Reweight: a Stata Module to reweight Survey Data to External Totals

Daniele Pacifico
Working Papers

The working paper series promotes the dissemination of economic research produced in the Department of the Treasury (DT) of the Italian Ministry of Economy and Finance (MEF) or presented by external economists on the occasion of seminars organized by MEF on topics of institutional interest to the DT, with the aim of stimulating comments and suggestions.

The views expressed in the working papers are those of the authors and do not necessarily reflect those of the MEF and the DT.

© Copyright:

2014, Daniele Pacifico.

The document can be downloaded from the Website www.dt.tesoro.it and freely used, providing that its source and author(s) are quoted.

Editorial Board: Lorenzo Codogno, Mauro Marè, Libero Monteforte, Francesco Nucci, Franco Peracchi
Organizational coordination: Marina Sabatini
Reweight: a Stata Module to reweight Survey Data to External Totals

Daniele Pacifico (*)

Abstract

This paper describes reweight, a Stata module to reweight survey data to external aggregate totals.

JEL: C83, C87, C63, C15.

Keywords: st0001, reweight, survey reweighting, external totals, calibration of survey data.

1 Introduction

In this contribution we overview the reweight package for the calibration of survey data to external, user-provided totals. The methodology closely follows Deville and Särndal (1992) and the recursive algorithm that implements the calibration is from Creedy (2003).

Section 2 contains the theoretical background for survey reweighting; section 3 shows how to use the reweight package and section 4 contains an empirical application based on accessible data.

2 An overview of the calibration method

Let us consider a survey of $N$ individuals and $K$ individual-level variables, such as income, sex, age, etc. We can collect these variables for the generic agent $i$ in the following vector: $x_i = [x_{i1}, x_{i2}, \ldots, x_{iK}]$. If we define the survey weight with the vector $s = [s_1, s_2, \ldots, s_i, \ldots, s_N]$ the estimated $1 \times K$ vector of totals is given by:

$$ t = \sum_{i=1}^{N} s_i x_i $$

(1)

If external information are available about the real population totals for these $K$ variables it is possible to compute a new vector of survey weights, $w = [w_1, w_2, \ldots, w_i, \ldots, w_N]$ that is as close as possible to the original one and that respect the conditions:

$$ t = \sum_{i=1}^{N} w_i x_i $$

(2)

(*) - Ministry of the Economy and Finance, Department of the Treasury. Via XX Settembre, 97, 00187 Rome, Italy.
- Centre for the Analysis of Public Policies, Department of Economics, University of Modena and Reggio Emilia, Viale Berengario, 51 41121 Modena, Italy. E-mail: daniele.pacifico@tesoro.it

1 Please, type findit reweight in the command bar and download the reweight package from the BoCo archive.
where \( t \) is the \( 1 \times K \) vector of true totals. Indeed, if we denote the distance between the original and the new weights with the function \( G(s_i, w_i) \), the new weights can be obtained by minimizing the following Lagrangian function with respect to \( w \):

\[
L = \sum_{i=1}^{N} G(s_i, w_i) + \sum_{k=1}^{K} \lambda_k \left[ t_k - \sum_{i=1}^{N} w_i x_{ik} \right]
\]  

where \( \lambda = [\lambda_1, \lambda_2, \ldots, \lambda_K]' \) are the Lagrange multipliers. Clearly, the solution of the minimization problem strongly depends on the property of the distant function \( G(s_i, w_i) \). In the following we require the function \( G(s_i, w_i) \) to respect the following properties:

1. The first derivative of \( G(s_i, w_i) \) with respect to \( w_i \) must be expressed as a function of the ratio between the new and the original weights:

\[
\frac{\partial G(s_i, w_i)}{\partial w_i} = g \left( \frac{w_i}{s_i} \right)
\]  

2. The inverse of the first derivative of \( G(s_i, w_i) \) must be obtained explicitely.

If these properties hold, then the \( N \) first order conditions for the problem in 3 are:

\[
g \left( \frac{w_i}{s_i} \right) - x_i' \lambda = 0 \quad i = 1, 2, ..., N
\]  

and the new weights can be obtained as:

\[
w_i = s_i g^{-1}(x_i' \lambda) \quad i = 1, 2, ..., N
\]  

given a solution for the Lagrange multipliers.

The Lagrange multipliers can be obtained through an iterative procedure after some algebraic manipulation of equation 6. In particular, let us substitute 6 into 2:

\[
t = \sum_{i=1}^{N} w_i x_i = \sum_{i=1}^{N} s_i g^{-1}(x_i' \lambda) x_i
\]  

and then subtract from both sides equation 4:

\[
t - \hat{t} = \sum_{i=1}^{N} s_i g^{-1}(x_i' \lambda) x_i - \sum_{i=1}^{N} s_i x_i = \sum_{i=1}^{N} s_i \left[ g^{-1}(x_i' \lambda) - 1 \right] x_i
\]  

If we define \( a = t - \hat{t} \) equation 8 can be rewritten as:

\[
f(\lambda) = a - \sum_{i=1}^{N} s_i \left[ g^{-1}(x_i' \lambda) - 1 \right] x_i = 0
\]
The root of this function can be computed by means of the Newton’s method, which involves the following iterative algorithm:

\[\lambda^{[t+1]} = \lambda^{[t]} - \left[\frac{\partial f(\lambda)}{\partial \lambda}\right]^{-1} f(\lambda)\]  

(10)

where \(f(\lambda)\) is evaluated with the previous value of the Lagrange multipliers, \(\lambda^{[t]}\). The generic element of the Hessian matrix that enter the previous recursion - \(\frac{\partial f(\lambda)}{\partial \lambda}\) - can be computed from equation 9:

\[\frac{\partial f_s(\lambda)}{\partial \lambda_k} = -\sum_{i=1}^{N} s_i x_{is} \frac{\partial g^{-1}(x'_i \lambda)}{\partial \lambda_k} = -\sum_{i=1}^{N} s_i x_{is} x_{ik} \frac{\partial g^{-1}(x'_i \lambda)}{\partial (x'_i \lambda)}\]  

(11)

Given a set of initial values for \(\lambda\) the recursion in 10 can be repeatedly evaluated until convergence. The following paragraph considers some of the most common distance functions, providing the related equations for the empirical implementation of the recursive algorithm explained in this section.

2.1 The chi-squared distant function

The chi-squared distant function is one of the most popular choice in the applied literature. The main reason is that the minimization problem in equation 3 has an explicit solution that can be obtained immediately without the iterative procedure overviewed in section 2. The chi-squared aggregate distant function is given by:

\[G(s, w) = \frac{1}{2} \sum_{i=1}^{N} \left(\frac{w_i - s_i}{s_i}\right)^2\]  

Hence, substituting this function into 3 we obtain:

\[L = \frac{1}{2} \sum_{i=1}^{N} \left(\frac{w_i - s_i}{s_i}\right)^2 + \sum_{k=1}^{K} \lambda_k \left[t_k - \sum_{i=1}^{N} w_i x_{ik}\right]\]  

(13)

Therefore, the \(N\) first order conditions are:

\[\frac{\partial L}{\partial w_i} = \left(\frac{w_i}{s_i} - 1\right) - \sum_{k=1}^{K} \lambda_k x_{ik} = 0\]

\[\left(\frac{w_i}{s_i} - 1\right) - x'_i \lambda = 0\]  

\[s_i \left(1 + x'_i \lambda\right) = w_i\]  

(14)

2. Clearly, the method is limited to distant functions for which \(g^{-1}(x'_i \lambda)\) Quick convergence is ensured by the use of the closed-form solution of \(\frac{\partial g^{-1}(x'_i \lambda)}{\partial (x'_i \lambda)}\), during the recursion.
for $i = 1, 2, \ldots, N$. In order to obtain the Lagrangian multipliers we pre-multiply 14 by $x_i$, rearrange and sum over all $N$:

$$
\sum_{i=1}^{N} s_i x_i' \lambda = \sum_{i=1}^{N} w_i x_i - \sum_{i=1}^{N} s_i x_i
$$

(15)

Then, given the conditions in 1 and 2, equation 15 can be rewritten as:

$$
\left[ \sum_{i=1}^{N} s_i x_i' \right] \lambda = t - \hat{t}
\lambda = \left( \sum_{i=1}^{N} s_i x_i' \right)^{-1} (t - \hat{t})
$$

(16)

Finally, substituting 16 into 14 gives the new weights for the $N$ observations.

### 2.2 Alternative distant functions

The main limitation of the chi-squared distant function is that no constraints are placed on the size of the adjustment to each of the survey weights, hence it could happen that some of the calibrated weights become negative after the adjustment.

In order to avoid this, alternative distant functions that incorporate constraints on the size of the adjustment have been proposed in the literature. However, for these function a closed-form solution is no longer available and the iterative procedure explained in section 2 ought to be used. In what follows we report three additional distant functions coded into reweight that force the new weight to be strictly positive. We also report the equations needed to update the recursion during the iterative procedure, that is the functions $g(x_i', \lambda)$, $g^{-1}(x_i', \lambda)$ and $\frac{\partial g^{-1}(x_i', \lambda)}{\partial (x_i', \lambda)}$.

<table>
<thead>
<tr>
<th>A</th>
<th>$G(s_i, w_i)$</th>
<th>$g\left(\frac{w_i}{s_i}\right)$</th>
<th>$g^{-1}(x_i', \lambda)$</th>
<th>$\frac{\partial g^{-1}(x_i', \lambda)}{\partial (x_i', \lambda)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$2 \left( \sqrt{w_i} - \sqrt{s_i} \right)^2$</td>
<td>$2 \left( 1 - \left( \frac{w_i}{s_i} \right)^{-\frac{1}{2}} \right)^2$</td>
<td>$1 - x_i'^2 - \frac{x_i'^2}{2}$</td>
<td>$1 - \frac{x_i'^2}{2}$</td>
</tr>
<tr>
<td>B</td>
<td>$-s_i \ln \left( \frac{w_i}{s_i} \right) + w_i - s_i$</td>
<td>$1 - \left( \frac{w_i}{s_i} \right)^{-1}$</td>
<td>$1 - x_i' \lambda$</td>
<td>$(1 - x_i' \lambda)^{-2}$</td>
</tr>
<tr>
<td>C</td>
<td>$-w_i \ln \left( \frac{w_i}{s_i} \right) - w_i + s_i$</td>
<td>$\ln \left( \frac{w_i}{s_i} \right)$</td>
<td>$\exp \left( x_i' \lambda \right)$</td>
<td>$\exp \left( x_i' \lambda \right)$</td>
</tr>
</tbody>
</table>

**The DS distant function**

The reweight package allows also for the following distant function:

$$
G(s_i, w_i) = \left( r_U - \frac{w_i}{s_i} \right) \log \left( \frac{r_U - \frac{w_i}{s_i}}{r_U - 1} \right) + \left( \frac{w_i}{s_i} - r_U \right) \log \left( \frac{\frac{w_i}{s_i} - r_L}{1 - r_L} \right) + \frac{r_U - r_L}{\alpha} s_i
$$

(17)
where \( r_L < 1 < r_U \) are known positive constants and \( \alpha = \frac{r_U - r_L}{(1-r_L)(r_U-1)} \). The function above was suggested by Deville and Sarndal (1992) because differently from the other distant functions - it has the important property that the calibrated weights are now kept within a known range that can be set by the user, that is: \( r_L s_i < w_i < r_U s_i \).

The related elements of \( G(s_i, w_i) \) needed to update the iterative recursion are:

\[
g(w_i s_i) = \frac{1}{\alpha} \left[ \log \left( \frac{w_i s_i - r_L}{1 - r_L} \right) - \log \left( \frac{r_U - w_i s_i}{r_U - 1} \right) \right] \tag{18}
\]

\[
g^{-1}(x'_i \lambda) = \frac{r_L(r_U - 1) + r_U(1 - r_L)\exp(\alpha x'_i \lambda)}{(r_U - 1) + (1 - r_L)\exp(\alpha x'_i \lambda)} \tag{19}
\]

\[
\frac{\partial g^{-1}(x'_i \lambda)}{\partial(x'_i \lambda)} = g^{-1}(x'_i \lambda) \left( r_U - g^{-1}(x'_i \lambda) \right) \frac{(1 - r_L)\exp(\alpha x'_i \lambda)}{(r_U - 1) + (1 - r_L)\exp(\alpha x'_i \lambda)} \tag{20}
\]

3 The reweight command

The generic syntax for reweight is:

\[ \text{reweight varlist [if] [in], sweight(varname) neweight(string) total(matrix) dfunction(string) [options]} \]

Where varlist includes the calibration variables. The options for reweight are:

- **sweight(varname)** is required and specifies a numeric variable for the original survey weights.
- **newweight(varname)** is required and contains the name of the new weights computed by reweight.
- **total(matrix)** is required and contains a Stata \( 1 \times K \) matrix with the user-provided totals, with arguments to be inserted in the same order as the variables in varlist.
- **dfunction(varname)** is required and specifies the type of distant function to be used. The allowed functions are those introduced in section 2 i.e. the chi-squared, the type-A, the type-B, the type-C and the DS distant functions. Note that for all functions but the chi-squared reweight works with the Newton method outlined in the previous section.
- **svalues(varname)** contains a Stata \( 1 \times K \) matrix with user-provided starting values. The default starting values are the Lagrange multipliers obtained from the chi-squared distant function.
- **tolerance(#)** specifies the tolerance level for the iterative recursion. The default is 0.000001. reweight employs a double criterion to assess convergence. The first
is that the difference between the estimated and external totals must be lower than the tolerance level. The second criterion is that, from one iteration to the other, the percentage variations of the estimated distance between the new and the original weights must be lower than the tolerance level for each observation in the sample.

- **niter(#)** specifies the maximum number of iterations. The default is 50. Note that this option interacts with the option **ntries** explained herein below.

- **ntries(#)** specifies the maximum number tries when the algorithm does not achieve convergence within the maximum number of iterations. In such situations the algorithm automatically restarts with new random starting values up to #times. The default is **ntries(0)**. This option can be useful when the external totals are quite different from the survey totals. In this case different starting values may play an important role in ensuring convergence.

- **lowbound(#)** specifies the lower-bound of the ratio between the new and the original weight when using the DS distance function. The default is **lowbounds(0.2)**. Note that this value must be between 0 and 1.

- **upbound(#)** specifies the upper-bound of the ratio between the new and the original weight when using the DS distance function. The default is **upbounds(3)**. Note that this value must be bigger than 1.

- **mlowbound(#) and mupbound(#)** are relevant options only for the DS distant function when the option **ntries(#)** is effective. In this case, if the recursion does not achieve convergence the routine starts again with a new set of starting values and of new random bounds. **mlowbound(#)** specifies the maximum deviation from the highest value of the lower bound and **mupbound(#)** specifies the maximum deviation from the lowest value of the upper bound. The default is 0.1 for the lower bound and 6 for the upper bound. As an example, if mlowbound(#) is set to 0.5 than the new random value for the lower bound will be drawn from a uniform distribution in the range 0.5-1 and if mupbound(#) is set to 5 than the new random value for the upper bound will be drawn from a uniform distribution in the range 1-5.

4 Empirical application

In this section we show how to use **reweight** with the same hypothetical data used in [Creedy (2003)](http://fmwww.bc.edu/RePEc/bocode/r/reweight). Specifically, the dataset is composed of 20 observations and it contains four explanatory variables, \( x_1, x_2, x_3 \) and \( x_4 \). The original survey weights are in the variable **weight** and the numeric identifier for each observation is the variable **id**:

\[
\text{use http://fmwww.bc.edu/RePEc/bocode/r/reweight, clear}
\]

3. New starting values are obtained from a random perturbation of the Lagrange multipliers obtained with the chi-squared distance function. The perturbation for each multiplier is drawn from a uniform distribution with range -1 to 1.
The estimated survey totals are:

```
. tabstat x1 x2 x3 x4 [aw=weight], s(su)
```

<table>
<thead>
<tr>
<th></th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum</td>
<td>44</td>
<td>24</td>
<td>213</td>
<td>32</td>
</tr>
</tbody>
</table>

Let us suppose that external population totals were known for x1, x2, x3 and x4. We store these known totals in a Stata matrix:

```
. matrix t=(50 \ 20 \ 230 \ 35)
```

`reweight` can be used to obtain new weights that will calibrate the estimated totals to the external ones:

```
. foreach function in "chi2" "a" "b" "c" "ds" {
  2. cap reweight x1 x2 x3 x4, sw(weight) nw(w`function´) tot(t) df(`function´)
  3. }
```

<table>
<thead>
<tr>
<th></th>
<th>wchi2</th>
<th>wa</th>
<th>wb</th>
<th>wc</th>
<th>wds</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>2.753</td>
<td>2.674</td>
<td>2.654</td>
<td>2.697</td>
<td>2.706</td>
</tr>
<tr>
<td>2.</td>
<td>2.109</td>
<td>2.228</td>
<td>2.260</td>
<td>2.193</td>
<td>2.178</td>
</tr>
<tr>
<td>3.</td>
<td>5.945</td>
<td>5.998</td>
<td>6.012</td>
<td>5.982</td>
<td>5.976</td>
</tr>
<tr>
<td>4.</td>
<td>4.005</td>
<td>3.944</td>
<td>3.926</td>
<td>3.963</td>
<td>3.974</td>
</tr>
<tr>
<td>5.</td>
<td>2.484</td>
<td>2.514</td>
<td>2.521</td>
<td>2.505</td>
<td>2.501</td>
</tr>
<tr>
<td>6.</td>
<td>4.589</td>
<td>4.456</td>
<td>4.423</td>
<td>4.495</td>
<td>4.510</td>
</tr>
</tbody>
</table>
Survey reweighting

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7.</td>
<td>5</td>
<td>7.52</td>
<td>5.729</td>
<td>5.717</td>
<td>5.739</td>
</tr>
<tr>
<td>8.</td>
<td>4</td>
<td>4.005</td>
<td>3.944</td>
<td>3.926</td>
<td>3.963</td>
</tr>
<tr>
<td>9.</td>
<td>3</td>
<td>2.109</td>
<td>2.228</td>
<td>2.260</td>
<td>2.193</td>
</tr>
<tr>
<td>10.</td>
<td>3</td>
<td>3.120</td>
<td>3.086</td>
<td>3.074</td>
<td>3.098</td>
</tr>
<tr>
<td>11.</td>
<td>5</td>
<td>5.945</td>
<td>5.998</td>
<td>6.012</td>
<td>5.982</td>
</tr>
<tr>
<td>12.</td>
<td>4</td>
<td>3.986</td>
<td>3.814</td>
<td>3.762</td>
<td>3.870</td>
</tr>
<tr>
<td>13.</td>
<td>4</td>
<td>5.019</td>
<td>5.108</td>
<td>5.136</td>
<td>5.090</td>
</tr>
<tr>
<td>15.</td>
<td>5</td>
<td>4.678</td>
<td>4.665</td>
<td>4.666</td>
<td>4.667</td>
</tr>
<tr>
<td>16.</td>
<td>3</td>
<td>2.345</td>
<td>2.370</td>
<td>2.380</td>
<td>2.360</td>
</tr>
<tr>
<td>17.</td>
<td>4</td>
<td>5.070</td>
<td>5.191</td>
<td>5.232</td>
<td>5.150</td>
</tr>
<tr>
<td>18.</td>
<td>5</td>
<td>4.614</td>
<td>4.603</td>
<td>4.604</td>
<td>4.603</td>
</tr>
<tr>
<td>19.</td>
<td>4</td>
<td>4.967</td>
<td>5.028</td>
<td>5.043</td>
<td>5.010</td>
</tr>
<tr>
<td>20.</td>
<td>3</td>
<td>2.109</td>
<td>2.228</td>
<td>2.260</td>
<td>2.193</td>
</tr>
</tbody>
</table>

where the results are the same as to those reported in Creedy (2003) (Table 4).

As discussed in section 2, the DS distant function allows users to set the degree of the adjustment of the calibrated weights with respect to the survey weight:

```stata
reweight x1 x2 x3 x4, sw(weight) nw(wds1) tot(t) df(ds) upb(1.25) lowb(0.80)
Iteration 1
(output omitted)
Iteration 12
Iteration 13 - Converged
```

New totals using reweighted variables

<table>
<thead>
<tr>
<th>stats</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
</tr>
</thead>
<tbody>
<tr>
<td>sum</td>
<td>50</td>
<td>20</td>
<td>230</td>
<td>35</td>
</tr>
</tbody>
</table>

New weights obtained from the ds distance function

Current bounds: upper=1.25 - lower=.8

```stata
format wds1 %9.3f
.list weight wchi2 wds wds1
```

<table>
<thead>
<tr>
<th>weight</th>
<th>wchi2</th>
<th>wds</th>
<th>wds1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>3</td>
<td>2.753</td>
<td>2.706</td>
</tr>
<tr>
<td>2.</td>
<td>3</td>
<td>2.109</td>
<td>2.178</td>
</tr>
<tr>
<td>3.</td>
<td>5</td>
<td>5.945</td>
<td>5.976</td>
</tr>
<tr>
<td>4.</td>
<td>4</td>
<td>4.005</td>
<td>3.974</td>
</tr>
<tr>
<td>5.</td>
<td>2</td>
<td>2.484</td>
<td>2.501</td>
</tr>
<tr>
<td>6.</td>
<td>5</td>
<td>4.589</td>
<td>4.510</td>
</tr>
<tr>
<td>7.</td>
<td>5</td>
<td>5.752</td>
<td>5.747</td>
</tr>
<tr>
<td>8.</td>
<td>4</td>
<td>4.005</td>
<td>3.974</td>
</tr>
<tr>
<td>9.</td>
<td>3</td>
<td>2.109</td>
<td>2.178</td>
</tr>
<tr>
<td>10.</td>
<td>3</td>
<td>3.120</td>
<td>3.106</td>
</tr>
<tr>
<td>11.</td>
<td>5</td>
<td>5.945</td>
<td>5.976</td>
</tr>
<tr>
<td>12.</td>
<td>4</td>
<td>3.985</td>
<td>3.897</td>
</tr>
<tr>
<td>13.</td>
<td>4</td>
<td>5.019</td>
<td>5.065</td>
</tr>
<tr>
<td>14.</td>
<td>3</td>
<td>3.490</td>
<td>3.494</td>
</tr>
<tr>
<td>15.</td>
<td>5</td>
<td>4.678</td>
<td>4.665</td>
</tr>
</tbody>
</table>
Where the results are again identical to those reported in Creedy (2003).

The following figure shows the calibrated weights obtained with different distant functions, where the observations are arranged in ascending order with respect to the original weight:

```
. sort weight id
. gen id1=_n
. twoway (line wchi2 id1) (line wa id1) (line wb id1) (line wc id1) (line wds id1 > d1) (line wds1 id1, sort)
```

As the figure shows all the calibrated weights but those obtained with the DS function with bounds in the range 1.25-0.8 behave in a very similar manner. From the figure it can also be seen that reducing the allowed range for the ratio of calibrated to survey weights may place several weights to the range limits, particularly at the lower one given the chosen bounds.

4. In fact, the bottom part of the wds1 line is substantially “smoothed” with respect to the other lines, meaning that more weights are now placed at that limit. This result was also found in the empirical application contained in Creedy (2003) and should advice some carefulness when constraining the maximum deviation of the calibrated weights from the survey ones.
We now show the `ntries` option, which can help when the external totals are particularly different from the estimated ones. Let us redefine the matrix with the external totals, using numbers that are quite different from the estimated aggregates:

```
.matrix t=(70 \ 10 \ 250 \ 90)
```

Given these totals the chi-squared function requires such an adjustment of the new weights that some of them become negative. Moreover, also the other distant function but the type-C do not achieve convergence within the maximum number of iterations.

However, if we use the option `ntries(50)` a solution is found for the type-A and type-B distant functions after 8 and 9 tries respectively. For the DS distant function convergence is achieved only by increasing the number of maximum iterations for each try and by widening the bounds of the ratio $\frac{w_i}{w_i}$.

```
.reweight x1 x2 x3 x4, sw(weight) nw(chi2) tot(t) df(chi2)
.New weights obtained from the chi2 distance function are negative, try with ot
> her distance functions
.set seed 1234567890
.reweight x1 x2 x3 x4, sw(weight) nw(wds) tot(t) df(ds) ntries(50) niter(100)
> mlowb(0.1) mupb(8)
Iteration 1
(output omitted)
Iteration 100 Not Converged within the maximum number of iterations, the algori
> thm now tries with new starting values and new bounds up to 50 times:
try number 1
(output omitted)
try number 46
Converged, new starting values and new bounds saved in the return list.
```

New totals using reweighted variables

<table>
<thead>
<tr>
<th>stats</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
</tr>
</thead>
<tbody>
<tr>
<td>sum</td>
<td>70</td>
<td>10</td>
<td>250</td>
<td>90</td>
</tr>
</tbody>
</table>

New weights obtained from the ds distance function
Current bounds: upper=6.9838 - lower=.1211
.set seed 1234567890
.reweight x1 x2 x3 x4, sw(weight) nw(wa) tot(t) df(a) ntries(50)
Iteration 1
(output omitted)
Iteration 50 Not Converged within the maximum number of iterations, the algori
> thm now tries with new starting values up to 50 times:
try number 1
(output omitted)
try number 8
Converged, new starting values saved in the return list.

New totals using reweighted variables

---

5. When the `ntries` option is active then the new bounds for the DS function are drawn from two uniform distributions with ranges 0.3-1 for the lower bound and 1-6 for the upper bound. However, these ranges might not be wide enough when the new totals are relatively different from the estimated ones. In our application, for example, we had to widen the ranges as follows: 0.1-1 for the lower bound and 1 to 8 for the upper bound.
New weights obtained from the a distance function
. set seed 1234567890
. reweight x1 x2 x3 x4, sw(weight) nw(wb) tot(t) df(b) ntries(50)
Iteration 1
(output omitted)
Iteration 50 Not Converged within the maxiumum number of iterations, the algorit
> hm now tries with new starting values up to 50 times:
try number 1
(output omitted)
try number 18
Converged, new starting values saved in the return list.
New totals using reweighted variables
\[
\begin{array}{cccc}
\text{stats} & x1 & x2 & x3 & x4 \\
\text{sum} & 70 & 10 & 250 & 90 \\
\end{array}
\]

New weights obtained from the b distance function
. set seed 1234567890
. reweight x1 x2 x3 x4, sw(weight) nw(wc) tot(t) df(c)
Iteration 1
(output omitted)
Iteration 6 - Converged
New totals using reweighted variables
\[
\begin{array}{cccc}
\text{stats} & x1 & x2 & x3 & x4 \\
\text{sum} & 70 & 10 & 250 & 90 \\
\end{array}
\]

New weights obtained from the c distance function

5 References


About the author
Daniele Pacifico works with the Italian Department of the Treasury (Rome, Italy).
Ministry of Economy and Finance

Department of the Treasury

Directorate I: Economic and Financial Analysis

Address:
Via XX Settembre, 97
00187 - Rome

Websites:
www.mef.gov.it
www.dt.tesoro.it

e-mail:
dt.segretaria.direzione1@tesoro.it

Telephone:
+39 06 47614202
+39 06 47614197

Fax:
+39 06 47821886