Fast SAS Macros for balancing Samples user's guide

Guillaume Chauvet[1] and Yves Tillé[2]

(1) Laboratoire de Statistique d'Enquête
CREST - ENSAI
École Nationale de la Statistique et de l'Analyse de l'Information
rue Blaise Pascal, Campus de Ker Lann, 35170 Bruz, France
guillaume.chauvet@ensai.fr

(2) Institute of Statistics, University of Neuchâtel,
Espace de l'Europe 4, CP 805, 2002 Neuchâtel, Switzerland
yves.tille@unine.ch

November 9, 2005

Introduction

The Cube Method is a family of algorithms that enables to select balanced samples with equal or unequal inclusion probabilities. Balancing samples is an old issue ; Deville et al. (1988) gave a solution in case of equal inclusion probabilities and Ardilly (1991) gave a general but time expensive method. A general solution, usable for large files, only came recently (Deville and Tillé, 2004). In this user's guide, we present a very fast implementation of the Cube Method as well as macros for balancing samples based on this algorithm.

The paper is organised as follows. In Section 1, a short remind of the principles of the Cube Method is given and the new algorithm is presented. This part is extracted from an article submitted with Yves Tillé (Chauvet and Tillé (2004)). For a detailed presentation of the Cube Method, see Deville and Tillé (2004) and Deville and Tillé (2005). In Section 2, the macro `%exe_cube` (version 1) using the fast algorithm is presented and examples are given. In Section 3, a particular use of balanced sampling called stratified sampling is presented. The macro `%echant_strat` (version 1), which enables to perform stratified sampling, is presented with examples in the Section 4.

It should be noticed that both `exe_cube` and `echant_strat` use SAS IML and some sub-macros, namely the macros: `vol`, `atterrissage1`, `atterrissage2` and `atterrissage3`. All these macros can be found in the SAS files "`fast_cube`" and "`fast_cube_stratification`".

The softwares `exe_cube` and `echant_strat` are free and come with no warranty. You are welcome to contact Guillaume Chauvet at `chauvet@ensai.fr` for any comment or suggestion.

1 A fast algorithm of balanced sampling

1.1 Introduction

The Cube method, that allows the selection of balanced samples, was developed at ENSAI (France) (see Deville and Tillé, 2004, 2005; Tillé, 2001) and students of the Ecole Nationale de la Statistique et de l'Analyse de l'Information (ENSAI) initially wrote the program. The program currently used at Institut National de la Statistique et des Etudes Economiques (INSEE, France) was written by Frederic Tardieu, and then finalized by Bernard Weytens thanks to improvements suggested by the Unit of Statistical Methods of the INSEE and by the Methodological Unit of the Renovated Census.

The method was first dedicated to the selection of primary units in two-stage sampling, because the execution was proportional to the square of the population size. This method was already applied to several important statistical problems. For instance, the rotation groups of municipalities and addresses of the French renovated census were selected by means of the cube method (Bertrand et al. (2004) ; Dumais and Isnard (2000)). The cube method is actually a family of algorithm the implementation of which admits a lot of variants. We propose a very fast implementation. The originality consists of applying the basic step on a subset of units and not on the whole population. This subset evolves at each step of the algorithm, and the execution time doesn't depend any more on the square of the population size.

In Section 1.2, the notation is defined. In Section 1.3, we give a brief reminder of the cube method. The new algorithm is proposed in 1.4. Next, in Section 1.5, we discuss the implementation of the fast algorithm, and some numerical results are presented. Finally, in Section 1.6, we show that this algorithm can be applied to the problem of unequal probability sampling.

1.2 Notation and balance sampling

Consider a finite population U of size N whose units can be identified by labels $k \in \{1, \ldots, N\}$. The aim is to estimate the total $Y = \sum_{k \in U} y_k$ of a variable of interest y that takes the values $y_k, k \in U$, for the units of the population. Suppose also that the vectors of values $\mathbf{x}_k = (x_{k1} \ \ldots \ x_{kj} \ \ldots \ x_{kp})'$ taken by p auxiliary variables are known for all the units of the population. The p vectors $(x_{1j} \ \ldots \ x_{kj} \ \ldots \ x_{Nj})', j = 1, \ldots, p$, are assumed without loss of generality to be linearly independent.

A sample is denoted by a vector $\mathbf{s} = (s_1 \ \ldots \ s_k \ \ldots \ s_N)'$, of $\mathbb{R}^N$ where s_k takes the value 1 if k is in the sample and is 0 otherwise. A sampling design $p(.)$ is a probability distribution on the set $\mathcal{S} = \{0, 1\}^N$ of all the possible samples. The random sample $\mathbf{S}$ is a random vector of $\mathbb{R}^N$ that takes the value $\mathbf{s}$ with probability $\Pr(\mathbf{S} = \mathbf{s}) = p(\mathbf{s})$. The inclusion probability of unit k is the probability $\pi_k = \Pr(S_k = 1)$ that unit k is in the sample, and the vector of inclusion probabilities is $\boldsymbol{\pi} = (\pi_1 \ \ldots \ \pi_k \ \ldots \ \pi_N)'$. Note that

$$\boldsymbol{\pi} = E(\mathbf{S}) = \sum_{\mathbf{s} \in \mathcal{S}} p(\mathbf{s})\mathbf{s} \in \mathbb{R}^N.$$

The joint inclusion probability $\pi_{k\ell} = \Pr(S_k = 1 \text{ and } S_\ell = 1)$ is the probability that two distinct units are jointly in the sample. The Horvitz-Thompson estimator given by $\widehat{Y} = \sum_{k \in U} S_k y_k / \pi_k$ is an unbiased estimator of Y. The Horvitz-Thompson estimator of the jth auxiliary total $X_j = \sum_{k \in U} x_{kj}$

is $\widehat{X}_j = \sum_{k\in U} S_k x_{kj}/\pi_k$. The Horvitz-Thompson estimator vector,

$$\widehat{\mathbf{X}} = \sum_{k\in U} S_k \mathbf{x}_k/\pi_k,$$

estimates without bias the totals of the auxiliary variables, $\mathbf{X} = \sum_{k\in U} \mathbf{x}_k$.

The aim is to construct a balanced sampling design, i.e. a sampling design such that $\widehat{\mathbf{X}} = \mathbf{X}$, and that satisfies a predetermined vector of inclusion probabilities $\boldsymbol{\pi}$. Nevertheless, in most cases, an exact balanced sampling design does not exist. The objective is thus to find an approximately balanced design, that is, one for which $\widehat{\mathbf{X}} \approx \mathbf{X}$. If $\mathbf{a}_k = \mathbf{x}_k/\pi_k$, and $\mathbf{A}$ is a $p \times N$ matrix such that

$$\mathbf{A} = (\mathbf{a}_1 \ \ldots \ \mathbf{a}_k \ \ldots \ \mathbf{a}_N),$$

then a balanced sampling design is such that

$$\mathbf{AS} = \mathbf{A}\boldsymbol{\pi}, \tag{1}$$

which is called the system of balancing equations.

1.3 The cube method

The cube method is composed of two phases called the flight phase and the landing phase. In the flight phase, the constraints are always exactly satisfied. The objective is to round off randomly to 0 or 1 almost all the inclusion probabilities. The landing phase consists of managing as well as possible the fact that the system of balancing equations (1) cannot always be exactly satisfied. The flight phase is described in Algorithm 1.

Algorithm 1: General balanced procedure: flight phase

First initialize at $\boldsymbol{\pi}(0) = \boldsymbol{\pi}$. Next, at time $t = 0, ..., T$, repeat the three following steps.

Step 1: Generate any vector $\mathbf{u}(t) = \{u_k(t)\} \neq 0$, random or not, such that $\mathbf{u}(t)$ is in the kernel of the matrix $\mathbf{A}$, and $u_k(t) = 0$ if $\pi_k(t)$ is an integer.

Step 2: Compute $\lambda_1^*(t)$ and $\lambda_2^*(t)$, the largest values of $\lambda_1(t)$ and $\lambda_2(t)$ such that $0 \leq \boldsymbol{\pi}(t) + \lambda_1(t)\mathbf{u}(t) \leq 1$, and $0 \leq \boldsymbol{\pi}(t) - \lambda_2(t)\mathbf{u}(t) \leq 1$. Note that $\lambda_1(t) > 0$ and $\lambda_2(t) > 0$.

Step 3: Select

$$\boldsymbol{\pi}(t+1) = \begin{cases} \boldsymbol{\pi}(t) + \lambda_1^*(t)\mathbf{u}(t) & \text{with probability } q(t) \\ \boldsymbol{\pi}(t) - \lambda_2^*(t)\mathbf{u}(t) & \text{with probability } 1 - q(t), \end{cases} \tag{2}$$

where $q(t) = \lambda_2^*(t)/\{\lambda_1^*(t) + \lambda_2^*(t)\}$.

The general procedure is repeated until it is no longer possible to carry out Step 1.

If T is the last step of Algorithm 2 and that $\boldsymbol{\pi}^* = \boldsymbol{\pi}(T)$, then Deville and Tillé (2004) have shown that,

1. $E(\boldsymbol{\pi}^*) = \boldsymbol{\pi}$,
2. $\mathbf{A}\boldsymbol{\pi}^* = \mathbf{A}\boldsymbol{\pi}$,
3. if $q = \text{card}\{k|0 < \pi_k^* < 1\}$, then $q \leq p$, where p is the number of auxiliary variables.

Vector $\boldsymbol{\pi}^*$ can be a sample, but in most cases, there are at most q non-integer elements in $\boldsymbol{\pi}^*$. If $q > 0$, the rounding problem, i.e. the fact that the system of balancing equations cannot be always satisfied, is processed by the 'landing phase'. Two solutions are possible for the landing phase (Deville and Tillé, 2004). The first one consists of relaxing a constraint and to run the flight phase again, until it is no longer possible to 'move' within the constraint hyperplane. The constraints are thus relaxed successively. The second solution uses a linear program for obtaining the best approximated balanced design (Deville and Tillé, 2004). The flight phase consumes most of the execution time. The new implementation concerns only the flight phase, and the landing phase remains unchanged.

Due to the complexity of the cube algorithm, the joint inclusion probabilities cannot be derived exactly. Deville and Tillé (2005) have however shown that the variance of a balanced sample can be well approximated and estimated without knowing the joint inclusion probabilities.

1.4 A very fast implementation

The aim of this new implementation is to obtain a reduction of the execution time. In the general algorithm, the search for a vector $\mathbf{u}$ in Ker$\mathbf{A}$ is extremely expensive. The basic idea is to use a submatrix $\mathbf{B}$ containing only $p+1$ columns of $\mathbf{A}$. Note that the number of variables p is smaller than the population size N, and that rank $\mathbf{B} \leq p$. The dimension of the kernel of $\mathbf{B}$ is thus larger or equal to 1.

A vector $\mathbf{v}$ of Ker$\mathbf{B}$ can then be used to construct a vector $\mathbf{u}$ of Ker$\mathbf{A}$ by complementing $\mathbf{v}$ with zeros for the columns of $\mathbf{B}$ that are not in $\mathbf{A}$. With this idea, all the computation can be done only on $\mathbf{B}$. This method is described in Algorithm 2.

If $\widetilde{T}$ is the last step of the algorithm and that $\widetilde{\boldsymbol{\pi}} = \boldsymbol{\pi}(\widetilde{T})$, then we have

1. $E(\widetilde{\boldsymbol{\pi}}) = \boldsymbol{\pi}$,
2. $\mathbf{A}\widetilde{\boldsymbol{\pi}} = \mathbf{A}\boldsymbol{\pi}$,
3. if $\tilde{q} = \text{card}\{k|0 < \tilde{\pi}_k < 1\}$, then $\tilde{q} \leq p$, where p is the number of auxiliary variables.

In the case where some of the constraints can be satisfied exactly, the flight phase can be continued. Suppose that $\mathbf{C}$ is the matrix containing the columns of $\mathbf{A}$ that correspond to non-integer values of $\widetilde{\boldsymbol{\pi}}$, and $\boldsymbol{\phi}$ is the vector of non-integer values of $\widetilde{\boldsymbol{\pi}}$. If $\mathbf{C}$ is not full-rank, one or several steps of the general Algorithm 1 can still be applied on $\mathbf{C}$ and $\boldsymbol{\phi}$. A return to the general Algorithm 1 is thus necessary for the last steps.

Algorithm 2: fast algorithm for the flight phase

(a) *Initialization*

(i) Before applying the algorithm, the units with null inclusion probabilities are removed from the population and the units with inclusion probabilities equal to 1 are definitively selected in the population. The algorithm is thus applied on the the remaining units such that $0 < \pi_k < 1$.

(ii) The inclusion probability are loaded in vector $\boldsymbol{\pi}$.

(iii) Vector $\boldsymbol{\psi}$ is made up of the first $p+1$ elements of $\boldsymbol{\pi}$.

(iv) A vector of ranks is created $\mathbf{r} = (1, 2, \ldots, p, p+1)'$.

(v) Matrix $\mathbf{B}$ is made up of the first $p+1$ columns of $\mathbf{A}$.

(vi) Initialize $k = p+2$.

(b) *Basic loop*

(i) A vector $\mathbf{u}$ is taken in the kernel of $\mathbf{B}$,

(ii) Only $\boldsymbol{\psi}$ is modified (and not the vector $\boldsymbol{\pi}$) according the basic technique. Compute λ_1^* and λ_2^*, the largest values of λ_1 and λ_2 such that $0 \leq \boldsymbol{\psi} + \lambda_1 \mathbf{u} \leq 1$, and $0 \leq \boldsymbol{\psi} - \lambda_2 \mathbf{u} \leq 1$. Note that $\lambda_1^* > 0$ and $\lambda_2^* > 0$.

(iii) Select $\boldsymbol{\psi} = \begin{cases} \boldsymbol{\psi} + \lambda_1^* \mathbf{u} \text{ with probability } q \\ \boldsymbol{\psi} - \lambda_2^* \mathbf{u} \text{ with probability } 1-q, \end{cases}$
where $q = \lambda_2^* / (\lambda_1^* + \lambda_2^*)$.

(iv) *(The units that corresponds to $\psi(i)$ integer are removed from* $\mathbf{B}$*, and are replaced by inclusion probabilities of new units. The algorithm stops at the end of the file.).*
For $i = 1, \ldots, p+1$:

If $\boldsymbol{\psi}(i) = 0$ or $\boldsymbol{\psi}(i) = 1$ then

- $\boldsymbol{\pi}(\mathbf{r}(i)) = \boldsymbol{\psi}(i)$
- $\mathbf{r}(i) = k$
- If $k \leq N$ then
  - $\boldsymbol{\psi}(i) = \boldsymbol{\pi}(k)$
  - For $j = 1, \ldots, p$, $\mathbf{B}(i,j) = \mathbf{A}(k,j)$
  - $k = k+1$
- ElseIf Goto Step (c) (i)

(v) Goto Step (b) (i)

(c) *End of the first part of the flight phase*

(i) For $i = 1, \ldots, p+1, \boldsymbol{\pi}(\mathbf{r}(i)) = \boldsymbol{\psi}(i)$.

1.5 Implementation and numerical results

The implementation of the fast algorithm is quite simple. Matrix $\mathbf{A}$ never has to be completely loaded in memory and thus remains in a file that can be read sequentially. For this reason, there is no restriction on the population size. The execution time depends linearly on the population size. The search for a vector $\mathbf{u}$ in the submatrix $\mathbf{B}$ limits the choice of the direction $\mathbf{u}$. In most cases, only one direction is possible. In order to increase the randomness of the sampling design, the units can possibly be randomly mixed before applying the algorithm.

Algorithm 2 has been implemented by means of a SAS-IML macro. We have tested the processing capacity of the algorithm to select samples in large populations with a lot of balancing variables. We have used a population of 313,702 units, corresponding to the addresses of the big municipalities (10,000 inhabitants or more) of the Rhone-Alpes French region. All the units are selected with the same inclusion probability equal to 1/5. The balancing variables are:

- a constant in order to obtain a fixed sample size,
- 18 sociodemographic variables, which are presented in Table 1,

- 81 variables that are the products of an indicator variable of the presence of the addresses in the 81 municipalities and the number of households in the addresses.

Table 1: List of the sociodemographic variables

NLOG	Number of households
NLOGCO	Number of households in collective addresses
H0019	Number of men, age less than 20
H2039	Number of men, ages 20 to 39
H4059	Number of men, ages 40 to 59
H6074	Number of men, ages 60 to 74
H7599	Number of men, age more than 75
F0019	Number of women, age less than 20
F2039	Number of women, ages 20 to 39
F4059	Number of women, ages 40 to 59
F6074	Number of women, ages 60 to 74
F7599	Number of women, age more than 75
ACTIFS	Number of working persons
INACTIFS	Number of nonworkers
NATFN	Number of people with French nationality by birth
NATFA	Number of people with French nationality by acquisition
NATHE	Number of foreigners outside European Union
NATUE	Number of foreigners from European Union

The last 81 variables ensure that the number of households is balanced in each municipality. One hundred balancing variables are thus used. A sample of 62,741 addresses is then selected, by means of a personal computer (Pentium 3, 1 Gh). The population has been sorted in decreasing order of the size of the address. The selection has been completed in about 1 hour and 50 minutes. The condition of fixed size is perfectly realized. With the former version of the algorithm the execution time should be multiplied by about 3000. Next we have computed the ratio of the square of the difference between the Horvitz-Thompson estimators of the total and the variances of the Horvitz-Thompson estimator under simple random sampling:

$$R = \frac{(\widehat{X}_j - X)^2}{\mathrm{var}_{\mathrm{simple}}(\widehat{X}_j)}.$$

These ratios are presented in Table 2, and show the dramatic improvement of accuracy.

1.6 Case of unequal probability sampling

When $J = 1$ and that the only auxiliary variable is $x_k = \pi_k$, then the problem of balanced sampling amounts to sampling with unequal probability. In this case, $\mathbf{A} = (1 \dots 1)$. At each step, matrix $\mathbf{B} = (1, 1)$, and $\mathbf{u} = (-1, 1)'$. Algorithm 2 can be simplified dramatically as presented in Algorithm 3 that is a very simple method of sampling with unequal inclusion probabilities. Actually Algorithm 3 is an implementation of the pivotal method (Deville and Tillé, 1998) in the framework of the splitting method.

Table 2: Ratio of the square of the difference between the Horvitz-Thompson estimators of the total of the variances under simple random sampling

Variable	Ratio	Variable	Ratio	Variable	Ratio
NLOG	$2.7\ 10^{-5}$	H7599	$1.2\ 10^{-4}$	ACTIFS	$9.7\ 10^{-7}$
NLOGCO	$2.5\ 10^{-5}$	F0019	$6.0\ 10^{-6}$	INACTIFS	$2.3\ 10^{-5}$
H0019	$1.4\ 10^{-6}$	F2039	$1.9\ 10^{-6}$	NATFN	$2.0\ 10^{-5}$
H2039	$8.2\ 10^{-5}$	F4059	$2.0\ 10^{-6}$	NATFA	$1.3\ 10^{-5}$
H4059	$3.0\ 10^{-7}$	F6074	$8.6\ 10^{-5}$	NATHE	$7.8\ 10^{-5}$
H6074	$3.2\ 10^{-5}$	F7599	$6.7\ 10^{-5}$	NATUE	$5.8\ 10^{-4}$

Algorithm 3: pivotal methof for unequal inclusion probabilities			
	Eventually sort the data in a random order		
	Definition a, b, u real; i, j, k integer;		
	$a = \pi_1; b = \pi_2; i = 1; j = 2;$		
	For $k = 1, \ldots, N : s_k = 0;$		
	$k = 3$		
	While $k \leq n$:		
		u = uniform random variable in [0,1]	
		If $a + b > 1$ then	If $u < \frac{1-b}{2-a-b}$: $b = a + b - 1$; $a = 1$
			Else $a = a + b - 1$; $a = 1$
		If $k \leq N$ then	If $u < \frac{b}{a+b}$: $b = a + b$; $a = 0$
			Else $a = a + b - 1$; $b = 0$
		If a is an integer and $k \leq n$ then $s_i = a$; $a = \pi_k$; $i = k$; $k = k+1$	
		If b is an integer and $k \leq n$ then $s_j = b$; $b = \pi_k$; $j = k$; $k = k+1$	
	$s_i = a$; $s_j = b$		

1.7 Case of Poisson sampling

Deville notices that with $p = 0$, i.e. without any balancing equations, the fast algorithm amounts to Poisson sampling. Algorithm 2 can then be simplified as follows:

Algorithm 4: Poisson sampling			
	Definition: ψ real ; i, k integer		
	Initialise $\psi = \pi_1$; $i = 1$; $k = 2$		
	u is any non zero scalar		
	While $k \leq N$		
		If $u > 0$ then	$\lambda_1^* = \frac{1-\psi}{u}$; $\lambda_2^* = \frac{\psi}{u}$
			Select $\boldsymbol{\psi} = \begin{cases} \boldsymbol{\psi} + \lambda_1^* \mathbf{u} = 1 \text{ with proba } q \\ \boldsymbol{\psi} - \lambda_2^* \mathbf{u} = 0 \text{ with proba } 1 - q, \end{cases}$ where $q = \frac{\lambda_2^*}{\lambda_1^* + \lambda_2^*} = \pi_k$
		If $u < 0$ then	$\lambda_1^* = -\frac{\psi}{u}$; $\lambda_2^* = -\frac{1-\psi}{u}$
			Select $\boldsymbol{\psi} = \begin{cases} \boldsymbol{\psi} + \lambda_1^* \mathbf{u} = 0 \text{ with proba } q \\ \boldsymbol{\psi} - \lambda_2^* \mathbf{u} = 1 \text{ with proba } 1 - q, \end{cases}$ where $q = \frac{\lambda_2^*}{\lambda_1^* + \lambda_2^*} = 1 - \pi_k$

2 The macro for balanced sampling

2.1 Description

The `exe_cube` macro enables to select a balanced sample and returns a data table containing the result of the sampling.

2.2 The Input Data

The data relative to the population in which we want to select a balanced sample must be put into a SAS table, containing all units of the population, and at least:

- An identifying variable
- The variable of inclusion probabilities
- The balancing variables

This table may not contain missing values for the variables quoted below. The variable of inclusion probabilities, as well as the balancing variables, must be of numerical type.

2.3 Syntax of the macro

2.3.1 Parameters relative to the Data Base

All these parameters are compulsory.

- BASE = name of SAS library
  Name of the SAS library containing the SAS table of Input data.
- DATA = name of SAS table
  Name of the SAS table containing the Input data.
- ID = variable
  Name of the variable that identifies the units of the population
- PI = variable
  Name of the variable of inclusion probabilities
- CONTR = variable(s)
  Names of the variables on which the sample will be balanced. The names must be spaced with blanks.

2.3.2 Parameters relative to the sampling

All these parameters are optional.

- ATTER = option

  States the option selected for the landing phase. Possible values are:

  - ATTER = 1
    The balancing variables are progressively abandoned. The last variable in the CONTR parameter is removed first, then the variable before and so on.
  - ATTER = 2
    The landing phase is performed by considering all the possible samples among the remaining units, and selecting preferably those providing a low difference to the balancing.
  - ATTER = 3
    The landing is performed like with ATTER=2, but only considering the samples whose size equals the sum of inclusion probabilities. We obtain a fixed sample size. If this option is used, the variable of inclusion probabilities must be put in the CONTR parameter.

  The default value is: ATTER=1. This is the fastest landing option. To ensure a reasonable execution time, the option ATTER=2 should not be used with more than 14 balancing variables, and the option ATTER=3 should not be used with more than 18 balancing variables.

- COMPEQ = option

  Equals 1 if the complementary of the sample has to be balanced on the same variables too, and 0 otherwise. The default value is: COMPEQ=0

  Here we use a result of Tillé and Favre (2004). The proof can be found in Annexe 1. This option allows selecting several non-overlapping samples, balanced on the same variables, with fixed inclusion probabilities. Suppose we want to select two non-overlapping samples, balanced on the variable x, with inclusion probabilities π_k. We select the first balanced sample S_1 as usual, with option COMPEQ=1. Then we select a sample S_2 in the complementary of S_1, with inclusion probabilities $\frac{\pi_k}{1-\pi_k}$, balanced on the variable $(z_k) = \left(\frac{x_k}{1-\pi_k}\right)$. This method can be applied to any number of balancing variables. We can select up to $\min_{k\in U} \left\lfloor \frac{1}{\pi_k} \right\rfloor$ balanced samples with this method, where ($\lfloor x \rfloor$ is the larger integer smaller than n). This option multiplies by 2 the number of balancing variables, thus by about 4 the execution time.

  If all inclusion probabilities are equal, the complementary of the sample is automatically balanced on the same variables, so the option becomes useless. See Appendix 1 for details.

2.3.3 Parameters relative to the Output

- SORT = name of SAS table

  Name of the SAS table containing the Output data. This table belongs to the library quoted in BASE. Il contains all the units of the population, and a variable ECH equal to 1 if the unit has been selected in the sample, and 0 otherwise.

2.3.4 Some examples

```
/*****************************************/
/* Definition of the data                */
/* id is an identifying variable         */
/* pi1 and pi2 are two different variables*/
/*    of inclusion probabilities         */
/* var1, var2 and var3 are 3 numerical   */
/*    variables                          */
/*****************************************/
data a1;
input id $ pi1 pi2 var1 var2 var3;
cards;
1 0.3 0.2 1 4 7
2 0.3 0.4 2 6 4
3 0.3 0.8 3 7 9
4 0.3 0.2 2 3 2
5 0.3 0.7 8 6 4
6 0.3 0.1 3 5 6
7 0.3 0.5 2 6 3
8 0.3 0.3 2 1 3
9 0.3 0.4 4 7 6
10 0.3 0.5 8 2 5
;run;
/*****************************************/
/* Example 1                             */
/*****************************************/
/* Balancing on var1 and var3, with pi2 as*/
/* inclusion probabilities, with the 1st  */
/* landing option, without balancing the  */
/* complementary                         */
/*****************************************/
%exe_cube(base=work,data=a1,id=id,pi=pi2,contr=var1 var3,
            sort=ech,atter=1,compeq=0);
/*****************************************/
/* Example 2                             */
/*****************************************/
/* Balancing on var1, var2 and var3, with */
/* pi2 as inclusion probabilities and     */
/* fixed size, with the 2d landing option,*/
/* and balancing the complementary        */
/* (fixed size compels to put pi2 as a    */
/* balancing variable)                    */
/*****************************************/
%exe_cube(base=work,data=a1,id=id,pi=pi2,contr=pi2 var1 var2 var3,
          sort=ech,atter=2,compeq=1);
/*****************************************/
/* Example 3                             */
```

```
/****************************************/
/* Balancing on var1 and var2, with pi1 as*/
/* inclusion probabilities, with the 3st  */
/* landing option, without balancing the  */
/* complementary                          */
/* (in fact, inclusion probabilities are  */
/* equal, so the complementary is         */
/* automatically balanced)                */
/* (the use of the 3d landing option      */
/* compels to put pi1 as a balancing      */
/* variable)                              */
/****************************************/
%exe_cube(base=work,data=a1,id=id,pi=pi1,contr=pi1 var1 var2,
          sort=ech,atter=3,compeq=0);
```

2.3.5 Some numerical examples

We use a population of 26471 units corresponding to the city of Lyon, given by the 1999 Census. The samples are selected by means of a personal computer (Penthium 4, 1.8 Gh).

Example 1 We first select a sample with equal probabilities $\frac{1}{5}$, balanced on the socio-demographic variables quoted in Table 1 (18 variables) and on a constant for the condition of fixed sample size. We use the first landing option. A sample of 5345 units is drawn in a few seconds. Results are presented in Table 3.

Example 2 We want to select a sample of 1500 addresses, with probabilities proportional to the size of the address (the size is given by the number of households), balanced on the socio-demographic variables quoted in Table 1 (18 variables). We also balance on the variable of inclusion probabilities and use the third landing option, for the sample to be of exact fixed size. The sample is drawn in less than one minute. The condition of fixed size is perfectly realized. Results are presented in Table 4.

Example 3 Now, suppose we want to select several samples, balanced on the former variables. We still use probabilities proportional to the size of the address ; we want to select 3 samples of 500 addresses. In the population, all reverse inclusion probabilities are higher than 3.98, thus the coordinated sampling is possible. We also balance on the variable of inclusion probabilities. We use the first landing option and the option COMPEQ=1. Indeed, as the number of balancing variables is very big (38, corresponding to the 19 basics balancing variables, and 19 other variables generated by option COMPEQ=1), the sampling couldn't be performed in a reasonable time with options ATTER=2 or 3.
Results are presented in Table 5, page 13. The fixed size is perfectly obtained for each of the samples.

Table 3: Relative difference between the real total and the Horvitz-Thompson estimator of the total for the balancing variables

Variable	Horvitz-Thompson estimator of the total	Real total	Relative difference (%)
NLOG	251 380	251 279	-0,04%
NLOGCO	243 480	243 381	-0,04%
H0019	46 390	46 395	0,01%
H2039	75 145	75 116	-0,04%
H4059	46 080	46 078	0,00%
H6074	20 735	20 726	-0,04%
H7599	10 440	10 435	-0,05%
F0019	46 145	46 156	0,02%
F2039	83 980	83 957	-0,03%
F4059	51 900	51 881	-0,04%
F6074	28 645	28 637	-0,03%
F7599	21 440	21 421	-0,09%
ACTIFS	206 780	206 732	-0,02%
INACTIFS	224 120	224 070	-0,02%
NATFN	376 425	376 326	-0,03%
NATFA	21 815	21 833	0,08%
NATHE	22 990	22 978	-0,05%
NATUE	9 670	9 665	-0,05%

Table 4: Relative difference between the real total and the Horvitz-Thompson estimator of the total for the balancing variables

Variable	Horvitz-Thompson estimator of the total	Real total	Relative difference (%)
NLOG	251 279	251 279	0,00%
NLOGCO	243 071	243 381	0,13%
H0019	46 596	46 395	-0,43%
H2039	75 091	75 116	0,03%
H4059	46 195	46 078	-0,25%
H6074	20 733	20 726	-0,03%
H7599	10 495	10 435	-0,57%
F0019	46 196	46 156	-0,09%
F2039	83 966	83 957	-0,01%
F4059	51 983	51 881	-0,20%
F6074	28 644	28 637	-0,02%
F7599	21 512	21 421	-0,42%
ACTIFS	206 834	206 732	-0,05%
INACTIFS	224 576	224 070	-0,23%
NATFN	376 919	376 326	-0,16%
NATFA	21 906	21 833	-0,33%
NATHE	22 993	22 978	-0,07%
NATUE	9 591	9 665	0,76%

Table 5: Relative difference between the real total and the Horvitz-Thompson estimator of the total for the balancing variables

Variable	Real total	HT estimator of the total given by the 1st sample	Relative Difference (%)	HT estimator of the total given by the 2nd sample	Relative Difference (%)	HT estimator of the total given by the 3d sample	Relative Difference (%)
NLOG	251 279	251 279	0,00%	251 279	0,00%	251 279	0,00%
NLOGCO	243 381	243 238	0,06%	243 238	0,06%	243 741	0,13%
H0019	46 395	46 541	-0,31%	46 549	-0,33%	46 408	-0,43%
H2039	75 116	74 940	0,23%	75 042	0,10%	75 347	0,03%
H4059	46 078	46 686	-1,32%	46 229	-0,33%	46 196	-0,25%
H6074	20 726	20 715	0,05%	20 687	0,19%	20 754	-0,03%
H7599	10 435	10 093	3,27%	10 548	-1,08%	10 099	-0,57%
F0019	46 156	46 639	-1,05%	46 433	-0,60%	46 456	-0,09%
F2039	83 957	84 069	-0,13%	84 121	-0,20%	84 187	-0,01%
F4059	51 881	51 753	0,25%	52 173	-0,56%	52 282	-0,20%
F6074	28 637	28 914	-0,97%	28 479	0,55%	28 540	-0,02%
F7599	21 421	21 270	0,71%	21 482	-0,28%	21 044	-0,42%
ACTIFS	206 732	207 197	-0,22%	206 907	-0,08%	207 851	-0,05%
INACTIFS	224 070	224 422	-0,16%	224 835	-0,34%	223 462	-0,23%
NATFN	376 326	376 200	0,03%	378 181	-0,49%	377 177	-0,16%
NATFA	21 833	22 435	-2,76%	21 260	2,63%	21 348	-0,33%
NATHE	22 978	23 431	-1,97%	22 820	0,69%	23 391	-0,07%
NATUE	9 665	9 553	1,16%	9 482	1,89%	9 397	0,76%

3 Global balancing and stratified balancing

3.1 Notation

We keep the same notation as in part 1. We suppose here that U is divided into H non-overlapping strata $U_1, \ldots, U_H$. We remind that the sampling design is said to be balanced on the variable x if

$$\sum_{k \in U} \frac{S_k x_k}{\pi_k} = \sum_{k \in U} x_k$$

We say that the sampling design is balanced by strata on the variable x if

$$\sum_{k \in U_h} \frac{S_k x_k}{\pi_k} = \sum_{k \in U_h} x_k, \text{ for all } h = 1 \ldots H \tag{3}$$

Note that if a sampling design is balanced by strata, it is globally balanced on the whole population.

This technique has been used in the French renovated census for the building of the rotation groups of small municipalities ; in each French region, these rotation groups are made up by selecting samples balanced globally on socio-demographic variables, and balanced by French department on the number of households (in order to ensure that a reasonable number of municipalities of each department can be found in any of the five rotation groups).

3.2 Drawbacks of a direct balancing by strata

Stratified balance sampling can be performed by selecting a sample directly in the whole population. Indeed, (3) is equivalent to

$$\sum_{k \in U} \frac{S_k (x_k \ 1_{k \in U_h})}{\pi_k} = \sum_{k \in U} x_k \ 1_{k \in U_h} \text{ for all } h = 1 \ldots H$$

We thus only need to select a sample in U, balanced on the variables equal to the product of the balancing variables $x_1, \ldots, x_p$ and the indicator variables:

$$1_{k \in U_h} = \begin{cases} 1 & \text{if } k \in U_h \\ 0 & \text{otherwise,} \end{cases}$$

which means balancing on $H \times p$ variables. This method has several drawbacks:

- If $H \times p$ is too big, we cannot perform the landing phase by searching the sample that gives a low difference to the balancing state, because the number of possible samples is too important. The only landing option available is the first, i.e. to progressively remove some constraints
- All strata don't have the same quality of balancing. With the first option for the landing phase, the balancing is worst for the stratum corresponding to the variables removed first
- The fixed size cannot be obtained in each stratum

The program developed here draws it's inspiration from a remark on the treatment of big data bases (Rousseau and Tardieu, 2004). The idea is the following:

- We first try to balance by strata: we perform a flight phase independently on each stratum, balancing on the auxiliary variables
- When it is no more possible to balance by strata, we look for a global balancing: we gather the units that have not been sampled or rejected during the flight phases in the strata, then we perform a last flight phase on all these units before landing

The justification can be found in Appendix 2.

4 The macro for stratified balancing

4.1 Description of the macro

The macro `echant_strat` enables to select a sample, globally balanced on the whole population and approximately balanced on strata.

4.2 The Input Data

There must be as many input SAS tables as strata in the population: each of these tables contain, for one particular stratum, the data relative to its units, and at least:

- The variable of inclusion probabilities
- The balancing variables

This table may not contain missing values for the variables quoted below. The variable of inclusion probabilities, as well as the balancing variables, must be of numerical type.

4.3 Syntax of the macro

4.3.1 Parameters relative to the Data Base

All these parameters are compulsory.

- BASE = name of SAS library
  Name of the SAS library containing the SAS tables of Input data.
- DATA = SAS table(s)
  Name(s) of the SAS table(s) containing the Input data. The names must be spaced with blanks. Each table contains the units of one stratum.
  For example, suppose that the population is stratified into 4 strata U_1, U_2, U_3, U_4. 3 tables are created, say STRAT1 for stratum U_1, gathering the units of U_1, STRAT2 for stratum

U_2, gathering the units of U_2, and so on. The syntax will be: DATA= STRAT1 STRAT2 STRAT3 STRAT4.

- PI = variable
  Name of the variable of inclusion probabilities

- CONTR =variable(s)
  Names of the variables on which the sample will be balanced. The names must be spaced with blanks.

4.3.2 Parameters relative to the Output

- SORT = name of SAS table
  Name of the SAS table containing the Output data. This table belongs to the library quoted in BASE. Il contains all the units of the population, and a variable ECH equal to 1 if the unit has been selected in the sample, and 0 otherwise.

4.4 An example

```
/*****************************************/
/* Definition of the data                */
/* id si an identifying variable         */
/* pi1 and pi2 are two different variables*/
/*    of inclusion probabilities         */
/* var1, var2 and var3 are 3 numerical   */
/*    variables                          */
/*****************************************/
/* The population U is divided into 2 non */
/* overlapping strata                    */
/*****************************************/
/* Table a1 gathers the units of the 1st  */
/* stratum                               */
/*****************************************/
data a1;
input id $ pi var1 var2 var3;
cards;
1 0.2 1 4 7
2 0.4 2 6 4
3 0.8 3 7 9
4 0.2 2 3 2
5 0.7 8 6 4
6 0.1 3 5 6
7 0.5 2 6 3
8 0.3 2 1 3
9 0.4 4 7 6
10 0.5 8 2 5
;run;
/*****************************************/
```

```
/* Table a2 gathers the units of the 2nd  */
/* stratum                               */
/*****************************************/
data a2;
input id $ pi var1 var2 var3;
cards;
11 0.4 1 2 6
12 0.2 6 6 3
13 0.6 4 2 5
14 0.9 2 1 5
15 0.4 2 5 4
16 0.5 5 7 2
17 0.7 4 1 7
18 0.6 2 2 3
;
run;
/*****************************************/
/* Example                               */
/*****************************************/
/* Stratified balancing, on variables var1*/
/* var2, with pi as inclusion            */
/* probabilities, and fixed size by      */
/* stratum                               */
/*****************************************/
%echant_strat(base=work,data=a1 a2,id=id,pi=pi,contr=pi var1 var2,
              sort=ech2);
```

4.5 A numerical example

Once again, we use the population corresponding to the adresses of the city of Lyon. This city is divided into 36 strata called Iris. A 37^{th} Iris which contained very few adresses is gathered with another one.
By means of the `echant_strat` macro, we select a sample with equal inclusion probabilities ($\frac{1}{5}$), balanced on the variables quoted in Table 1 (18 variables). We thus require a sample:

- Globally balanced (on the whole city)
- Approximately balanced in each Iris
- Of fixed size in each Iris

We get a sample of 5295 units in a few seconds. Table 6 compares the sample sizes we get in each stratum with those we wanted to get. If we round the sample sizes wanted, the condition of fixed size is perfectly realized in the strata (except in the $36^{\text{ème}}$ one, to within one unit).

The estimations on the whole city are presented in Table 7. The global balancing is perfectly realized.

Table 6: Comparison between the sample sizes obtained and the sample sizes wanted by stratum

Stratum	1	2	3	4	5	6	7	8	9	10	11	12
Sample size wanted	277.8	222.8	231.4	101.4	34.6	260.4	259.2	160	128.6	20.6	268.8	285
Sample size obtained	278	223	231	102	35	260	259	160	129	21	268	285
Stratum	13	14	15	16	17	18	19	20	21	22	23	24
Sample size wanted	179.8	50.8	213.6	220.8	199	81.4	24.4	245	213.6	142.8	122.4	113
Sample size obtained	180	51	214	221	199	82	25	245	214	143	122	113
Stratum	25	26	27	28	29	30	31	32	33	34	35	36
Sample size wanted	134.4	103.6	46.6	153	157.2	114.2	71.4	102.6	55.4	155.2	124.6	18.6
Sample size obtained	134	104	46	153	157	114	71	103	55	156	125	17

Table 7: Relative difference between the real total and the Horvitz-Thompson estimator of the total for the balancing variables

Variable	Horvitz-Thompson estimator of the total	Real total	Relative difference (%)
NLOG	251 707	251 279	-0.17%
NLOGCO	243 820	243 381	-0.18%
H0019	46 446	46 395	-0.11%
H2039	75 304	75 116	-0.25%
H4059	46 166	46 078	-0.19%
H6074	20 751	20 726	-0.12%
H7599	10 450	10 435	-0.14%
F0019	46 295	46 156	-0.30%
F2039	84 142	83 957	-0.22%
F4059	51 954	51 881	-0.14%
F6074	28 634	28 637	0.01%
F7599	21 421	21 421	0.00%
ACTIFS	207 084	206 732	-0.17%
INACTIFS	224 474	224 070	-0.18%
NATFN	377 005	376 326	-0.18%
NATFA	21 886	21 833	-0.24%
NATHE	23 001	22 978	-0.10%
NATUE	9 655	9 665	0.10%

As for strata (see Table 8), only five strata present a bad balancing (the 5, 10, 19, 33, 34). Except the last one, they are small strata in which a very small sample has been drawn. As quoted before,

Table 8: Indicators of the quality of balancing by stratum for the balancing variables

Stratum	1	2	3	4	5	6	7	8	9	10	11	12
Maximum relative difference (modulus)	4%	19%	6%	4%	44%	2%	12%	13%	12%	48%	7%	3%
Average relative difference (modulus)	2%	3%	2%	2%	11%	1%	4%	4%	2%	20%	2%	1%
Stratum	13	14	15	16	17	18	19	20	21	22	23	24
Maximum relative difference (modulus)	4%	19%	7%	6%	9%	9%	24%	3%	17%	10%	22%	23%
Average relative difference (modulus)	2%	6%	2%	2%	3%	4%	13%	1%	4%	2%	6%	4%
Stratum	25	26	27	28	29	30	31	32	33	34	35	36
Maximum relative difference (modulus)	29%	16%	13%	13%	9%	27%	14%	12%	27%	33%	29%	16%
Average relative difference (modulus)	7%	3%	4%	6%	2%	8%	8%	6%	11%	16%	7%	3%

we could have perform a similar sampling with the other macro of balanced sampling, `exe_cube`. We would have drawn one sample directly in the whole population. But we would have needed the following balancing variables:

- The inclusion probability (to get a fixed sample size) and the 18 variables quoted above to obtain a global balancing. If we take into account the colinearities, that means: 17 variables
- A variable indicating the belonging to one stratum, to get a fixed sample size by stratum. That means: 35 balancing variables
- Variables equal to the product of the socio-demographic variables (18) and the variables indicating the belonging to a stratum (36) to get a stratified balancing. If we take into account the colinearities, that means: $16 \times 35 = 560$ balancing variables

For the same kind of sampling, we would have needed 612 balancing variables. The sampling would have been much slower, and the balancing would have been very badly performed in some strata.

Appendix 1: Balancing a sample and its complementary

Let U be a finite population. A sample s is said to be balanced on the variable x if

$$\sum_{k\in s}\frac{x_k}{\pi_k}=\sum_{k\in U}x_k$$

Let $\bar{s}$ be another sample, defined as the complementary of s in U. The inclusion probabilities are then $\bar{\pi_k}=\mathbb{P}(k\in\bar{s})=1-\pi_k$, and then the sample $\bar{s}$ is said to be balanced on the variable x if:

$$\sum_{k\in\bar{s}}\frac{x_k}{1-\pi_k}=\sum_{k\in U}x_k$$

The balancing of a sample s and its complementary on a variable x can be achieved by selecting a sample s balanced on variables (x_k) and $\left(\frac{x_k}{1-\pi_k}\right)$. Indeed, we get :

$$\sum_{k\in s}\frac{x_k}{\pi_k}=\sum_{k\in U}x_k$$

by definition, and:

$$\begin{array}{rcl}\sum_{k\in\bar{s}}\frac{x_k}{1-\pi_k} & = & \sum_{k\in U}\frac{x_k}{1-\pi_k}-\sum_{k\in s}\frac{x_k}{1-\pi_k}\\ & = & \sum_{k\in s}\frac{x_k}{\pi_k(1-\pi_k)}-\sum_{k\in s}\frac{x_k}{1-\pi_k}\\ & = & \sum_{k\in s}\frac{x_k}{1-\pi_k}\left(\frac{1}{\pi_k}-1\right)\\ & = & \sum_{k\in s}\frac{x_k}{\pi_k}=\sum_{k\in U}x_k\end{array}$$

Thus, $\bar{s}$ is also balanced.

Appendix 2: Stratified balancing

Let U be a finite population, stratified into H parts $U_1,\ldots,U_H$. Let π_k be the inclusion probability of unit k and $\mathbf{x_k}$ the vector of balancing variables.
We follow the process described in 3.2, and first perform a flight phase independently on each stratum (Phase 1). With the same notations as in Algorithm 1, we get at the end of Phase 1:

$$\sum_{k\in U_h}\mathbf{x_k}=\sum_{k\in U_h}\frac{\mathbf{x_k}}{\pi_k}\pi_k=\sum_{k\in U_h}\frac{\mathbf{x_k}}{\pi_k}\pi_k^*=\sum_{k\in S_h^*}\frac{\mathbf{x_k}}{\pi_k}+\sum_{k\in U_h^*}\frac{\mathbf{x_k}}{\pi_k}\pi_k^*,\ \text{for all } h=1\ldots H$$

where S_h^* denotes the units sampled in stratum U_h and U_h^* the remaining units (neither rejected nor selected, i.e. with $0<\pi_k^*<1$).

For Phase 2, we gather the remaining units and select a sample with inclusion probabilities π_k^*, balanced on variables $\frac{\mathbf{x_k}}{\pi_k}\pi_k^*$. Let $U^*=\bigcup_{h=1}^H U_h^*$, S^* the sample selected in U^* and S_h^{**} the units of S^* which belong to U_h. The balancing implies:

$$\sum_{k\in U^*}\frac{\mathbf{x_k}}{\pi_k}\pi_k^*=\sum_{k\in S^*}\frac{\mathbf{x_k}}{\pi_k}$$

The final sample S is the union of the units selected in Phase 1 and those selected in Phase 2, i.e. $S = S^* \bigcup_{h=1}^{H} S_h^*$. We have:

$$\begin{aligned}
\sum_{k\in S} \frac{\mathbf{x_k}}{\pi_k} &= \sum_{k\in S^*} \frac{\mathbf{x_k}}{\pi_k} + \sum_{h=1}^{H} \sum_{k\in S_h^*} \frac{\mathbf{x_k}}{\pi_k} \\
&= \sum_{k\in U^*} \frac{\mathbf{x_k}}{\pi_k}\pi_k^* + \sum_{h=1}^{H} \sum_{k\in S_h^*} \frac{\mathbf{x_k}}{\pi_k} \\
&= \sum_{h=1}^{H} \left[\sum_{k\in S_h^*} \frac{\mathbf{x_k}}{\pi_k} + \sum_{k\in U_h^*} \frac{\mathbf{x_k}}{\pi_k}\pi_k^*\right] \\
&= \sum_{h=1}^{H} \sum_{k\in U_h} \mathbf{x_k} = \sum_{k\in U} \mathbf{x_k}
\end{aligned}$$

i.e., the sample is globally balanced, and for each stratum h:

$$\begin{aligned}
\sum_{k\in S_h} \frac{\mathbf{x_k}}{\pi_k} &= \sum_{k\in S_h^{**}} \frac{\mathbf{x_k}}{\pi_k} + \sum_{k\in S_h^*} \frac{\mathbf{x_k}}{\pi_k} \\
&= \sum_{k\in S_h^{**}} \frac{\mathbf{x_k}}{\pi_k} + \sum_{k\in U_h} \mathbf{x_k} - \sum_{k\in U_h^*} \frac{\mathbf{x_k}}{\pi_k\pi_k^*} \\
&= \sum_{k\in U_h} \mathbf{x_k} + \sum_{k\in U_h^*} \frac{\mathbf{x_k}}{\pi_k} \left[1_{k\in S_h^{**}} - \pi_k^*\right] \simeq \sum_{k\in U_h} \mathbf{x_k}
\end{aligned}$$

We also have an approximate balancing by stratum.

References

Ardilly, P. (1991). Échantillonnage représentatif optimum à probabilités inégales. *Annales d'Économie et de Statistique*, 23:91–113.

Bertrand, P., Christian, B., Chauvet, G., and Grosbras, J.-M. (2004). Plans de sondage pour le recensement rénové de la population. In *Séries INSEE Méthodes: Actes des Journées de Méthodologie Statistique*, Paris. Paris: INSEE, to appear.

Chauvet, G. and Tillé, Y. (2004). A fast algorithm of balanced sampling. *To appear in Journal of Computational Statistics*.

Deville, J.-C., Grosbras, J.-M., and Roth, N. (1988). Efficient sampling algorithms and balanced sample. In Verlag, P., editor, *COMPSTAT, Proceeding in Computational Statistics*, pages 255–266.

Deville, J.-C. and Tillé, Y. (1998). Unequal probability sampling without replacement through a splitting method. *Biometrika*, 85:89–101.

Deville, J.-C. and Tillé, Y. (2004). Efficient balanced sampling: the cube method. *Biometrika*, 91:893–912.

Deville, J.-C. and Tillé, Y. (2005). Variance approximation under balanced sampling. *Journal of Statistical Planning and Inference*, 128:411–425.

Dumais, J. and Isnard, M. (2000). Le sondage de logements dans les grandes communes dans le cadre du recensement rénové de la population. In *Séries INSEE Méthodes: Actes des Journées de Méthodologie Statistique*, volume 100, pages 37–76, Paris. pp. 37-76, Paris: INSEE.

Rousseau, S. and Tardieu, F. (2004). La macro SAS CUBE d'échantillonnage équilibré, documentation de l'utilisateur. Technical report, INSEE.

Tillé, Y. (2001). *Théorie des sondages: échantillonnage et estimation en populations finies*. Dunod, Paris.

Tillé, Y. and Favre, A.-C. (2004). Co-ordination, combination and extension of optimal balanced samples. *Biometrika*, 91:913–927.