
5 Calibrated cross-sectional household weights ..................... 18
6 Conclusions .................................................. 23
References .................................................. 23
1 Introduction

In this deliverable of the Synergies for Europe’s Research Infrastructures in the Social Sciences (SERISS) project, Work Package 2 (Representing the population), Task 2.3 (Weighting for complex survey design), we illustrate the computation of calibrated cross-section weights using the \texttt{sreweight} Stata command implemented by Pacífico (2014). Our examples are based on data from the first wave of the Survey on Health, Ageing and Retirement in Europe (SHARE), where calibration is used to compensate for problems of unit nonresponse in the baseline sample. Since the basic units of analysis can be either individuals or households, these 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 both cases, 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. As discussed in the SERISS deliverable 2.9, calibration margins for the target populations investigated by SHARE are taken from the regional demographic statistics given by Eurostat. 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.

The remainder of the report is organized as follows. Section 2 illustrates how to extract individual and household level SHARE databases for cross-sectional and longitudinal studies, while Section 3 describes the Stata code for constructing the underlying vectors of calibration margins. Sections 4 and 5 show, respectively, the computation of calibrated cross-section individual and household weights. Finally, Section 6 offers some conclusions.

2 Cross-sectional SHARE data

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). The target population 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. This section shows how to extract cross-sectional samples of indi-
individuals and households for representing these alternative variants of the SHARE target population.

2.1 Individual data

The following Stata code allows to extract the sample of individuals interviewed in the first wave of SHARE. Information on accessing the SHARE data can be found via the SHARE website (www.share-project.org).

```
. * Extract individual data from SHARE wave 1
. *---------------------------------------------------------------------------
. local SHARE_w1 "C:\DATA\SHARE\sharew1\Release 6.0.0"
. *
. qui use "SHARE_w1\sharew1_relb-0-0_cv_r", clear
. keep hhid1 country gender yrbirth interview
. qui gen age_w1=yrbirth if yrbirth
. qui if interview==1
. drop interview
. qui merge mergeid using "SHARE_w1\sharew1_relb-0-0_gv_weights", ///
> keep(dw_w1 cciw_w1) sort
. assert _merge==3
. drop _merge
. qui merge mergeid using "SHARE_w1\sharew1_relb-0-0_gv_housing", ///
> keep(nuts3_2003) sort
. assert _merge==3
. drop _merge
. sort mergeid
. qui saveold mydata_w1_ind, replace
.
. tab country

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Austria</td>
<td>1,569</td>
<td>5.14</td>
<td>5.14</td>
</tr>
<tr>
<td>Germany</td>
<td>2,997</td>
<td>9.65</td>
<td>15.00</td>
</tr>
<tr>
<td>Sweden</td>
<td>3,049</td>
<td>10.02</td>
<td>25.02</td>
</tr>
<tr>
<td>Netherlands</td>
<td>2,948</td>
<td>9.75</td>
<td>34.77</td>
</tr>
<tr>
<td>Spain</td>
<td>2,316</td>
<td>7.61</td>
<td>42.38</td>
</tr>
<tr>
<td>Italy</td>
<td>2,653</td>
<td>8.69</td>
<td>50.77</td>
</tr>
<tr>
<td>France</td>
<td>3,122</td>
<td>10.24</td>
<td>60.60</td>
</tr>
<tr>
<td>Denmark</td>
<td>1,706</td>
<td>5.41</td>
<td>66.04</td>
</tr>
<tr>
<td>Greece</td>
<td>2,877</td>
<td>9.52</td>
<td>75.15</td>
</tr>
<tr>
<td>Switzerland</td>
<td>997</td>
<td>3.28</td>
<td>79.43</td>
</tr>
<tr>
<td>Belgium</td>
<td>3,610</td>
<td>12.52</td>
<td>91.95</td>
</tr>
<tr>
<td>Israel</td>
<td>2,450</td>
<td>8.05</td>
<td>100.00</td>
</tr>
</tbody>
</table>

| Total              | 30,434| 100.00  |        |
```

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 4, we shall illustrate how to reproduce the calibrated weights cciw_w1.

2.2 Household data

In addition to individual level information, SHARE collects household level data about consumption, income and wealth. For this type of variables, we are often interested in constructing a sample of households. Our Stata code is similar to the one for extracting the cross-sectional sample of individuals, 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 b.0.0"
. qui use "SHARE_w1\sharew1_re1b-0-0_data", clear
. keep merge hhid1 country gender yrbirth interview
. qui gen age_w1=2004-yrbirth if yrbirth>0
. sort merge
. bys hhid1: gen member=_n
. drop merge
. 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)

<table>
<thead>
<tr>
<th>Data</th>
<th>long</th>
<th>wide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of obs.</td>
<td>43993</td>
<td>20809</td>
</tr>
<tr>
<td>Number of variables</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>j variable (10 values)</td>
<td>member -&gt; (dropped)</td>
<td></td>
</tr>
<tr>
<td>xij variables:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gender_ -&gt; gender_1 gender_2 ... gender_10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>yrbirth_ -&gt; yrbirth_1 yrbirth_2 ... yrbirth_10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>age_w1_ -&gt; age_w1_1 age_w1_2 ... age_w1_10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>interview_ -&gt; interview_1 interview_2 ... interview_10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

. qui compress
. sort hhid1
. qui merge hhid1 using "SHARE_w1\sharew1_re1b-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_re1b-0-0_gv_housing", ///
   > keep(nuts3_2003)
   > assert _merge==3
. qui bys hhid1: keep if _n==1
. drop _merge
. sort hhid1
. qui saveold mydata_w1_hhs, replace
. tab country

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Austria</td>
<td>1,173</td>
<td>5.64</td>
<td>5.64</td>
</tr>
<tr>
<td>Germany</td>
<td>1,993</td>
<td>9.58</td>
<td>15.23</td>
</tr>
<tr>
<td>Sweden</td>
<td>2,137</td>
<td>10.27</td>
<td>25.48</td>
</tr>
<tr>
<td>Netherlands</td>
<td>1,746</td>
<td>9.35</td>
<td>34.84</td>
</tr>
<tr>
<td>Spain</td>
<td>1,646</td>
<td>8.10</td>
<td>42.94</td>
</tr>
</tbody>
</table>
```


The database `mydata_w1_hhs.dta` 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 5.

### 3 Calibration margins

This section shows how to construct the vector of calibration margins used to compute calibrated weights. We use 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 marginalsnuts1 is provided in Rossetti (2017).

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 to either the population in a given year (used for cross-sectional weights), or the 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. For cross-sectional weights, the macro mort_time (final year) macro is set to zero. The macro w defines simply a label for the wave of interest.

``` stata
. * 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 marginals_suits1.dta
* #-----------------------------------------------------------------------
* qui use "marginals_suits1", 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_TH
- 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 a=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_TH=0nullmat(`cc'w'w'P_AGE_TH)\(`age_low','`age_upp')
    17. local t=`t'+1
    18. }
    19. }
- matrix col `cc'w'w'P_SA =POP
- matrix row `cc'w'w'P_SA ="rname"
- matrix col `cc'w'w'P =POP
- matrix row `cc'w'w'P =TOT
- matrix col `cc'w'w'P_AGE_TH =`age_thr_low age_thr_upp"
- matrix list `cc'w'w'P
- matrix list `cc'w'w'P_SA

* # NUTS 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/`nreg' {  
    2. local nn_lab: label REG "nn''
    3. qui sum pop if year==`pop_time' ///
    & sex==2 ///
    & nuts==`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') ///
    & nuts=="`nn_lab"

8
4 Calibrated cross-sectional individual weights

In this section we 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'           //
```
As described in the previous section, this do-file creates one scalar and three vectors in the form of
Stata matrices. The scalar $DE_{wl}P$ contains the German target population aged 50+ at the time of the wave 1 interview, while the vectors $DE_{wl}P-SA$ and $DE_{wl}P-NUTS1$ 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 $DE_{wl}P-SA$ and the vector obtained by excluding the first component of $DE_{wl}P-NUTS1$ yields the vector of calibration margins $DE_{wl}P-MARG$ 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_wl_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 wl and dw wl 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 wl
<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>age wl</td>
<td>2,996</td>
<td>63.95628</td>
<td>9.749357</td>
<td>30</td>
<td>97</td>
</tr>
<tr>
<td>gender</td>
<td>2,997</td>
<td>1.541875</td>
<td>.4980265</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>dw wl</td>
<td>2,996</td>
<td>5013.145</td>
<td>1819.142</td>
<td>1967.101</td>
<td>9525.167</td>
</tr>
</tbody>
</table>
In the following code we define a set of binary indicators for our calibration variables. More precisely, we generate the binary indicators $x_{i1} \ldots x_{i8}$ for the 8 gender-age groups and the binary indicators $x_{i9} \ldots x_{i23}$ for the 15 NUTS1 regional areas. This list of these indicators is stored in the local macro list Cvar.
At this stage of the procedure we have all necessary ingredients for reproducing the SHARE calibrated weights. This can be done by the svrweight command. In addition to the list of calibration variables, this command requires to specify three arguments: the option neweight 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.
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(1m)`. Notice that, since the original design weights `dw_wl` 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.

Next, we compare our calibrated weights `my_wgt` with both the original design weights `dw_wl` and the calibrated weights `c wt_wl` available in the release 6.0.0 of the SHARE data.
. gen my_wgt_f=(my_wgt==1)
. bysort hhid1: gen double hh=(_n==1)
. table my_wgt_f, c(count hh) sum(my_wgt) row format(2%0f)

<table>
<thead>
<tr>
<th>my_wgt_f</th>
<th>N(hh)</th>
<th>sum(my_wgt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2928</td>
<td>30274231</td>
</tr>
<tr>
<td>1</td>
<td>69</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>2997</td>
<td>30274231</td>
</tr>
</tbody>
</table>

. compare my_wgt cciw_w1

<table>
<thead>
<tr>
<th>my_wgt=cciw_w1</th>
<th>count</th>
<th>minimum</th>
<th>average</th>
<th>maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>jointly defined</td>
<td>2928</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>jointly missing</td>
<td>69</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>2997</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

. 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 wt" ///
  >     2 "calibrated wt" ///
  >   ) ///
  >   row(2) col(3) symsize(5) rowg(*.4) ///
  >   region(lc(white)) position(1) ring(0) ///
  > ) ///

. twoway
  > (kdensity r_wgt, lc(red) lp(-)) ///
  > , ///
  > ytitle(density) xtitle(cal wgt / des wgt) ///
  > xlabel(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 $l = \{0.01, .25, .50, .75, .90\}$ and $u = \{3.5, 4.0, 4.5, 5.0, 5.5\}$.

```plaintext
# 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_wl) ///
        > 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.  }
  10.  local wgt_list "wgt_list' my_wgt'_\$'ll_num'_\$'uu_num'
  11.  qui gen double r_wgt_\$'ll_num'_\$'uu_num'=my_wgt'_\$'ll_num'_\$'uu_num'/dw_wl
  12.  local r_wgt_list "r_wgt_list' r_wgt'_\$'ll_num'_\$'uu_num'
  13.  }
  14.  local uu_num='uu_num'+1
  15. }
* local ll_num=\$'ll_num'+1
...}
```
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).

```stata
local fig_wgt "
  foreach wgt of local wgt_list {
    local fig_wgt "'fig_wgt' (kdensity 'wgt')"
  }
  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 {
    local fig_r_wgt "'fig_r_wgt' (kdensity 'r_wgt')"
  }
  twoway 'fig_r_wgt', ytitle(density) xtitle(weights) graphr(c(white))

#---------------------------------------------------------
#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 srwgt "list_CVar" if now!=1. ///
    > nweight(my_wgt_`dd') sweight(dw_w1) ///
    > total(#cc)_#w#(u_P_MARG) ///
    > dfunction('dd') ///
    > niter(200)
    4. if "'dd'"="'chi2'"&r(converged)=="yes" {
      5. di in ye "Convergence : yes"
      6. 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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>my_wgt</td>
<td>2,928</td>
<td>10339.56</td>
<td>4391.609</td>
<td>3125.445</td>
<td>34565.25</td>
</tr>
<tr>
<td>my_wgt_chi2</td>
<td>2,928</td>
<td>10339.56</td>
<td>4392.394</td>
<td>3132.131</td>
<td>34667.13</td>
</tr>
<tr>
<td>my_wgt_c</td>
<td>2,928</td>
<td>10339.56</td>
<td>4397.699</td>
<td>3147.321</td>
<td>34797.94</td>
</tr>
</tbody>
</table>
```

17
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.

5 Calibrated cross-sectional household weights

Consider now the calibrated cross-sectional household weights \( c \text{ch}_w \). 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.
* 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 ageThr_low "00 00 00 00" // lower thresholds of age groups
* global ageThr_upp "09 09 09 09" // upper thresholds of age groups

* Get local macros

* Run CalMar.do

* Number of calibration equations

* Load my SHARE database and select the country-specific sample

* Recode of NUTS regional codes

* Replace region = .
* replace region=0 if nuts1=="DE1"
* replace region=1 if nuts1=="DE2"
* replace region=2 if nuts1=="DE3"
* replace region=3 if nuts1=="DE4"
* replace region=4 if nuts1=="DE5"
* replace region=5 if nuts1=="DE6"
* replace region=6 if nuts1=="DE7"
* replace region=7 if nuts1=="DE8"
* replace region=8 if nuts1=="DE9"
* replace region=9 if nuts1=="DEA"
* replace region=10 if nuts1=="DEB"
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.

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 \( xh_1 \cdot xh_{23} \)) 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 var`.

```stata
qui replace region=11 if nuts1=="DEC"
qui replace region=12 if nuts1=="DEF"
qui replace region=13 if nuts1=="DEF"
qui replace region=14 if nuts1=="DEF"
qui replace region=15 if nuts1=="DEF"
```

```stata
* 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. }
* nowh=(dw_w1==. in_elig_w1==0) noi tab nowh, mis
```
```
* qui mencode CI* - mv(0) o
* forvalues i=1(1)10 {
  2. assert CI1_`i'==0 & CI1_`i'<=CI1
  3. if `C2'>0 assert CI2_`i'==0 | (CI2_`i'<=CI2 & CI2_`i'>=C')
* }
* local list_CVar `m'
* 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'==1') 5.
* }
* local list_CVar 'list_CVar' xh_`i'
* cap drop CI*
* sum xh#
* }
```

<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>xh_1</td>
<td>1.993</td>
<td>0.035</td>
<td>0.175</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>xh_2</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>xh_3</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>xh_4</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>xh_5</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>xh_6</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>xh_7</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>xh_8</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>xh_9</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>xh_10</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>xh_11</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>xh_12</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>xh_13</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>xh_14</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>xh_15</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>xh_16</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>xh_17</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>xh_18</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>xh_19</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>xh_20</td>
<td>1.993</td>
<td>0.000</td>
<td>0.175</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>
```

```
. list hhid1
gender_1 age_wl_1
> gender_2 age_wl_2
> gender_3 age_wl_3
> gender_4 age_wl_4
> xh_1-xh_8
> region xh_9-xh_23

```
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 $\text{my} \_\text{wgt} \_\text{hh}$ coincide exactly with the calibrated cross-sectional household weights $\text{cchw} \_\text{w1}$ available in the release 6.0.0 of the SHARE data.

```
# #==================================================================
# * Compute calibrated weights (distance function: DS - case h)
# #==================================================================
> #weight (`list_CVar` if nwh!=1, ///
> nweight(`my_wgt_hh` sweight(dw_w1) ///
> total(`#cc`) w`(#w) P_MARG) ///
> dfun(den) upbound($) lowbound(.95) ///
> niter(200)
Iteration 1
(output omitted)
Iteration 200
Not Converged within the maximum number of iterations. Try to use the NTRIES option
> matrix start=J(23,1,0)
> #weight (`list_CVar` if nwh!=1, ///
> nweight(`my_wgt_hh` sweight(dw_w1) ///
> total(`#cc`) w`(#w) P_MARG) ///
> dfun(den) upbound($) 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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Original</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>xh_1</td>
<td>239619</td>
<td>94653</td>
</tr>
<tr>
<td>xh_2</td>
<td>1486431</td>
<td>2850711</td>
</tr>
<tr>
<td>xh_3</td>
<td>2802064</td>
<td>5051364</td>
</tr>
<tr>
<td>xh_4</td>
<td>265222</td>
<td>446270</td>
</tr>
<tr>
<td>xh_5</td>
<td>1034436</td>
<td>2301710</td>
</tr>
<tr>
<td>xh_6</td>
<td>1855672</td>
<td>3764017</td>
</tr>
<tr>
<td>xh_7</td>
<td>3031223</td>
<td>5387424</td>
</tr>
<tr>
<td>xh_8</td>
<td>2870268</td>
<td>4975022</td>
</tr>
<tr>
<td>xh_9</td>
<td>2270691</td>
<td>4411850</td>
</tr>
<tr>
<td>xh_10</td>
<td>573515</td>
<td>1207394</td>
</tr>
<tr>
<td>xh_11</td>
<td>49594</td>
<td>96912</td>
</tr>
<tr>
<td>xh_12</td>
<td>14028</td>
<td>257730</td>
</tr>
<tr>
<td>xh_13</td>
<td>338807</td>
<td>620460</td>
</tr>
<tr>
<td>xh_14</td>
<td>1420156</td>
<td>2222725</td>
</tr>
<tr>
<td>xh_15</td>
<td>318948</td>
<td>640323</td>
</tr>
<tr>
<td>xh_16</td>
<td>1503702</td>
<td>2793072</td>
</tr>
<tr>
<td>xh_17</td>
<td>3565874</td>
<td>6584773</td>
</tr>
<tr>
<td>xh_18</td>
<td>750026</td>
<td>1492760</td>
</tr>
<tr>
<td>xh_19</td>
<td>144463</td>
<td>413326</td>
</tr>
<tr>
<td>xh_20</td>
<td>95367</td>
<td>1779362</td>
</tr>
<tr>
<td>xh_21</td>
<td>477429</td>
<td>1018621</td>
</tr>
<tr>
<td>xh_22</td>
<td>740787</td>
<td>1067047</td>
</tr>
<tr>
<td>xh_23</td>
<td>51534</td>
<td>924271</td>
</tr>
</tbody>
</table>
```

Note: type=ds distance function used
Current bounds: upper=5 - lower=0.95
- 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)

<table>
<thead>
<tr>
<th>my_wgt_hh_f</th>
<th>N(hh)</th>
<th>sum(my_wgt_h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1,992</td>
<td>187539116</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>1,993</td>
<td>187539116</td>
</tr>
</tbody>
</table>

- compare my_wgt_hh cchw_w1

<table>
<thead>
<tr>
<th>count</th>
<th>minimum</th>
<th>difference</th>
<th>maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>my_wgt_hh=cchw_w1</td>
<td>1992</td>
<td></td>
<td></td>
</tr>
<tr>
<td>jointly defined</td>
<td>1992</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>jointly missing</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>1993</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6 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. 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 Generations and Gender Programme (GGP).
References


Rossetti, C. (2017) "Database containing necessary information for computation of population margins". Deliverable 2.9 of the SERISS project funded under the European Union’s Horizon 2020 research and innovation programme GA No: 654221. Available at: www.seriss.eu/resources/deliverables

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