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.

www.seriss.eu @SERISS_EU

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

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 `sreweight` Stata command implemented by Pacifico (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 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_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 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 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 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 marginsnuts1is 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.

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

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}
```

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

Next, 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.

5 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)
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,.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
```

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

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

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 admin- istrative 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

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.

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

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

Figure 1: Design and calibrated weights

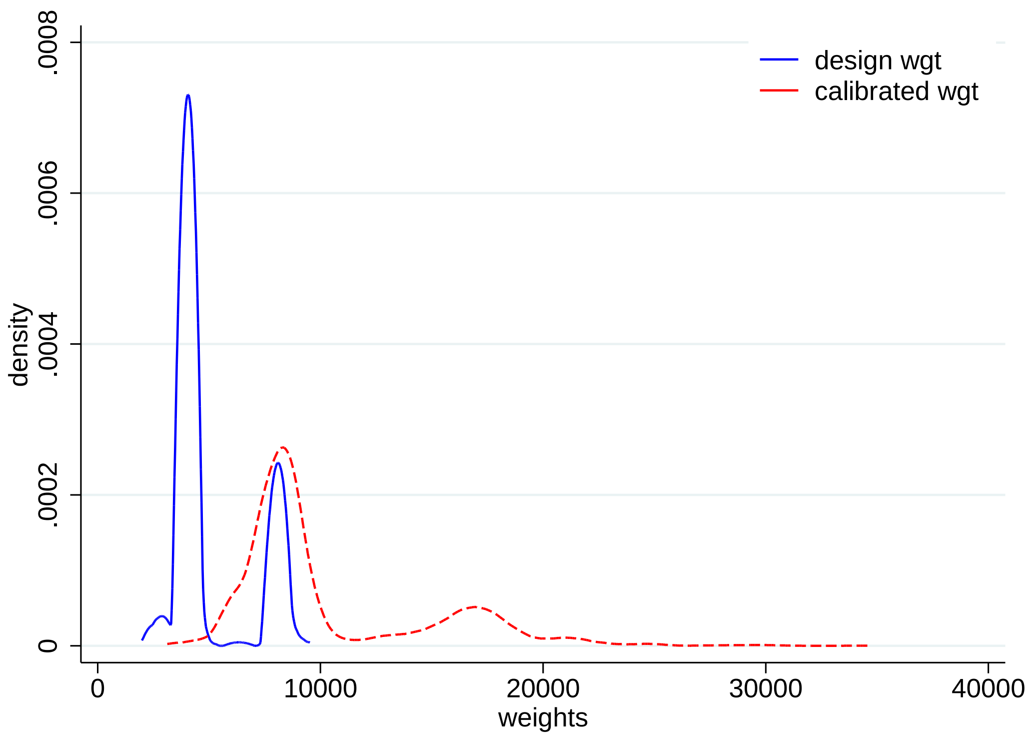

Figure 2: Ratio between calibrated and design weights

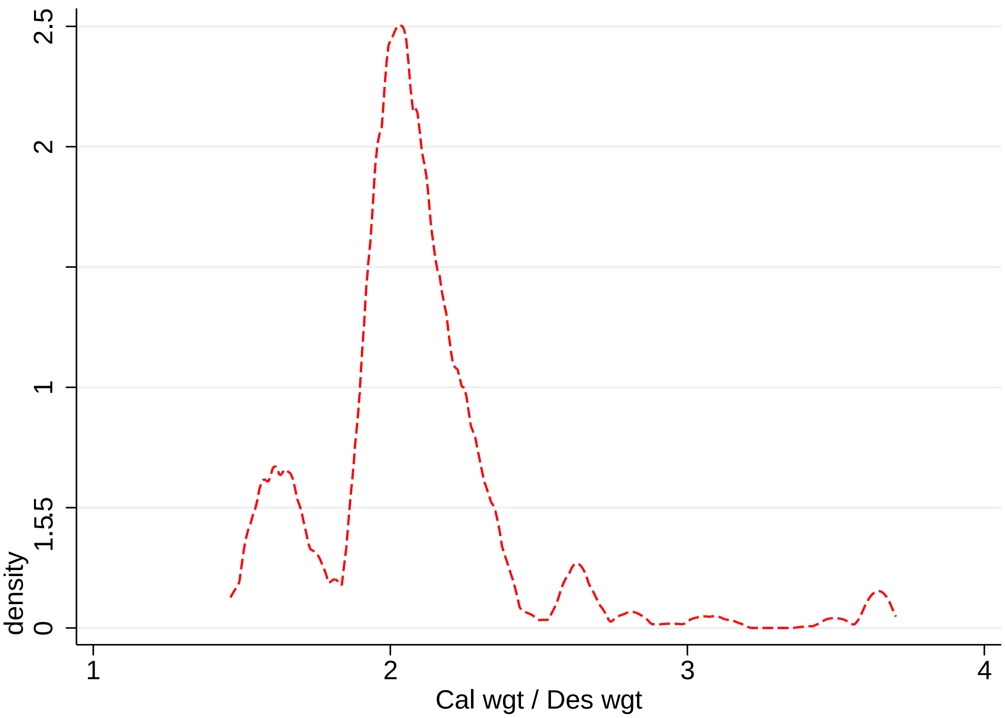

Figure 3: Calibrated weights with alternative bounds in the logit distance function

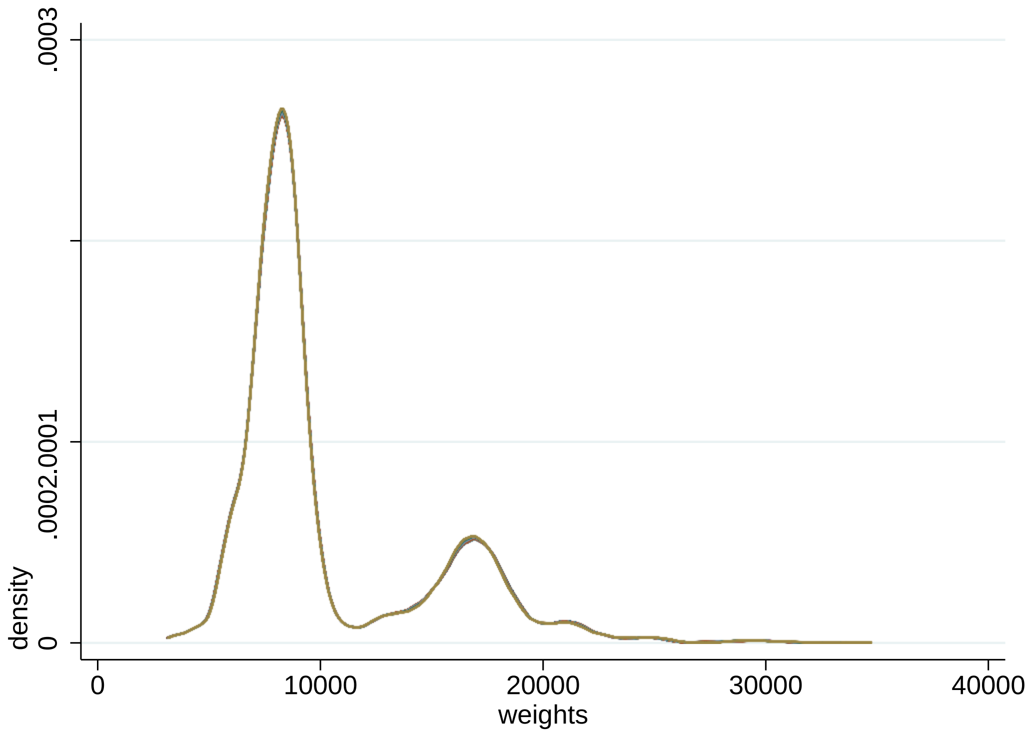

Figure 4: Ratio between calibrated weights with alternative bounds in the logit distance function and design weights

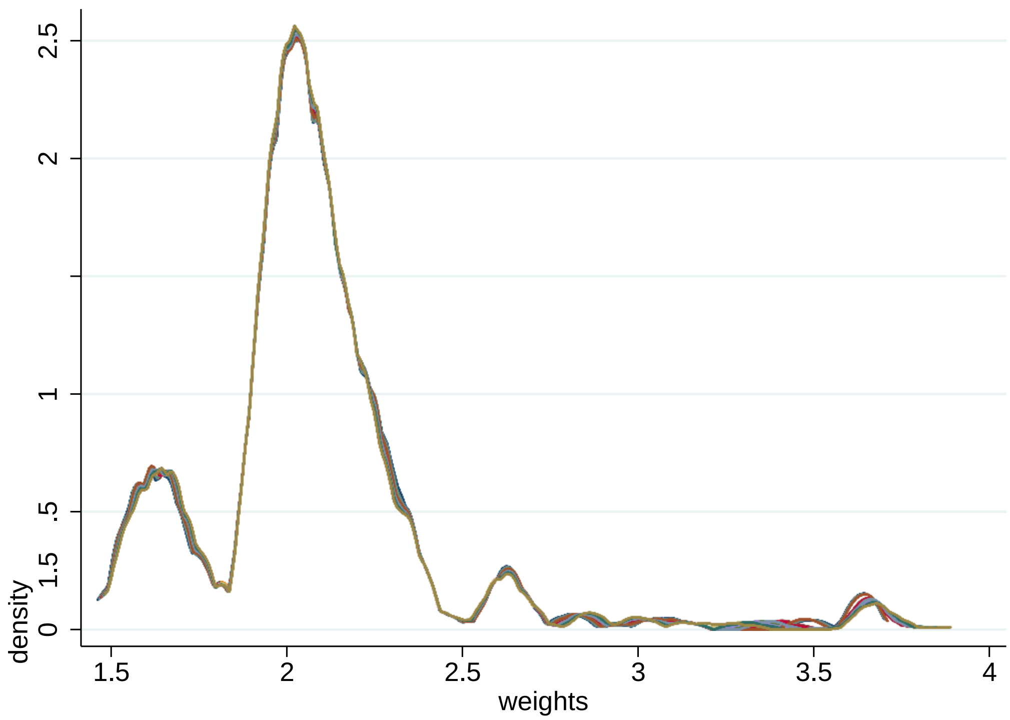

Figure 5: Calibrated weights with alternative specifications of the distance function

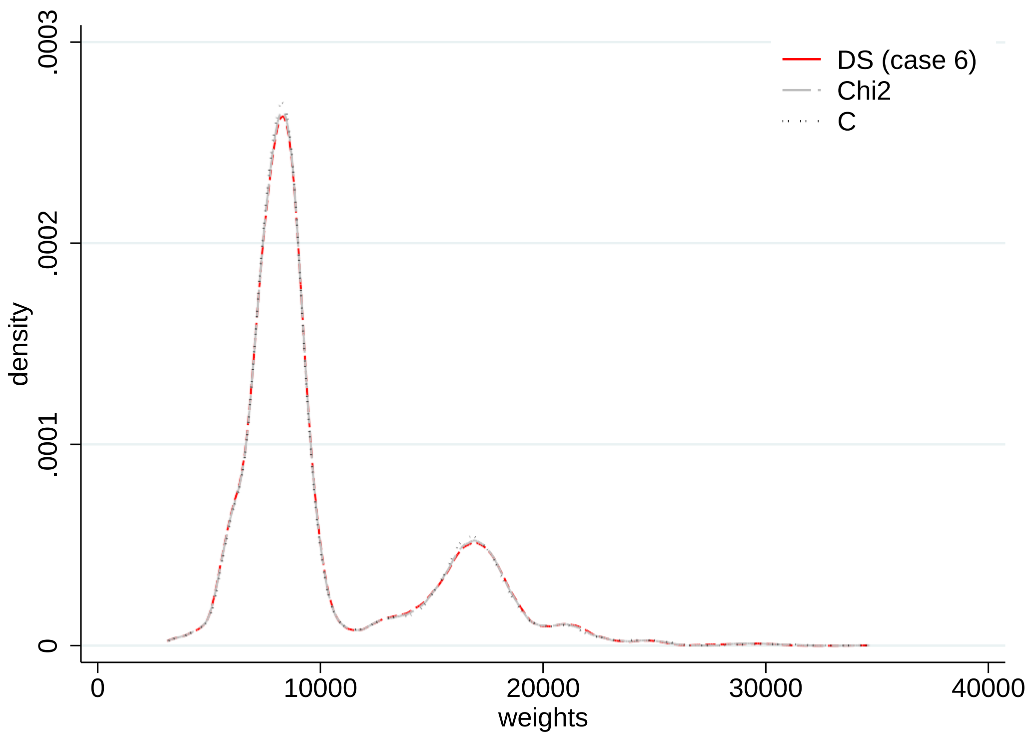

Figure 6: Ratio between calibrated and design weights with alternative specifications of the distance function and design weights