*Journal of Statistical Software*

*October 2014, Volume 61, Issue 4.* *http://www.jstatsoft.org/*

# SamplingStrata: An R Package for the Optimization of Stratified Sampling

**Giulio Barcaroli**
Italian National Institute of Statistics (Istat)

**Abstract**

When designing a sampling survey, usually constraints are set on the desired precision levels regarding one or more target estimates (the $Y$'s). If a sampling frame is available, containing auxiliary information related to each unit (the $X$'s), it is possible to adopt a stratified sample design. For any given stratification of the frame, in the multivariate case it is possible to solve the problem of the best allocation of units in strata, by minimizing a cost function subject to precision constraints (or, conversely, by maximizing the precision of the estimates under a given budget). The problem is to determine the best stratification in the frame, i.e., the one that ensures the overall minimal cost of the sample necessary to satisfy precision constraints. The $X$'s can be categorical or continuous; continuous ones can be transformed into categorical ones. The most detailed stratification is given by the Cartesian product of the $X$'s (the *atomic strata*). A way to determine the best stratification is to explore exhaustively the set of all possible partitions derivable by the set of atomic strata, evaluating each one by calculating the corresponding cost in terms of the sample required to satisfy precision constraints. This is unaffordable in practical situations, where the dimension of the space of the partitions can be very high. Another possible way is to explore the space of partitions with an algorithm that is particularly suitable in such situations: the *genetic algorithm*. The R package **SamplingStrata**, based on the use of a genetic algorithm, allows to determine the best stratification for a population frame, i.e., the one that ensures the minimum sample cost necessary to satisfy precision constraints, in a multivariate and multi-domain case.

*Keywords*: genetic algorithm, theory of partitions, optimal stratification, sample design, sample allocation, R package.

## 1. Introduction

Let us suppose we need to design a sample survey, having available a complete frame contain-

ing information on the target population (identifiers plus auxiliary information). If our sample design is a stratified one, we need to choose how to form strata in the population, in order to get the maximum advantage of the available auxiliary information. In other words, we have to decide how to combine the values of the auxiliary variables (henceforth, the $X$ variables) in order to determine a new categorical variable, called *stratum*. To do so, we have to take into consideration the target variables of our sample survey (henceforth, the $Y$ variables): if, to form strata, we choose the $X$ variables most correlated to the $Y$, the efficiency of the samples drawn by the resulting stratified frame may be greatly increased. In order to handle the whole auxiliary information in a homogeneous way, we have to reduce continuous data to categorical (by calculating equal frequency intervals, or using a $k$-means clustering technique, or other suitable methods). Then, for every set of candidate auxiliary variables $X$, we have to decide (i) what variables to consider as active variables in strata determination, and (ii) for each active variable, what set of values (in general, what aggregation of atomic values) have to be considered. Each choice determines a particular stratification of the target population, i.e., a possible solution to the problem of best stratification. Here, by best stratification, we mean the stratification that ensures the minimum cost of the sample, necessary to satisfy precision constraints, set on the estimates of the target variables $Y$ (constraints expressed as maximum expected coefficients of variation in different domains of interest). Therefore, the validity of a particular stratification can be measured by the associated minimum cost of a sample, whose estimates are expected to satisfy given precision levels. In general, the number of possible alternative stratifications for a given population frame may be very high, in some cases even innumerable. In these cases it is not possible to enumerate them in order to find the best stratification. Instead, adopting the *evolutionary approach*, the use of a *genetic algorithm* enables to explore the range of possible solutions in order to find a near-optimal solution after an affordable number of iterations. The implementation of the genetic algorithm in the package **SamplingStrata** (Barcaroli, Pagliuca, and Willighagen 2014) makes use of a modified version of the functions available in the **genalg** package (Willighagen 2014).

The remaining paper is organized as follows: Section 2 describes in brief the general approach followed for the implementation of the package (for a more detailed illustration of methodological aspects see Ballin and Barcaroli 2013); Section 3 illustrates how to employ the package in practical situations; Section 4 evaluates the performance of the genetic algorithm method by comparing it, in the univariate case, to the methods implemented in package **stratification** (Baillargeon and Rivest 2012a); Section 5 concludes the paper.

## 2. The general approach

In the field of stratified sampling, many studies dealing with the problem of optimizing stratification have been conducted. A general review of the proposed methods is contained in Gonzales and Eltinge (2010). Basically, the optimization can be conceived as based on four distinct components:

1. objective function;
2. constraints;
3. input parameters;
4. decision variables.

The *total cost* associated with a specific sample (depending on the allocation of units in the strata, and on the *per unit interviewing cost*, that may vary from stratum to stratum), and the *expected precision* related to each target estimate, can be associated to the first two components in an interchangeable way: in fact, it is possible to minimize the total cost under a set of precision constraints, or to maximize precision levels under given budget constraints. In both cases, optimization is performed on the basis of input parameters such as the *variances of the target variables* and the *number of population units* in each stratum. While in general it is not difficult to assign population units to the different strata, as this only depends on the auxiliary information in the frame (which is available by definition), much more complex is to get the information necessary to estimate the variability of target variables in each stratum. In fact, this information is not available at unit level (otherwise we would not carry out a specific survey on these variables), so it is necessary to estimate their variability by harnessing different possible sources:

1. census data;
2. data from previous rounds of the same survey;
3. data on proxy variables.

In the first case, the lower the time gap, the higher the reliability of the estimate. In the second, together with the time gap also the sampling errors on the estimates should be taken into account. Finally, in the third case also the correlation between target and proxy variables must be considered when evaluating the quality of the estimates. An effort should be made to model the relationships between target variables and all the available information, including the auxiliary one in the sampling frame, in order to increase this quality. Henceforth we assume that estimates of acceptable reliability of the variability of target variables in strata are available. Should this assumption not apply, the method here proposed would not be applicable.

A first important distinction can be made between the case where the precision of only one target variable is taken into consideration in the objective function or in the constraints (univariate case), or when more than one of them are considered (multivariate case). A further complication may be given by the necessity to consider different domains to which estimates (and related precision levels) have to be referred to (multi-domain case).

A second, more important, distinction is related to the decision variables. Many optimization methods are based on decision variables that state how many population units have to be selected in each stratum: in other words, strata in the population are assumed as given, and the optimization consists in determining the best allocation of sampling units in population strata. Under this setting, the optimization problem can be solved using an application of the Cauchy-Schwarz inequality (Cochran 1977) or Lagrangian multipliers (Varberg and Purcell 1997). Well known solutions in the multivariate case are the ones given by Bethel (1989) and Chromy (1987).

However, the way the population is stratified is of the greatest importance with respect to the optimization of the sample design. The relationships between the survey target variables and the stratification variables are at the basis of the stratified sampling. In order to take maximum advantage of these relationships, choices regarding the way we define population strata should enter into the optimization process together with the allocation choices. Up until

recently, on the contrary, optimization of stratified sampling has been considered as a two-step process: first, a stratification is chosen, by exploiting all the auxiliary information available on sampling units, or only a subset, selected on the basis of known correlations between target and stratification variables; then, given the chosen stratification, the problem of allocation is solved (Dalenius and Hodges 1959). The Lavallée and Hidiroglou method for the stratification of a population which is skewed with respect to a unique stratification variable (Lavallée and Hidiroglou 1988) can be considered an exception, as it allows to determine both, strata boundaries and best allocation, but only in the univariate case. Also the method proposed by Keskinturk and Er (2007), which makes use of the genetic algorithm approach, suffers from the same limitation.

The approach, which is implemented in the R (R Core Team 2014) package **SamplingStrata** available from the Comprehensive R Archive Network (CRAN) at `http://CRAN.R-project.org/package=SamplingStrata`, permits a joint optimization of both population stratification and sampling units allocation, in the multivariate and multi-domain case. It is based on the following assumptions:

1. the optimal stratification of a population frame depends on the particular sample survey that has to be planned;

2. the optimality of the solution can be measured against its cost, expressed in terms of the number of units to be sampled (together with the per unit interview cost), required to satisfy precision constraints, set on estimates of totals or means of the target variables;

3. the multivariate and multi-domain case must be contemplated in order to ensure generality;

4. the availability of auxiliary information in the population frame permits to define a space of alternative stratifications: this space should be rigorously generated and, in principle, the best solution could be found by exhaustively evaluating each stratification with regard to the cost of the associated sample;

5. as in practical situations it is not possible to enumerate the space of stratifications (because of its dimension), a heuristic is necessary in order to explore this space without (or with a negligible) loss of optimality: from this point of view genetic algorithms have been proven to be particularly efficient.

## 2.1. Best allocation for a given stratification

In this section we briefly recall the approach proposed by Bethel (1989) in order to find the minimum cost and the best allocation of a sample in given strata, in the multivariate case. Given $H$ strata, let $N_h$ and $S^2_{h,g}$ (with $h = 1, \ldots, H$ and $g = 1, \ldots, G$) be respectively the population and the variances of the $G$ different target variables $Y$'s in each stratum $h$. Assuming a simple random sampling of $n_h$ units without replacement in each stratum, the variance of the Horvitz-Thompson estimator of the total ($\hat{Y}_g$) is given by

$$\mathsf{VAR}(\hat{Y}_g) = \sum_{h=1}^{H} N_h^2 \left(1 - \frac{n_h}{N_h}\right) \cdot \frac{S^2_{h,g}}{n_h}, \qquad g = 1, \ldots, G. \tag{1}$$

Let us consider now the following cost function:

$$C(n_1, \ldots, n_H) = C_0 + \sum_{h=1}^{H} C_h n_h. \tag{2}$$

In this function we can distinguish a fixed component ($C_0$, not dependent on sample size and its allocation), and a variable one (the sum of the products between the cost for interviewing one unit in the stratum ($C_h$) and the allocation of units in that stratum ($n_h$)). If we define upper limits on the expected sampling variances defined by Equation 1 by setting

$$\mathsf{VAR}(\hat{Y}_g) \leq U_g, \tag{3}$$

then to find the best solution to the problem of sample allocation in a stratified design is equivalent to finding the vector of the allocations $n_h$ that minimizes Equation 2 given Equation 3.

An algorithm that is proved to converge to the solution was provided by (Bethel 1989). The package **SamplingStrata** provides the function `bethel` that allows to determine the best allocation given two different inputs:

1. information on the distribution of the $Y$'s in the strata;
2. precision constraints on the $Y$'s.

In this particular implementation, precision constraints are expressed in terms of maximum expected coefficient of variation (CV) for each $Y$:

$$CV(\hat{Y}_g) = \frac{\sqrt{\mathsf{VAR}(\hat{Y}_g)}}{\mathsf{E}(\hat{Y}_g)} \tag{4}$$

By so doing, we are able to remove the dependence on the scale (or range) of the values associated with the different $Y$'s.

So the problem becomes:

$$\begin{cases} C_0 + \sum_{h=1}^{H} C_h n_h \rightarrow \min \\ CV(\hat{Y}_1) \leq U_1 \\ CV(\hat{Y}_2) \leq U_2 \\ \ldots \\ CV(\hat{Y}_G) \leq U_G \end{cases} \tag{5}$$

The Bethel algorithm implemented in the `bethel` function can be used not only for its original purpose (to determine the best allocation on the basis of the precision constraints), but also to evaluate any given stratification of the population frame. In other words, given two different stratifications adopted for the same population frame, we should prefer the one for which the solution identified by the Bethel algorithm has a smaller cost.

## 2.2. Space of stratifications for a given frame

Let us consider a sampling frame, that is a set of $N$ records containing information related to $N$ units belonging to the reference population. The available information can be grouped into two distinct sets of variables:

- the variables allowing the identification of units, in order to be able to contact them while carrying out the survey;
- the variables useful to optimize the sample design (the *auxiliary variables* $X$).

In our setting, we assume that:

1. a set of $M$ auxiliary variables $X_m$ $(m = 1, \ldots, M)$ are available;
2. only categorical (nominal or ordinal) auxiliary variables are considered: when continuous variables are present in the set, they have to be converted into categorical ones by means of a transformation algorithm (for example, by applying a $k$-means clustering algorithm, see Hartigan and Wong 1979);
3. with each (categorical) auxiliary variable we can associate a domain set given by the vector $d_m = \{x_1, \ldots, x_{k_m}\}$, where an integer value is assigned to each value in the domain.

Under these assumptions, the most detailed stratification for the frame is given by considering the Cartesian product $CP = X_1 \times X_2 \times \ldots \times X_M$.

The maximum number of strata is equal to $K = \prod_{m=1}^{M} k_m - I^*$, where $I^*$ is the number of non-valid or missing combinations in the frame.

We call *atomic strata* the result of the Cartesian product, i.e., the strata obtained by cross-classifying the units using all the values of all the auxiliary variables, and indicate the corresponding set as $L = \{l_1, l_2, \ldots, l_K\}$.

Starting from this set of atomic strata, it is possible to derive the set of all possible partitions $P_1, P_2, \ldots, P_B$, where each partition is defined as a collection of sets $T_1, T_2, \ldots, T_q$ $(1 \leq q \leq K)$ where each $T_i$ is a subset of $L$.

Accordingly to the *theory of partitions* (Hankin and West 2007) the following conditions must hold:

$$\begin{cases} T_i \cap T_j = \emptyset \quad \text{if } i \neq j, \\ \bigcup_{i=1}^{q} T_i = L, \\ T_i \neq \emptyset \quad \text{for } i = 1, \ldots, q. \end{cases} \tag{6}$$

Each partition is equivalent to a given stratification of the frame. The set of all possible partitions can be considered as the *space of stratifications*.

For example, let us consider a set of atomic strata $L = \{l_1, l_2, l_3\}$. The different partitions that can be generated by this set are:

$$\begin{aligned} P_1 &= \{(l_1, l_2, l_3)\}, & P_2 &= \{(l_1), (l_2, l_3)\}, \\ P_3 &= \{(l_2), (l_1, l_3)\}, & P_4 &= \{(l_3), (l_1, l_2)\}, \\ P_5 &= \{(l_1), (l_2), (l_3)\}. \end{aligned}$$

The cardinality of the set of all the possible partitions is given by the *Bell number*:

$$B_K = \sum_{i=0}^{K-1} \binom{n}{i} \cdot B_i, \qquad (B_0 = 1). \tag{7}$$

### 2.3. Choosing the best stratification by applying the genetic algorithm

In principle, having fixed precision constraints on a set of target estimates for a given sample survey, it is possible to choose the best stratification for a population frame where a set of auxiliary variables are available, by executing the following steps:

1. determine the set of atomic strata by cross-classifying sampling units using all the values of all auxiliary variables;
2. calculate distributional parameters (mean and variance) of target variables if information related to $Y$'s is available for each unit in the frame (or, if not, by using proxy information by other sources) in atomic strata;
3. solve the allocation problem for the atomic strata and associate the cost of the solution found;
4. generate all possible partitions from the set of atomic strata (in order to generate all the possible partitions it is possible to use the package **partitions**; Hankin 2013);
5. for each generated partition:
   - calculate distributional parameters (mean and variance) of target variables in current strata by aggregating the corresponding information available in atomic strata;
   - solve the allocation problem for the current partition and associate the cost of the solution;
6. choose the best stratification as the one given by the partition with the minimal associated cost.

Unfortunately, this procedure, that is based on an exhaustive enumeration of all possible partitions, is in most cases not feasible, as the number of partitions to be evaluated is too high. In fact, considering the formula for the calculation of the Bell number reported in Equation 7, this number grows very rapidly with regard to the dimension of the set of atomic elements (for example, $B_3 = 5$, $B_4 = 15$, $B_{10} = 115{,}975$ and $B_{100} \sim 4.76 \times 10^{115}$).

The function `optimizeStrata` available in package **SamplingStrata** allows to explore the space of stratifications without being obliged to exhaustively enumerate it, by using a search technique known as the *genetic algorithm*.

Genetic algorithms belong to the class of *evolutionary algorithms* that make use of techniques based on concepts derived by biology, such as *inheritance*, *mutation*, *crossover*, *fitness* and *selection* (DeJong 2006).

In order to apply the genetic algorithm to the problem of finding the best stratification, the following setting has been adopted:

1. a given *stratification* is considered as an *individual* in a *population* (or *generation* of individuals);
2. an *individual* is characterized by a *genome* that is optimized in the course of the *evolution*;

3. the *genome* is represented by a vector whose dimension is given by the number of atomic strata ($K$): with each position in this vector an atomic stratum is associated;

4. to each element in the vector an integer value lying in the interval $[1, K]$ is assigned randomly: atomic strata that share the same integer value, collapse in an *aggregate stratum*;

5. the *fitness* of each *individual* is evaluated by solving the system reported in Equation 5 (using the Bethel algorithm);

6. in the passage from one generation to the next, the fittest individuals are privileged: a percentage of those with highest fitness are directly moved to the next generation, the others are randomly selected with probability proportional to their fitness, in order to let them procreate *children*;

7. each child is procreated by applying *crossover* to their parents (a swap of the genes contained in the two genomes), and applying *mutation* to the resulting genome.

At the end of the evolution (the chain of generations), the individual with the absolute best fitness will be chosen: the genome of this individual represents a stratification in which all or some of the atomic strata have been aggregated.

## 3. A general procedure for the use of package SamplingStrata

The optimization of the sampling design starts by considering the available population frame, defining the target estimates of the survey and establishing precision constraints on them. It is then possible to determine the best stratification and the optimal allocation. Finally, the sample can be drawn from the frame stratified accordingly to the optimal stratification. Formally, these are the required steps:

1. analysis of the frame data: identification of available auxiliary information;

2. manipulation of auxiliary information: in case auxiliary variables are of continuous type, they have to be transformed into categorical variables;

3. construction of atomic strata: on the basis of the categorical auxiliary variables available in the sampling frame, the set of atomic strata can be obtained by cross-classifying the units by using all the values of all the auxiliary variables;

4. characterization of each atomic stratum with the information related to the target variables (mean and standard deviation for each $Y$, estimated by using available information: by census, previous surveys or proxy variables data);

5. choice of the precision constraints for each target estimate, possibly differentiated by domain;

6. optimization of stratification and determination of required sample size and allocation;

7. analysis of the resulting optimized strata;

8. association of new labels to sampling frame units, each of them indicating the new strata resulting from the optimal aggregation of the atomic strata;
9. selection of units from the sampling frame with a *stratified random sample* selection scheme;
10. analysis of the solution found.

In the following, we will illustrate each step starting from a real sampling frame, the data frame `swissmunicipalities` that is available in the R package **sampling** (Tillé and Matei 2012).

### 3.1. Analysis of the frame data and manipulation of auxiliary information

As a first step, we have to define a *frame* data frame containing the following information:

- a unique identifier of the unit (no restriction on the name, for instance *id*);
- the (optional) identifier of a pre-defined stratum to which the unit belongs;
- the values of $M$ auxiliary variables (named from X1 to XM);
- the (optional) values of $G$ target variables (named from Y1 to YG);
- the values of the domain of interest for which we want to produce estimates (named *domainvalue*).

By executing the following statements in the R environment:

```
R> data("swissmunicipalities", package = "sampling")
```

we get the `swissmunicipalities` data frame, that contains 2,896 observations (each observation refers to a Swiss municipality in 2003). Among the others, we can find the following variables:

- `REG`: Swiss region,
- `Nom`: municipality name,
- `Surfacesbois`: wood area,
- `Surfacescult`: area under cultivation,
- `Alp`: mountain pasture area,
- `Airbat`: area with buildings,
- `Airind`: industrial area,
- `Pop020`: number of men and women aged between 0 and 19,
- `Pop2040`: number of men and women aged between 20 and 39,

- `Pop4065`: number of men and women aged between 40 and 64,
- `Pop65P`: number of men and women aged 65 and over,
- `POPTOT`: total population.

First, we define the identifier of the frame:

```
R> frame <- NULL
R> frame$id <- swissmunicipalities$Nom
```

Let us suppose to plan a survey whose target estimates are the totals of the population by age class in each Swiss region. In this case, the $Y$'s variables will be:

- `Y1`: number of men and women aged between 0 and 19,
- `Y2`: number of men and women aged between 20 and 39,
- `Y3`: number of men and women aged between 40 and 64,
- `Y4`: number of men and women aged 65 and over.

Consequently, the following statements are executed:

```
R> frame$Y1 <- swissmunicipalities$Pop020
R> frame$Y2 <- swissmunicipalities$Pop2040
R> frame$Y3 <- swissmunicipalities$Pop4065
R> frame$Y4 <- swissmunicipalities$Pop65P
```

We suppose that the values of these variables in the frame have been obtained from past surveys (for instance, from a census), or from administrative data: it should always be taken into account that they could be out of date, or not completely reliable.

As for the auxiliary variables ($X$'s), we can use all those characterizing the area use (*surfaces* pertaining *wood, mountain or pasture, cultivations, industries, buildings*). As these variables are of continuous type, we have to transform them into categorical (ordinal) form first. In order to do that, it is possible to apply a $k$-means clustering method by using the function `var.bin`:

```
R> library("SamplingStrata")
R> set.seed(1508)
R> frame$X1 <- var.bin(swissmunicipalities$POPTOT, bins = 18)
R> frame$X2 <- var.bin(swissmunicipalities$Surfacesbois, bins = 3)
R> frame$X3 <- var.bin(swissmunicipalities$Surfacescult, bins = 3)
R> frame$X4 <- var.bin(swissmunicipalities$Alp, bins = 3)
R> frame$X5 <- var.bin(swissmunicipalities$Airbat, bins = 3)
R> frame$X6 <- var.bin(swissmunicipalities$Airind, bins = 3)
```

Now, we have six different auxiliary variables of categorical type, the first with 18 different modalities, the others with 3 modalities.

Finally, we have to set the values of the *domainvalue* variable, which is mandatory. As we want to obtain estimates for each one of the seven regions, we set:

```
R> frame$domainvalue <- swissmunicipalities$REG
R> frame <- as.data.frame(frame)
```

Now, the `frame` data frame looks like:

```
R> head(frame)
```

```
          id    Y1     Y2     Y3    Y4 X1 X2 X3 X4 X5 X6 domainvalue
1     Zurich 57324 131422 108178 66349 18  3  2  1  3  3           4
2     Geneve 32429  60074  57063 28398 17  1  1  1  3  2           1
3      Basel 28161  50349  53734 34314 17  1  1  1  3  3           3
4       Bern 19399  44263  39397 25575 17  2  3  1  3  3           2
5   Lausanne 24291  44202  35421 21000 17  2  2  1  3  2           1
6 Winterthur 18942  28958  27696 14887 16  3  3  1  3  3           4
```

This is the format required by the package.

### 3.2. Construction of atomic strata

The `strata` data frame reports information regarding each stratum in the population. There is one row for each stratum. The total number of strata is given by the number of different combinations of $X$ values in the frame. For each stratum, the following information is required:

1. the identifier of the stratum (named `STRATO`), concatenation of the values of the variables $X$'s;
2. the values of the $M$ auxiliary variables (named `X1` to `X`$M$) corresponding to those in the frame;
3. the total number of units in the population belonging to the stratum (named $N$);
4. a flag (named `CENS`) indicating if the stratum is to be censused (= 1) or sampled (= 0);
5. a variable indicating the cost of interviewing a single unit in the stratum (named `COST`);
6. for each target variable $Y$, its estimated mean and standard deviation (named respectively `M`$i$ and `S`$i$);
7. the value of the domain of interest to which the stratum belongs (named `DOM1` and corresponding to the variable `domainvalue` in the `frame` data frame).

If in the `frame` data frame the values of the target $Y$'s variables (from a census, or from administrative data) are also present, it is possible to automatically generate the `strata` data frame by invoking the `buildStrataDF` function. Let us consider again the `frame` data frame that we have built in previous steps. We can apply to it the `buildStrataDF` function:

```
R> strata <- buildStrataDF(frame)
```

```
Computations have been done on population data
```

This is the structure of the created data frame:

```
R> str(strata)

'data.frame':        641 obs. of  19 variables:
 $ STRATO: Factor w/ 295 levels "10*1*1*1*1*1",..: 1 2 3 6 9 11 15 19 21 22 ...
 $ N     : int  1 1 3 1 1 1 1 184 1 4 ...
 $ M1    : num  1686 1980 2006 1800 1724 ...
 $ M2    : num  2174 2379 2199 2001 2089 ...
 $ M3    : num  2397 2474 2847 3045 2261 ...
 $ M4    : num  964 665 1122 1424 1220 ...
 $ S1    : num  0 0 277 0 0 ...
 $ S2    : num  0 0 88.2 0 0 ...
 $ S3    : num  0 0 132 0 0 ...
 $ S4    : num  0 0 121 0 0 ...
 $ COST  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ CENS  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ DOM1  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ X1    : int  10 10 10 10 10 10 10 1 1 11 ...
 $ X2    : int  1 1 1 1 1 2 2 1 1 1 ...
 $ X3    : int  1 1 1 2 3 1 2 1 1 1 ...
 $ X4    : int  1 1 1 1 1 2 1 1 1 1 ...
 $ X5    : int  1 1 2 2 2 2 2 1 1 2 ...
 $ X6    : int  1 2 1 1 2 1 3 1 2 1 ...
```

It is worth noting that the total number of different atomic strata is lower than the expected dimension of the Cartesian product of the $X$'s (which is 4,374): this is due to the fact that not all combinations of the value of the auxiliary variables are present in the sampling frame. Variables `COST` and `CENS` are initialized to 1 and 0, respectively, for all strata. It is possible to give them different values:

1. for variable `COST`, it is possible to differentiate the cost of interviewing per unit by assigning real values;
2. for variable `CENS`, it is possible to set it equal to 1 for all strata that are of the 'take-all' type (i.e., all units in those strata must be selected).

On the contrary, if there is no information in the frame regarding the target variables, it is necessary to build the strata data frame starting from other sources, for instance a previous round of the same survey, or from other surveys.

### 3.3. Choice of the precision constraints for each target estimate

The `cv` data frame contains precision constraints that are set on target estimates. This means to define a maximum coefficient of variation for each variable and for each domain value. Each row of this frame is related to precision constraints in a particular domain of interest, identified by the `DOM1` value. In the case of the Swiss municipalities, we have chosen to define the following constraints:

```
R> cv <- data.frame(DOM = "DOM1", CV1 = 0.05, CV2 = 0.05, CV3 = 0.05,
+    CV4 = 0.05, domainvalue = 1:7)
R> cv

   DOM  CV1  CV2  CV3  CV4 domainvalue
1 DOM1 0.05 0.05 0.05 0.05           1
2 DOM1 0.05 0.05 0.05 0.05           2
3 DOM1 0.05 0.05 0.05 0.05           3
4 DOM1 0.05 0.05 0.05 0.05           4
5 DOM1 0.05 0.05 0.05 0.05           5
6 DOM1 0.05 0.05 0.05 0.05           6
7 DOM1 0.05 0.05 0.05 0.05           7
```

In this way, we have set a maximum of 5% to the coefficients of variation expected for variables `Y1`, `Y2`, `Y3` and `Y4`, in each of the 7 different domains (Swiss regions) in domain level `DOM1`. Of course we could differentiate the precision constraints region by region. It is important to underline that the values of `domainvalue` are the same than those in the `frame` data frame, and correspond to the values of variable `DOM1` in the `strata` data frame.

If we want to determine the total size of the sample required to satisfy these precision constraints, considering the current stratification of the frame (the 641 atomic strata), we can do this by simply using the function `bethel` (it is worth noting that the format of the constraints data frame for the `bethel` function is different from the one accepted by the `optimizeStrata` function, as in `bethel` it is not possible to differentiate precision levels in the various subdomains) :

```
R> errors <- cv[1, 1:5]
R> allocation <- bethel(strata, errors)
R> length(allocation)

[1] 641

R> sum(allocation)

[1] 893
```

This is the required amount of units to be sampled when the frame stratification is most detailed. In general, after the optimization, this number is greatly reduced.

### 3.4. Optimization of frame stratification

Once the `cv` and the `strata` data frames have been prepared, it is possible to apply the function that optimizes the stratification of the frame, that is `optimizeStrata`. This function operates on all subdomains, identifying the best solution for each one of them. Among the parameters to be passed to `optimizeStrata`, the most important are:

1. `cv`: the (mandatory) data frame containing the precision levels expressed in terms of maximum acceptable coefficients of variation that refer to the estimates on target variables $Y$'s of the survey;

2. `strata`: the (mandatory) data frame containing the information related to atomic strata;

3. `initialStrata`: the initial upper limit on the number of strata for each solution. Default value is `nrow(strata)`, i.e., the number of atomic strata;

4. `minnumstr`: the minimum number of units that must be allocated in each stratum. Default is `2`, that is the minimum value necessary to calculate sampling variance;

5. `iter`: the number of iterations (= generations) to be performed by the algorithm. Default is `20`;

6. `pops`: the dimension of each generation in terms of individuals. Default is `50`;

7. `mut_chance` (mutation chance): for each new individual, the probability that the value of a given chromosome (i.e., one bit in the solution vector), is changed. Default is `0.05`;

8. `elitism_rate`: this parameter indicates the rate of fittest solutions that must be transferred from one generation to another. Default is `0.2`.

In the case of the Swiss municipalities, `optimizeStrata` is performed with the following values for the parameters:

```
R> solution <- optimizeStrata(errors = cv, strata = strata, cens = NULL,
+    strcens = FALSE, initialStrata = nrow(strata), addStrataFactor = 0.00,
+    minnumstr = 2, iter = 400, pops = 20, mut_chance = 0.005,
+    elitism_rate = 0.2, highvalue = 1e+08, suggestions = NULL,
+    realAllocation = TRUE, writeFiles = TRUE)

Input data have been checked and are compliant with requirements

GA Settings
  Population size       = 20
  Number of Generations = 400
  Elitism               = 4
  Mutation Chance       = 0.005
```

The results of the execution are contained in the list `solution`, composed by two elements:

1. `solution$indices`: the vector of the indices that indicates to what aggregated stratum each atomic stratum belongs;

2. `solution$aggr_strata`: the data frame containing information on the optimal aggregated strata.

As we have set to '1' the cost of interviewing a unit in each atomic stratum, the cost of the best solution is given by the total size of the sample required to satisfy precision constraints. In our case:

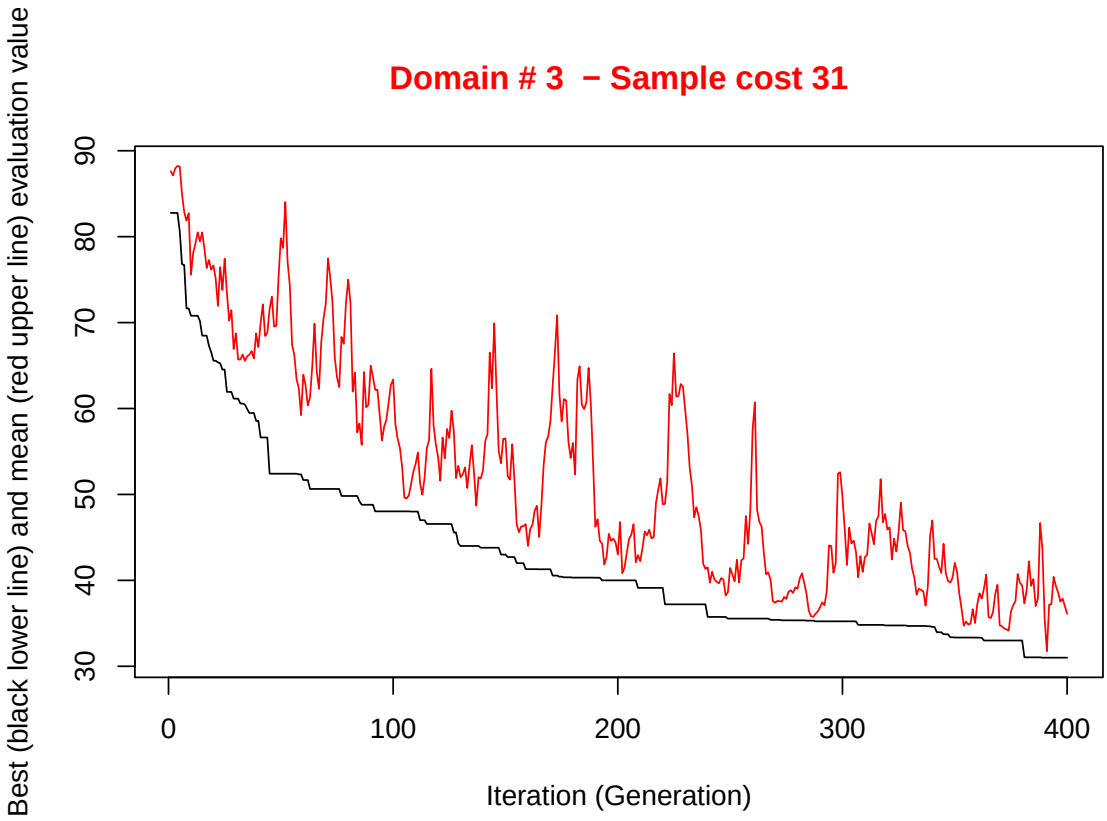


Figure 1: This graph illustrates the convergence of the solution in the course of the iterations. Along the $x$-axis the executed iterations are reported, from 1 to the maximum, while on the $y$-axis the cost of the sample required to satisfy precision constraints is reported. The upper (red) line represents the average sample size in each iteration, while the lower (black) line indicates the best solution found up to the $i$th iteration.

```
R> sum(ceiling(solution$aggr_strata$SOLUZ))

[1] 365
```

It can be seen that there has been a noticeable reduction in the size of the sample, compared to the solution found in the case of atomic strata.

The execution of `optimizeStrata` implies an independent optimization in each one of the 7 different domains (regions): the optimization run for region 3 is reported in Figure 1.

### 3.5. Analysis of results

A given solution represents an aggregation of the atomic strata. In order to analyze how atomic strata have been aggregated, it is possible to apply the function `updateStrata`, that assigns the labels of the new strata to the initial ones in the data frame `strata`, and produces:

1. a new file named 'newstrata.txt' containing all the information in the `strata` data frame

related to atomic strata, plus a label indicating to which new stratum a given atomic stratum belongs;

2. a table, contained in the file 'strata_aggregation.txt', showing in which way the auxiliary variables $X$ determine the new strata.

The function is invoked in this way:

```
R> newstrata <- updateStrata(strata, solution, writeFiles = TRUE)
R> head(newstrata)
```

```
        STRATO N       M1       M2       M3       M4       S1       S2       S3
1 10*1*1*1*1*1 1 1686.000 2174.000 2397.000  964.000   0.0000  0.00000   0.0000
2 10*1*1*1*1*2 1 1980.000 2379.000 2474.000  665.000   0.0000  0.00000   0.0000
3 10*1*1*1*2*1 3 2006.333 2198.667 2847.333 1121.667 277.4074 88.21313 131.6215
4 10*1*2*1*2*1 1 1800.000 2001.000 3045.000 1424.000   0.0000  0.00000   0.0000
5 10*1*3*1*2*2 1 1724.000 2089.000 2261.000 1220.000   0.0000  0.00000   0.0000
6 10*2*1*2*2*1 1 1811.000 2013.000 2488.000 1203.000   0.0000  0.00000   0.0000
       S4 COST CENS DOM1 X1 X2 X3 X4 X5 X6 LABEL      STRATUM
1   0.000    1    0    1 10  1  1  1  1  1    11 10*1*1*1*1*1
2   0.000    1    0    1 10  1  1  1  1  2    16 10*1*1*1*1*2
3 121.206    1    0    1 10  1  1  1  2  1     4 10*1*1*1*2*1
4   0.000    1    0    1 10  1  2  1  2  1     6 10*1*2*1*2*1
5   0.000    1    0    1 10  1  3  1  2  2     4 10*1*3*1*2*2
6   0.000    1    0    1 10  2  1  2  2  1     4 10*2*1*2*2*1
```

Now, the atomic strata are associated with the aggregate strata defined in the optimal solution, by means of the variable `LABEL`. If we want to analyze in detail the new structure of the stratification, we can look at the '`strata_aggregation.txt`' file:

```
R> strata_aggregation <- read.delim("strata_aggregation.txt")
R> head(strata_aggregation)
```

```
  DOM1 AGGR_STRATUM X1 X2 X3 X4 X5 X6
1    1            1  4  1  1  2  1  1
2    1            1  6  1  1  1  2  1
3    1            1  6  2  1  1  1  1
4    1            1  6  2  2  2  2  1
5    1            1  7  1  1  1  1  2
6    1            1  9  3  2  2  2  2
```

In this structure, for each aggregate stratum the values of the $X$'s variables in each contributing atomic stratum are reported. It is then possible to understand the meaning of each aggregate stratum produced by the optimization.

### 3.6. Updating the frame and selecting the sample

Once the optimal stratification has been obtained, to be operational we need to accomplish the following two steps:

1. to update the frame units with new stratum labels (combination of the new values of the auxiliary variables $X$);

2. to select the sample from the frame stratified accordingly to the solution found.

To do the first, we execute the following command:

```
R> framenew <- updateFrame(frame, newstrata, writeFiles = TRUE)
```

The function `updateFrame` receives, as arguments, the indication of the data frame in which the frame information is saved, and of the data frame produced by the execution of the `updateStrata` function. The execution of this function produces a data frame (`framenew`), and also a file (named 'framenew.txt') containing, for each unit, the label indicating to which aggregated stratum the unit belongs. The allocation of units is contained in the `SOLUZ` variable in the data frame `solution$aggr_strata`. It is now possible to select the sample from this new version of the frame:

```
R> sample <- selectSample(framenew, solution$aggr_strata, writeFiles = TRUE)

*** Sample has been drawn successfully ***
 365  units have been selected from  183  strata

==> There have been  33  take-all strata
from which have been selected  43 units
```

The function `selectSample` produces two datasets:

1. 'sample.csv' containing the units of the frame that have been selected, together with the weights that have been calculated for each one of them;

2. 'sample.chk.csv' containing information on the selection: for each stratum, the number of units in the population, the planned sample, the number of selected units, the sum of their weights (that must equalise the number of units in the population).

### 3.7. Evaluation of the found solution

In order to be confident about the quality of the found solution, the function `evalSolution` allows to run a simulation, based on the selection of a desired number of samples from the frame to which the stratification, identified as the best, has been applied.

The user can invoke this function also indicating the number of samples to be drawn:

```
R> evalSolution(framenew, solution$aggr_strata, nsampl = 1000,
+    writeFiles = TRUE)
```

For each drawn sample, the estimates related to the $Y$'s are calculated. Their means and standard deviations are also computed, in order to produce the CV related to each variable in every domain. These CV's are written to an external CSV file:

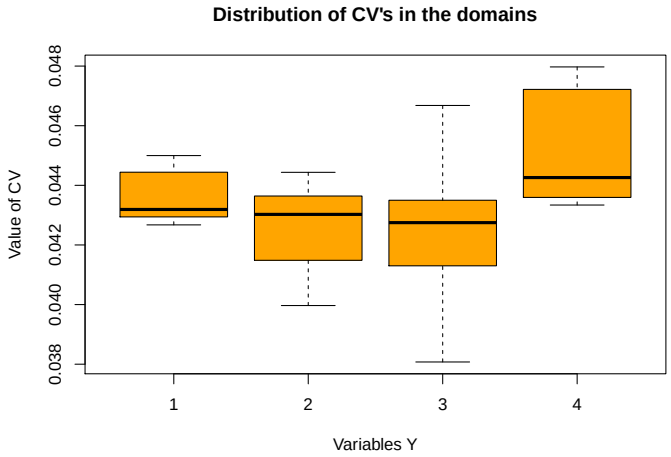


Figure 2: Distribution of expected CV's in the different domains for each target variable.

```
R> expected_cv <- read.csv("expected_cv.csv")
R> expected_cv
```

```
        CV1        CV2        CV3        CV4  dom
1 0.04456437 0.03996783 0.04139047 0.04374294 DOM1
2 0.04319048 0.04165655 0.04408840 0.04333896 DOM2
3 0.04303537 0.04375964 0.04291049 0.04793896 DOM3
4 0.04499780 0.04443586 0.04667874 0.04649659 DOM4
5 0.04284689 0.04302516 0.04120950 0.04344615 DOM5
6 0.04267311 0.04352299 0.04274853 0.04426018 DOM6
7 0.04431571 0.04131362 0.03807466 0.04797181 DOM7
```

These values are on average compliant with the precision constraints set (see also Figure 2).

## 4. A comparative assessment of the GA method

The R package **stratification** (Baillargeon and Rivest 2012b) implements various methods in order to choose strata boundaries and determine the best allocation in the univariate case, i.e., in the case of only one stratification variable (and only one target variable, that can be coincident with the stratification one), having set the number of strata. These methods are:

- the cumulative root frequency method, proposed by Cochran (1977);
- the geometric stratification method, introduced by Gunning and Horgan (2004);
- the Lavallée and Hidiroglou (LH) stratification method (Lavallée and Hidiroglou 1988), with two different optimization algorithms: Sethi's algorithm (Sethi 1963) and Kozak's random search algorithm (Kozak 2004).

The package contains a number of datasets, namely `Debtors`, `UScities`, `UScolleges` and `USbanks`. It is possible to apply to each dataset the different methods available in package

| Method | Sample size |
|---|---|
| Geometric | 109 |
| Cumulative Root Frequency | 113 |
| LH (Sethi algorithm) | 107 |
| LH (Kozak algorithm) | 91 |

Table 1: Sample sizes obtained by different methods applied to `USbanks` dataset.

**stratification**, together with the genetic algorithm in package **SamplingStrata**, and verify the performance of each of them. The goal is to minimize the total sample size $n$ under a given constraint on the CV of the unique target variable. For instance, considering the dataset `USbanks`,

```
R> library("stratification")
R> data("USbanks", package = "stratification")
R> LHkozak <- strata.LH(x = USbanks, CV = 0.01, Ls = 5,
+    alloc = c(0.5, 0, 0.5), takeall = 0, algo = "Kozak")
R> LHkozak

Given arguments:
x = USbanks
CV = 0.01, Ls = 5, takenone = 0, takeall = 0
allocation: q1 = 0.5, q2 = 0, q3 = 0.5
model = none
algo = Kozak: minsol = 1000, idopti = nh, minNh = 2, maxiter = 10000,
              maxstep = 20, maxstill = 200, rep = 5, trymany = TRUE

Strata information:
          |      type rh |    bh   E(Y)   Var(Y)  Nh nh   fh
stratum 1 | take-some  1 | 115.5  93.05   177.82 110 10 0.09
stratum 2 | take-some  1 | 169.5 139.25   173.85 101  9 0.09
stratum 3 | take-some  1 | 257.5 208.19   429.85  54  8 0.15
stratum 4 | take-some  1 | 376.5 312.94   860.51  35  7 0.20
stratum 5 |  take-all  1 | 978.0 597.42 34630.14  57 57 1.00
Total                                          357 91 0.25

Total sample size: 91
Anticipated population mean: 225.6246
Anticipated CV: 0.009846464
Note: CV=RRMSE (Relative Root Mean Squared Error) because takenone=0.
```

The application of the LH method (making use of Kozak's algorithm) yields a total sample size of 91. It is the best solution compared to the others obtained by applying the geometric and the cumulate root frequency. The results obtained by the different methods are reported in Table 1.

We now apply the genetic algorithm to the same problem:

```
R> frame <- data.frame(Y1 = USbanks,
+    X1 = rep(1:length(unique(USbanks)), table(USbanks)),
+    domainvalue = rep(1, length(USBanks)))
R> frame$id <- row.names(frame)
R> strata <- buildStrataDF(frame)
R> strata <- strata[order(strata$X1), ]
R> cv <- data.frame(DOM = "DOM1", CV1 = 0.01, domainvalue = 1)
R> pop <- 20
R> z <- cut(strata$M1,quantile(strata$M1, probs = seq(0, 1, 0.2)),
+    label = FALSE, include.lowest = TRUE)
R> v <- matrix(z, nrow = pop - 1, ncol = nrow(strata), byrow = TRUE)
R> solution <- optimizeStrata(cv, strata, cens = NULL, strcens = FALSE,
+    alldomains = TRUE, dom = NULL, initialStrata = 5, addStrataFactor = 0.0,
+    minnumstr = 2, iter = 10000, pops = pop, mut_chance = 0.0005,
+    elitism_rate = 0.2, highvalue = 1e+08, suggestions = v,
+    realAllocation = TRUE, writeFiles = TRUE)
```

These are the resulting aggregated strata (graphically represented in Figure 3) :

```
R> solution$aggr_strata
```

```
  STRATO       M1       S1   N DOM1 COST CENS     SOLUZ
1      1  93.0545  13.3347 110    1    1    0  9.926579
2      2 139.2475  13.1852 101    1    1    0  9.012219
3      3 208.1852  20.7329  54    1    1    0  7.576654
4      4 314.8056  30.9523  36    1    1    0  7.540829
5      5 601.3036 185.4436  56    1    1    0 56.000000
```

```
R> sum(bethel(solution$aggr_strata, cv))
```

```
[1] 92
```

If we compare the strata obtained by the best method in package **stratification** and the ones obtained by the execution of `optimizeStrata` function in package **SamplingStrata**, we notice they are almost the same except for very small differences (Table 2). In particular, the first three strata are equal, the only difference is in the allocation equal to '10' in the second stratum, due to the rounding of the value 9.01 to the next upper integer.

The execution of all methods is repeated for each one of the four data frame, and related results are reported in Table 3.

Analyzing these results, we can say that the results of the application of the genetic algorithm are in general the second best after the LH method (with Kozak's algorithm). We should remark, however, that the differences between the GA and LH methods are due to the rounding rule (i.e., rounding to the next upper integer) of the GA method: without this penalization the two methods would be practically equivalent.

In conclusion, the fact that the genetic algorithm method is able to reproduce the results of consolidated methods in the univariate case, certifies its validity. In addition, it has to

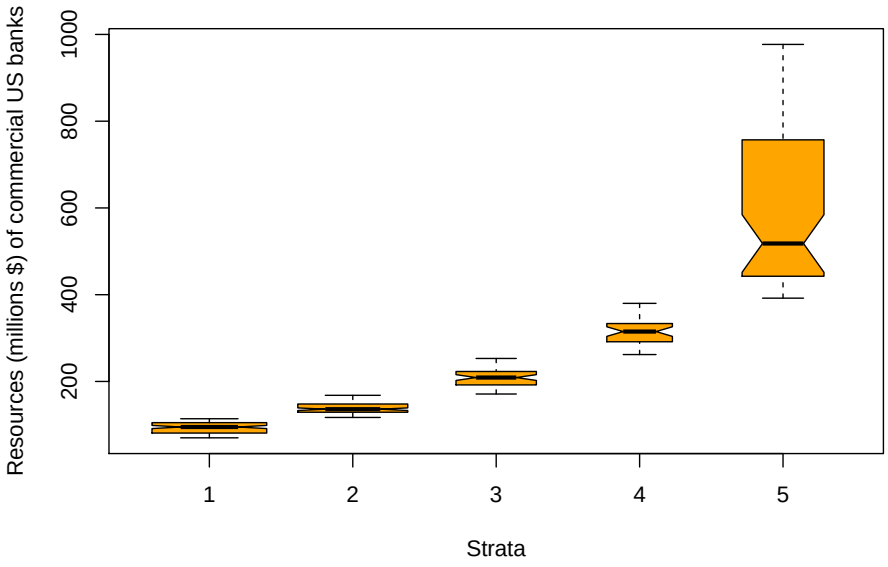


Figure 3: Strata resulting from the execution of the genetic algorithm.

| Stratum | LH upper bound | GA upper bound | LH population | GA population | LH allocation | GA allocation |
|---|---|---|---|---|---|---|
| 1 | 115.5 | 115.5 | 110 | 110 | 10 | 10 |
| 2 | 169.5 | 169.5 | 101 | 101 | 9 | 10 |
| 3 | 257.5 | 257.5 | 54 | 54 | 8 | 8 |
| 4 | 376.5 | 386.0 | 35 | 36 | 7 | 8 |
| 5 | 977.0 | 977.0 | 57 | 56 | 57 | 56 |

Table 2: Strata obtained by LH method (Kozak's algorithm) and GA (genetic algorithm).

| Dataset | CV | Strata | Geometric | CumFreq | LH (Sethi) | LH (Kozak) | Genetic |
|---|---|---|---|---|---|---|---|
| `UScities` | 0.01 | 5 | 180 | 175 | 183 | 172 | 172 |
| `UScolleges` | 0.01 | 5 | 197 | 190 | 162 | 158 | 161 |
| `USbanks` | 0.01 | 5 | 109 | 113 | 107 | 91 | 92 |
| `Debtors` | 0.0359 | 5 | 103 | 84 | 81 | 80 | 81 |

Table 3: Results of application of all methods to the four datasets.

be underlined that the GA is characterized by a general applicability to the multivariate and multi-domain cases, while all the other methods are strictly limited to the univariate single-domain case.

## 5. Conclusions

The approach implemented in the R package **SamplingStrata** allows to determine the best

stratification of a population frame, i.e., the one that ensures the minimization of the sample cost. When the data collection cost is the same in each stratum, the total cost is directly proportional to the number of units in the sample, and in this case what is minimized is the sample size. Its application is convenient whenever the following conditions occur:

1. a number of different auxiliary variables $X$'s are available in the population frame, so that different alternative solutions can be defined;

2. the number of different domains is not too high, and for each domain the first condition above is satisfied;

3. information directly or indirectly related to target variables $Y$'s is available for each unit in the population frame.

For instance, the above conditions hold in the case of agricultural surveys in the Italian National Institute of Statistics. In this statistical domain, the sampling frame contains the Decennial Agricultural Census data, and in many cases the survey target variables are a subset of those contained in the frame. In other cases, we may have a yearly survey on a census basis, and a monthly survey is carried out on a subset of the units surveyed yearly. In these cases the $Y$ variables are the same, and the method implemented in the package works ideally. If the $Y$ variables are not directly available, their values in the frame could be estimated by means of a predictive approach, using previous rounds of the same survey to estimate models linking target and auxiliary variables.

In any case, strong assumptions based on (explicit or implicit) models are made, and this should be taken into account when designing the sample.

## References

Baillargeon S, Rivest LP (2012a). "The Construction of Stratified Designs in R with the Package **stratification**." *Survey Methodology*, **37**(1), 53–65.

Baillargeon S, Rivest LP (2012b). ***stratification**: Univariate Stratification of Survey Populations*. R package version 2.2-3, URL http://CRAN.R-project.org/package=stratification.

Ballin M, Barcaroli G (2013). "Joint Determination of Optimal Stratification and Sample Allocation Using Genetic Algorithm." *Survey Methodology*, **39**(2), 369–393.

Barcaroli G, Pagliuca D, Willighagen E (2014). ***SamplingStrata**: Optimal Stratification of Sampling Frames for Multipurpose Sampling Surveys*. R package version 1.0-2, URL http://CRAN.R-project.org/package=SamplingStrata.

Bethel J (1989). "Sample Allocation in Multivariate Surveys." *Survey Methodology*, **15**(1), 47–57.

Chromy JB (1987). "Design Optimization with Multiple Objectives." In *Proceedings of the American Statistical Association Section on Survey Research Methods*, pp. 194–199.

Cochran WG (1977). *Sampling Techniques*. 3rd edition. John Wiley & Sons, New York.

Dalenius T, Hodges JL (1959). "Minimum Variance Stratification." *Journal of American Statistical Association*, **54**(285), 88–101.

DeJong KA (2006). *Evolutionary Computation: A Unified Approach*. MIT Press, Boston.

Gonzales JM, Eltinge JL (2010). "Optimal Survey Design: A Review." In *Section on Survey Research Methods – JSM, October 2010*. URL https://http://www.bls.gov/osmr/abstract/st/st100270.htm.

Gunning P, Horgan JM (2004). "A New Algorithm for the Construction of Stratum Boundaries in Skewed Populations." *Survey Methodology*, **30**(2), 159–166.

Hankin RKS (2013). ***partitions***: *Additive Partitions of Integers*. R package version 1.9-15, URL http://CRAN.R-project.org/package=partitions.

Hankin RKS, West LJ (2007). "Set Partitions in R." *Journal of Statistical Software, Code Snippets*, **23**(2), 1–12. URL http://www.jstatsoft.org/v23/c02/.

Hartigan JA, Wong MA (1979). "A $K$-Means Clustering Algorithm." *Journal of the Royal Statistical Society C*, **28**(1), 100–108.

Keskinturk T, Er S (2007). "A Genetic Algorithm Approach to Determine Stratum Boundaries and Sample Sizes of Each Stratum in Stratified Sampling." *Computational Statistics & Data Analysis*, **52**(1), 53–67.

Kozak M (2004). "Optimal Stratification Using Random Search Method in Agricultural Surveys." *Statistics in Transition*, **6**(5), 797–806.

Lavallée P, Hidiroglou M (1988). "On the Stratification of Skewed Populations." *Survey Methodology*, **14**(1), 33–43.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Sethi VK (1963). "A Note on the Optimum Stratification of Populations for Estimating the Population Means." *Australian Journal of Statistics*, **5**(1), 20–33.

Tillé Y, Matei A (2012). ***sampling***: *Survey Sampling*. R package version 2.5, URL http://CRAN.R-project.org/package=sampling.

Varberg D, Purcell EJ (1997). *Calculus*. 7th edition. Prentice Hall, New Jersey.

Willighagen E (2014). ***genalg***: *R Based Genetic Algorithm*. R package version 0.1.1.1, URL http://CRAN.R-project.org/package=genalg.

**Affiliation:**

Giulio Barcaroli
Italian National Institute of Statistics (Istat)
Methods, Tools and Methodological Support Division
E-mail: barcarol@istat.it

*Journal of Statistical Software* http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 61, Issue 4 *Submitted:* 2012-12-05
October 2014 *Accepted:* 2014-05-22