R2BEAT: An R Package for Performing Optimal Allocation and Sample Selection

Giulio Barcaroli[1], Ilaria Bombelli[2], Andrea Fasulo[3], Alessio Guandalini[4], and Marco D. Terribili[5]

[1]Independent consultant, Italy, giulio.barcaroli@gmail.com
[2]Italian Institute of Statistics (Istat), Italy, ilaria.bombelli@istat.it
[3]Italian Institute of Statistics (Istat), Italy, andrea.fasulo@istat.it
[4]Italian Institute of Statistics (Istat), Italy, alessio.guandalini@istat.it
[5]Italian Institute of Statistics (Istat), Italy, marcodionisio.terribili@istat.it

Abstract

The review presents the main features of the **R2BEAT** `R` package (Fasulo et al., 2023), which is designed for optimal sample allocation. The package integrates the Bethel (1989) algorithm, which extends optimal allocation (Tschprow, 1923; Neyman, 1934) to the multi-domain and multi-purpose case, and it also implements the extension proposed by Falorsi et al. (1998) for handling complex sampling designs. The package streamlines the entire sample design workflow, from sample optimisation to selection of sampling units.

Keywords: sampling, stratification, two-stage, design effect.

1 Introduction

Sample surveys conducted by National Statistical Institutes (NSIs) and other organisations often pursue multi-domain and multi-purpose objectives. Consequently, they are required to produce accurate estimates for multiple parameters and across various domains, both geographical and non-geographical.

Since surveys are subject to budgetary and logistical constraints, their design must be carefully planned to ensure high-quality estimates for the parameters of interest. Within this framework, several crucial decisions must be made, including determining the sample size, defining the stratification scheme, and allocating sampling units across strata and stages.

The proposed package, **R2BEAT** (standing for *R "to" Bethel Extended Allocation for Two-stage*), has been developed within this context (Barcaroli et al., 2023). It integrates the Bethel (1989) algorithm, which extends optimal allocation (Tschprow, 1923; Neyman, 1934) to the multi-domain and multi-purpose case, and it also implements the extension proposed by Falorsi et al. (1998) for handling complex sampling designs. Therefore, it fills an existing gap in the range of statistical software dedicated to sample size allocation, providing an advanced and flexible tool for the R community.

The paper is organised as follows. Section 2 describes the structure of the package and the case study used to illustrate its functionality. Section 3 explains the workflow for stratified sampling design - very common in economic surveys - while Section 4 focuses on two-stage sampling design with stratification of the primary stage units - widely used in household surveys. Finally, Section 5 provides conclusions.

For additional details and overview of further functions, readers may refer to the companion paper by Barcaroli et al. (2023). The workflow presented is based on the most recent functions available on the GitHub page of the package which are expected to be included in the forthcoming official release of the package.

2 Preliminarities

2.1 Structure of the package

The **R2BEAT** package provides a comprehensive set of functions for designing and selecting samples through optimal allocation, both for stratified and two-stage with stratification of the primary stage units.

The appropriate sampling design to apply in a specific situation depends on the information available in the sampling frame, for example, for implementing stratification or an unequal-inclusion-probability sampling design. In addition, to perform optimal allocation, information on the target variable(s) or at least on a suitable proxy is required. Such information may be obtained from a sampling frame, such as a register, or from a sample survey, either a concurrent survey or a previous wave of the same survey, and can be used to guide the allocation of the sampling units.

R2BEAT is able to manage both the scenarios and the typical workflow for designing and selecting a sample involves three main steps: (1) preparing the input data, (2) defining the sampling design and computing the optimal allocation, and (3) selecting the final sample units.

To install the latest stable version of `R2BEAT` from CRAN, use the command `install.packages("R2BEAT")` within the `R` environment. The most recent development version is available on GitHub and can be installed by executing `devtools::install_github("barcaroli/R2BEAT_2.0")`.

2.2 Case study

In this paper, we develop the workflow under a stratified sampling design and a two-stage sampling design with stratification of the primary sampling units[1]. In both cases, a sampling frame covering the entire population of interest is required, whereas information on the target variable(s) is obtained from a previously conducted sample survey. The case in which such information is available directly on the sampling frame differs slightly from this setting, and readers are referred to Barcaroli et al. (2023).

The sampling frame considered in this paper, `pop.RData`, refers to a population of 2,258,507 individuals and contains the following variables:

```
'data.frame':   2258507 obs. of  13 variables:
$ id_ind       : int  1 2 3 4 5 6 7 8 9 10 ...
$ id_hh        : Factor w/ 963018 levels "H1","H10","H100",..: 1 1 1 2 3 3 3 3 ...
$ municipality : num  1 1 1 1 1 1 1 1 1 1 ...
$ province     : Factor w/ 6 levels "north_1","north_2",..: 1 1 1 1 1 1 1 1 1 1 ...
$ region       : Factor w/ 3 levels "north","center",..: 1 1 1 1 1 1 1 1 1 1 ...
$ sex          : int  1 2 1 2 1 1 2 2 1 1 ...
$ cl_age       : Factor w/ 8 levels "(0,14]","(14,24]",..: 3 7 8 5 4 6 6 4 4 1 ...
$ active       : num  1 1 0 1 1 1 1 1 1 0 ...
$ unemployed   : num  0 0 0 0 0 0 0 0 0 0 ...
$ inactive     : num  0 0 1 0 0 0 0 0 0 1 ...
$ income_hh       : num  30488 30488 30488 21756 29871 ...
```

[1] To reproduce the analyses presented in these examples, all datasets are available for download at https://github.com/barcaroli/R2BEAT_datahttps://github.com/barcaroli/R2BEAT_data.

In particular, it contains

- `id_ind`: individual identifier,
- `id_hh`: household identifier to which the individual belongs,
- `municipality`: municipality identifier in which the individual lives,
- `province`: province (NUTS3) identifier in which the individual lives,
- `region`: region (NUTS2) identifier in which the individual lives;

demographic information:

- `sex`: sex of the individual,
- `cl_age`: age class of the individual in ten-year intervals;

information on target variables:

- `active`: binary indicator for occupational status “active”,
- `inactive`: binary indicator for occupational status “inactive”,
- `unemployed`: binary indicator for occupational status “unemployed”,
- `income_hh`: income.

Furthermore, for the present purpose, sampling data from a previous survey are also considered, `sample.RData`, comprising a two-stage (municipalities and individuals) sample of 9,421 units drawn from `Pop.RData`. This dataset includes the same variables described above and, in addition, the following variables useful for the present purpose:

- `weight`: the sampling weights assigned to each sampling unit,
- `stratum_2`: the strata used for stratifying the municipalities,
- `SR`: binary indicator for the Self-Representative (SR) municipalities. It is equal to 1 for the municipalities that are included certainly in the sample (inclusion probability equal to 1), 0 otherwise.

3 Stratified sampling design

Stratification of the sample is very common and highly effective. When one or more variables correlated with the survey’s target variables are available in the sampling frame, it is possible to partition the sampling units into strata and select an independent sample from each of them in order to obtain more efficient estimates.

Defining the proper number of sampling units to be collected in each stratum is an allocation problem. The optimal allocation (Tschprow, 1923; Neyman, 1934) assigns a larger portion of the sample to strata with greater population size and, in particular, to those characterised by higher variability of the target variable. In such strata, a greater sample size is required to achieve the desired level of efficiency of the estimates.

The allocation problem in the multivariate and multi-domain case can be formulated as an optimisation problem (Bethel, 1989), where the objective is to minimise the cost of the survey, usually expressed in terms of sample size, subject to a set of precision constraints on the estimates.

3.1 Step 1: Input preparation

From this premise, it follows straightforwardly that the inputs required to perform optimal allocation are strata information and a set of precision constraints for the estimates of the target variables.

The strata information can be obtained using the function prepareInputToAllocation_beat.1st. The parameters to be specified in the function are:

- frame: the sampling frame containing necessarily the identifier of the units, strata and domain variables and optionally the target variable(s).
- sample (optional): sample survey data, containing necessarily the strata and domains variables, the target variable(s) and the sampling weights. In this way, statistical summaries of the target variables will be estimated on the sample. Strata and domains variables must be consistent with those defined for samp_frame dataframe. Default is NULL, meaning that just sampling frame data are used,
- ID: name of the identifier of the units in the sampling frame.
- stratum: either name of the variable in samp_frame, which is taken as the stratum, or the name of the variables which have to be concatenated to obtain the stratum. In the latter case, the variables used to build the stratum are retained.
- domain: name of the variable(s) identifying the domain(s) for which estimates of the target variables must be disseminated. Domain(s) must be aggregation of the strata.
- target: names of the variable(s) in the sampling frame identifying the target variable(s) leading the planning of the survey.
- weight (optional): the sampling weights, whether the target variable(s) is (are) available only on sample data. The default is NULL, meaning that the target variables are available in the sampling frame and, therefore, the statistical summaries are computed on it.

Suppose that the sample is to be stratified by province, while the estimates of mean income and the incidence of unemployed individuals are to be controlled at the regional level. The parameter in prepareInputToAllocation_beat.1st can be set as follows:

```
input1 <- prepareInputToAllocation_beat.1st(frame=pop,
                                            sample=samp,
                                            ID="id_ind",
                                            stratum=c("province"),
                                            domain="region",
                                            target=c("income_hh","unemployed"),
                                            weights="d")
```

The function returns three objects:

1. file_strata: a dataframe of strata in which the population size, the mean (M1, M2, ...) and the standard deviation of the target variables in the population (S1, S2, ...) is provided. Furthermore, one column is specified for each domain. The global domain is included by default and is named DOM1. Two additional columns are filled automatically in: CENS, an identifier whether the stratum must be censused or not (the default is equal to 0 for all of them) and COST indicating the cost of the each interview in the stratum (the default is equal to 1 for all of them).

```
'data.frame': 6 obs. of  11 variables:
 $ STRATUM : Factor w/ 6 levels "north_1","north_2",..: 3 4 1 2 5 6
 $ province: Factor w/ 6 levels "north_1","north_2",..: 3 4 1 2 5 6
 $ DOM1    : Factor w/ 1 level "Total": 1 1 1 1 1 1
 $ DOM2    : Factor w/ 3 levels "north","center",..: 2 2 1 1 3 3
 $ N       : num  591517 205173 462420 482122 336608 ...
 $ M1      : num  21673 21054 27930 24332 16923 ...
 $ M2      : num  0.1032 0.1547 0.0212 0.0316 0.2922 ...
```

```
    $ S1      : num  19618 16565 26151 19252 14686 ...
    $ S2      : num  0.304 0.362 0.144 0.175 0.455 ...
    $ CENS    : num  0 0 0 0 0 0
    $ COST    : num  1 1 1 1 1 1
```

2. `var_list`: a vector of target variables as they appear in `file_strata` (i.e. `M1`, `M2`, ...or `S1`, `S2`, ...).
3. `ID_stratum`: a dataframe reporting the stratum to which each unit in the `samp_frame` belongs.

Then, the precision constraints in terms of coefficient of variation (CV) for each target variable in each domain have to be planned. Assume that the maximum acceptable coefficients of variation are 2% at the national level and 5% at the regional level for the mean of income, while 5% at the national level and 7% at the regional level for the incidence of unemployment. Then:

```
cv1 <- data.frame(DOM=c("DOM1", "DOM2"),
                  CV1=c(0.02, 0.05),
                  CV2=c(0.05, 0.07))
cv1
  DOM  CV1  CV2
1 DOM1 0.02 0.05
2 DOM2 0.05 0.07
```

3.2 Step 2: Optimal allocation

The optimal allocation is then computed using the `beat.1st` function and the inputs `file_strata` and `cv1` previously described:

```
alloc1 <- beat.1st(file_strata=input1$file_strata,
                   errors=cv1)
```

The final sample size of the optimal resulting allocation satisfying the precision constraints is 9,688. The object `alloc1` is a list of seven output objects:

1. `n`: a vector containing the optimal allocation for each stratum. Its total, `sum(alloc1$n)`, is equal to 9,688.
2. `file_strata`: the input dataset `file_strata` with an additional column, `n`, indicating the optimal allocation.
3. `alloc`: a dataframe specifying for each stratum the optimal allocation (`OPT`), the proportional allocation (`PROP`), and the uniform allocation (`UNIF`).
4. `sensitivity`: a dataframe with the precision constraints (`PlannedCV`), the expected CV (`ExpectedCV`), i.e. the CV that are expected to be obtained the optimal allocation), and the sensitivity (`Sensitivity 10%`) for each variable and each domain category. Sensitivity provides a suggestion about the expected variation in sample size if the planned errors change by 10%.
5. `ExpectedCV`: a dataframe with the maximum of the expected coefficients of variation (`Actual CV`), for each variable in each domain.
6. `PlannedCV`: a dataframe with the maximum coefficients of variation admissible for each domain and for each variable. It is the input errors dataframe, provided by the user.
7. `param_alloc`: a vector summarising all the parameters used for performing the optimal allocation.

In general, a reduction in the CV corresponds to a higher required level of precision, which in turn requires a larger sample size, and vice versa.

3.3 Step 3: Selection of sampling units

Given the allocation, the sample can be selected using the function `strata` from the `R` package **sampling** (Tillé and Matei, 2023). For proper implementation, prior to sample selection, it is recommended that both the sampling frame and the allocation dataframe are ordered by stratum:

```
library(sampling)
alloc1$file_strata <- alloc1$file_strata[order(alloc1$file_strata$STRATUM),]
pop <- merge(pop,input1$ID_stratum,by="id_ind")
pop <- pop[order(pop$STRATUM),]
s <- strata(data=pop,
            stratanames="STRATUM",
            size=alloc1$file_strata$n,
            method="srswor")
sample_str <- getdata(data=pop, m=s)
```

Finally, `sample_str` is the sample of size 9,688, stratified by province, which will yield estimates of mean income and the incidence of unemployment consistent with the planned precision constraints defined in `cv1`.

4 Two-stage sample design with stratification of the primary stage units

Sampling units may be organised in clusters; for example, individuals within households, workers within enterprises, or households within enumeration areas or municipalities.

For logistical and economic reasons, it may be useful to exploit this clustering. The typical case is household surveys. In these surveys, municipalities (Primary Stage Units, PSUs) are usually stratified. Then, within each stratum, a sample of municipalities is selected, typically with probability proportional to size, and within the selected municipalities a sample of households (Secondary Stage Units, SSUs) is drawn.

This sampling design is more convenient because it reduces the management complexity and therefore costs. However, this advantage comes at the expense of a reduction in the efficiency of the sample design, which must be taken into account when planning the sample.

In this context, the allocation problem is more complex, since the both PSUs and SSUs must be allocated. A solution can be obtained by following Falorsi et al. (1998). They propose iterating the Bethel algorithm, adjusting the design effect[2] at each iteration. Convergence is usually achieved within 5–6 iterations.

4.1 Step 1: Input preparation

The function `prepareInputToAllocation2` behaves similarly to the function described in Section 3.1 and likewise generates all the input objects required for the optimal allocation.

However, since this sample design is more complex, it needs more parameters:

[2] It denotes how much the sampling variance under the adopted sampling design is inflated with respect to SRS, on equal sample size. The design effect for SRS is equal to 1, whereas for clustered sampling designs it is greater than 1.

- `frame`: the sampling frame containing necessarily the identifier of the units, strata and domain variables and optionally the target variable(s).
- `RGdes`: the sampling data containing necessarily the strata and domains variables, the target variable(s) and the sampling weights. It must be a design object created with the `R` package **ReGenesees**[3].
- `RGcal` (optional): the sampling data containing necessarily the strata and domains variables, the target variable(s) and the sampling calibrated weights. It must be a calibration object created with the `R` package **ReGenesees**[4]. If NULL (default), it is set equal to the design object, `RGdes`.
- `id_PSU`: name of the identifier of the PSU.
- `id_SSU`: name of the identifier of the SSU.
- `stratum`: name of the variable in the sampling frame which is taken as the stratum. In contrast to the function used for the stratified sampling design, in the current version, this function does not perform variable concatenation. Therefore, it is recommended to prepare the concatenated variables beforehand.
- `target`: names of the variable(s) in the sampling frame identifying the target variable(s) leading the planning of the survey.
- `deff_level`: name of the variable(s) identifying the domain level at which compute the design effect. Although this information is applied in the algorithm at the stratum level, it is advisable to aggregate it to a higher hierarchical level to obtain more stable design effect estimates. The resulting design effect value is then applied to all strata belonging to the corresponding higher-level domains.
- `domain`: name of the variable(s) identifying the domain(s) for which estimates of the target variables must be disseminated. Domain(s) must be aggregation of the strata.
- `delta`: the average size of SSUs in terms of elementary units in each stratum. If SSUs match the survey units, `delta` must be equal to 1 in all the strata. Otherwise, it should be set equal to the average size of SSUs in terms of elementary units in the stratum.
- `minimum`: minimum number of SSUs to be interviewed in each selected PSU.

Suppose that the sample is to be a two-stage, municipalities and individuals, in which the municipalities are stratified by province, while the estimates of mean income and the incidence of unemployed individuals are to be controlled at the regional level and have been previously investigated in another sample survey. Furthermore, the design effect will be computed at the regional level.

A propedeutic step, before preparing the inputs for the optimal allocation, is to create the design object useful for computing the design effects and the estimator effects. The package **ReGenesees** is used for the present purpose also in the `prepareInputToAllocation2` function.

Since `samp` is a two-stage (municipalities, `municipality`, and individuals, `id_ind`) sample with stratification of the municipalities (`stratum_2`), the design object, `RGdes`, is defined as follows:

```
library(ReGenesees)
samp$stratum_2 <- as.factor(samp$stratum_2)
RGdes <- e.svydesign(data=samp,
                     ids=~municipality+id_ind,
                     strata=~stratum_2,
                     weights=~weight,
                     self.rep.str=~SR,
```

[3] For all the details see Zardetto (2015) and Zardetto (2023).

[4] See above.

```
                     check.data=TRUE)
```

Then, the parameter in `prepareInputToAllocation2` can be set as follows:

```
input2 <- prepareInputToAllocation2(frame=pop,
                                    RGdes=RGdes,
                                    id_PSU="municipality",
                                    id_SSU="id_ind",
                                    stratum="province",
                                    target=c("income_hh", "unemployed"),
                                    deff_level="region",
                                    domain="region",
                                    delta=1,
                                    minimum=120)
```

The function returns six dataframes:

1. `strata`: a dataframe with the same structure as the output provided by the function `prepareInputToAllocation_beat.1st` described in Section 3.1.
2. `deff`: a dataframe of strata with the design effect for each variable (`DEFF1, DEFF2, ...`) and the average size of the SSUs in the PSUs (`b_nar`).
3. `effst`: a dataframe with the estimator effect for each variable in each stratum[5]. When `RGcal` is NULL the estimator effect is equal to 1 for all the variables in each stratum.
4. `rho`: a dataframe with the intraclass correlation coefficient[6] for each variable in each stratum and for municipalities included for sure in the sample (Self-Representative). The correlation coefficient for larger municipalities (i.e. included certainly in the sample, since their selection probability is equal to 1) is equal to 1 by default.
5. `psu_file`: a dataframe of PSUs with the related stratum and their size (`PSU_MOS`) .
6. `des_file`: a dataframe of strata with the size, delta and minimum[7].

All of these objects, except `deff` (included for documentation purposes only), serve as inputs for the optimal allocation step.

Then, as before, the precision constraints in terms of coefficient of variation (CV) for each target variable in each domain have to be planned. Assume, in this case, that the maximum acceptable coefficients of variation is 2% at the national level and 5% at the regional level for the mean of income, while 5% at the national level and 7% at the regional level for the incidence of unemployment. Then:

```
cv2 <- data.frame(DOM=c("DOM1", "DOM2"),
                  CV1=c(0.02, 0.05),
                  CV2=c(0.05, 0.07))
```

[5] The estimator effect measures how much the sampling variance under the chosen estimator is inflated or deflated relative to the Horvitz–Thompson estimator (Horvitz and Thompson, 1952), under the same sample design. By definition, the Horvitz–Thompson estimator has an estimator effect equal to 1, while for instance a calibrated estimator (Deville and Särndal, 1992) typically yields values lower than 1.

[6] The correlation coefficient captures the degree of similarity among units within clusters. Positive values indicate strong within-cluster similarity, leading to higher design effects and poorer CVs. In contrast, negative values reflect greater within-cluster heterogeneity.

[7] By modifying this dataframe, it is possible to set different minimum values according to the strata.

```
cv2
  DOM  CV1  CV2
1 DOM1 0.02 0.05
2 DOM2 0.05 0.07
```

4.2 Step 2: Optimal allocation

The optimal allocation is then computed using the `beat.2st` function and the inputs previously described:

```
alloc2 <- beat.2st(file_strata=input2$file_strata,
                   errors=cv2,
                   des_file=input2$des_file,
                   psu_file=input2$psu_file,
                   rho=input2$rho,
                   effst=input2$effst)
```

```
  iterations PSU_SR PSU NSR PSU Total   SSU
1          0      0       0         0  9688
2          1     17      42        59 12677
3          2     19      76        95 11962
4          3     20      68        88 11944
```

The final sample size of the optimal resulting allocation satisfying the precision constraints comprises 88 PSUs, 20 Self-Representative (`PSU_SR`) and 68 Non-Self-Representative (`PSU_NSR`), and 11,944 SSUs.

The object `alloc2` is a list of eight output objects:

1. `iteractions`: a dataframe that, for each iteration, provides a summary of the number of PSUs (`PSU_Total`), distinguishing between Self-Representative (`PSU_SR`) and Non-Self-Representative (`PSU_NSR`) units, as well as the number of SSUs (`SSU`). This output is also printed to the screen.
2. `file_strata`: a dataframe equal to the input dataframe `file_strata` with additional columns: `DEFT1`, `DEFT2`, …reporting the square root of the design effect for each variable within each stratum, and `n`, specifying the optimal allocation.
3. `alloc`: a dataframe with optimal (ALLOC), proportional (PROP), equal (EQUAL) sample size allocation.
4. `planned`: a dataframe with the precision constraints (`Planned CV`) for each variable in each domain.
5. `expected`: a dataframe with the expected CVs with the given optimal allocation (`Expected CV`) for each variable in each domain.
6. `sensitivity`: a dataframe with a summary of the sensitivity at 10% for each domain and each variable.
7. `deft_c`: a dataframe with the design effect for each variable in each domain in each iteration. Note that `DEFT1_0`, `DEFT2_0`, …is always equal to 1 if `deft_start` is `NULL`. Otherwise is equal to `deft_start`. While `DEFT1`, `DEFT2`, …are the square root of the final design effect related to the given allocation reported also in `file_strata`.
8. `param_alloc`: a vector with a resume of all the parameter given for the allocation.

As before, a reduction in the CV corresponds to a higher required level of precision, which in turn requires a larger sample size, and vice versa. Moreover, for a fixed sample size, reducing the minimum number of units per PSU decreases the CV, since the sample is spread across more PSUs and the design effect decreases. Conversely, increasing the minimum leads to higher CVs.

4.3 Step 3: Selection of sampling units

The PSUs are then selected using:

```
sample_1st <- select_PSU(alloc=alloc2, type="OPT", pps=TRUE)
```

The selected PSUs are stored in the `sample_PSU` element of the output list. Using these, the final sample of secondary units can be selected:

```
PSU_sampled <- sample_1st$sample_PSU
sample_2st <- select_SSU(df = pop,
                         PSU_code ="municipality",
                         SSU_code ="id_ind",
                         PSU_sampled=PSU_sampled)
```

Finally, `sample_2st` is the two-stage sample of size 13,090, with the municipalities stratified by province, which will yield estimates of mean income and the incidence of unemployment consistent with the planned precision constraints defined in `cv2`.

A slight discrepancy may arise between the number of SSUs determined during allocation and those obtained after PSU selection. This occurs because the PSU selection process enforces the minimum number of SSUs (here, 120) per selected PSU, which may result in an increase in the total number of SSUs.

The two samples, `sample_str` and `sample_2st`, achieve the same level of precision, in terms of CVs, for the estimates of average income and unemployment incidence at both national and regional level. However, in the two-stage design the sampling units are clustered, which reduces the efficiency of the sample. As a result, a larger sample size, 13,090 instead of 9,688, is required to satisfy the same precision constraints.

5 Concluding remarks

R2BEAT stands out for its comprehensive approach to statistical data production, covering all stages from design to sample selection. It is especially flexible and adaptable, offering optimal allocation for both stratified and two-stage with stratification of the primary stage units sampling designs. This makes it valuable for various organisations, including national statistical institutes (NSIs), private research firms, research institutes and universities.

R2BEAT leverages auxiliary variables, improving sample design and allocation by making use of additional data from registers or previous surveys. Its user-friendly output allows for easy analysis and validation of the allocations and sample used in the survey. As the package stems from an ongoing development effort, it is continuously updated and maintained to guarantee maximum consistency and efficiency in the implementation of the methodology for sample design and selection.

References

Barcaroli, G., A. Fasulo, A. Guandalini, and M. Terribili (2023). "Two Stage Sampling Design and Sample Selection with the R Package R2BEAT". In: *The R Journal* 15.3, pp. 191–213.

Bethel, J. W. (1989). "Sample allocation in multivariate surveys". In: *Survey methodology* 15.1, pp. 47–57.

Deville, J.-C. and C.-E. Särndal (1992). "Calibration Estimators in Survey Sampling". In: *Journal of the American Statistical Association* 87.418, pp. 376–382. DOI: 10.1080/01621459.1992.10475217.

Falorsi, P. D., M. Ballin, C. De Vitiis, and G. Scepi (1998). "Principi e metodi del software generalizzato per la definizione del disegno di campionamento nelle indagini sulle imprese condotte dall'ISTAT". In: *Statistica Applicata* 10.2, pp. 235–257.

Fasulo, A., G. Barcaroli, S. Falorsi, A. Guandalini, D. Pagliuca, and M. D. Terribili (2023). *R2BEAT: Multistage Sampling Allocation and Sample Selection*. R package version 1.0.5. URL: https://CRAN.R-project.org/package=R2BEAT.

Horvitz, D. G. and D. J. Thompson (1952). "A Generalization of Sampling Without Replacement from a Finite Universe". In: *Journal of the American Statistical Association* 47.260, pp. 663–685.

Neyman, J (1934). "On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection". In: *Journal of the Royal Statistical Society* 97.4, pp. 558–625.

Tillé, Y. and A. Matei (2023). *sampling: Survey Sampling*. R package version 2.10. URL: https://CRAN.R-project.org/package=sampling.

Tschprow, A. (1923). "On the mathematical expectation of the moments of frequency distributions in the case of correlated observations". In: *Metron* 2, pp. 646–683.

Zardetto, D. (2015). "ReGenesees: an Advanced R System for Calibration, Estimation and Sampling Error Assessment in Complex Sample Surveys". In: *Journal of Official Statistics* 31.2, pp. 177–203. DOI: 10.1515/jos-2015-0013. URL: https://doi.org/10.1515/jos-2015-0013.

— (2023). *ReGenesees: R Evolved Generalised Software for Sampling Estimates and Errors in Surveys (R package)*. Version 2.3. R package. URL: https://github.com/DiegoZardetto/ReGenesees.