Package 'MultiWay.Sample.Allocation'

August 25, 2016

Type Package

Title Sample Allocation for multi-way stratified sampling designs and incomplete stratified sampling designs

Version 1.0

Date 2016-09-01

Author Paolo Righi

Maintainer Paolo Righi <parighi@istat.it>

Depends MASS

Description The package implements the sample allocation for one stage multi-way stratified (simple or with varying inclusion probabilities) and for incomplete stratified sampling designs. The allocation allows to control the expected sampling errors of the estimates of many parameters and several reference subpopulations.
The multi-way stratification is achieved by combining the variables defining the marginal cells of a multi-way contingency table.
In the context of the finite population sampling theory the marginal cells are identified by the categories of the variables defining the interest domain partitions being independent of each other. For example the stratification obtained by combining the categories of two partitions into domains of interest defines a two-way stratification. A partition is denoted as independent or marginal when the domains cannot be got as aggregation of other domains defined in other partitions. The allocation of the sample in the non-independent domains is done by aggregating the sample size of the marginal domains. The package defines the allocation of the sample in accordance with the precision constraints of the estimates.
In case of multi-way stratified designs, the sample size is fixed for each stratum. In case of incomplete stratified sampling design the sample size is fixed for each domain of interest. The allocation allows to control the expected sampling errors of the estimates of many parameters and several reference subpopulations.

License EUPL

R topics documented:

`allocation` *Definition of the inclusion probabilities*

Description

The function returns the inclusion probabilities for each profile in the input data frame namedata. A profile is given by the values of the YPRED variables and MARG variables. Expected Coefficient of Variation (CV) and statistics on the convergence of the algorithm are returned as well.

Usage

```
allocation(param1, output_mw, maxiter = 5, minouter, maxouter, minprof=0,
           integ.solution="NO ",convergence = 0.01)
```

Arguments

`param1`	LIST returned by parameters function
`output_mw`	LIST returned by allocation function in the previous sessions. Not required if the first session in launched. A session is carried out when the user launches the allocation function and a partial solution has been obtained (no convergence)
`maxiter`	maximum number of inner loop iterations within an outerloop
`minouter`	an integer identifying the index of the first outerloop of the session
`maxouter`	an integer identifying the index of the last outerloop of the session
`minprof`	an integer specifying the minimum number of units in the profile (record); by default, its value is 0
`integ.solution`	character specifying if the number of units in the profile (record) must be an integer; by default, its value is "NO ", if integer "YES"
`convergence`	a real number defining the convergence threshold of the algorithm

Details

The allocation function is applied after the parameters have been returned by the parameters function. The user can apply the allocation function in different sessions. At the end of a session an output (output_mw) is returned. This output saves the data of convergence process implemented in the previous sessions and it becomes the input for a further session. In this way the allocation process can be implemented by several steps. The session management depends on the values of minouter and maxouter according to the following rules:

(i) first session: minouter must be equal to 100; maxouter must be greater or equal to 100. If maxouter=100 then only one outer loop is performed; (ii) following sessions: minouter must be equal to maxouter+1 fixed in the previous session. Example: in the previous session minouter=103 and maxouter=105 then in the new session minouter must be equal to 106. If maxouter is equal to minouter only one outer loop is performed.

Depending on the dimension of the allocation problem (number of profiles, number of driving variables and number of margins), the function could be time spending.

Value

The function returns a LIST of data frames and matrices storing the inclusion probabilities computed at the end of each outer loop and the information of the convergence process. In the LIST the errors and hist_constraints data frames include the instrumental CV thresholds used in the inner loops and in the last outer loops (see note).

`output_mw` data frame containing for each profile the following variables:

- id: see parameters function;
- MARG(I): variable identifying the Ith domain partition (margin). See MARG in parameters function;
- pop: see pop in parameters function;
- size_out100: initial profile sample size. See start_size in parameters function;
- size_out101: profile sample size after the first outerloop;
- INC.PROB101: Inverse of size_out101. When the record in the namedata data frame is a population unit the value is the inclusion probability after the first outer loop
- ......
- size_outL: profile sample size after the Lth outer loop;
- INC.PROBL: Inverse of size_outL. When the record in the namedata data frame is a population unit the value is the inclusion probability after Lth outer loop.

`errors` data frame containing the instrumental CV thresholds used in the last inner loop of the last outerloop performed by the function;

`hist_constraints` matrix containing the instrumental CV thresholds used in each inner loop by outerloop by the function. The sample variable returnes the sample sizes;

`hist_expected_cv` matrix containing information for each inner loop in a given outer loop:

- out_iter: outer loop indicator variable;
- in_iter: inner loop indicator variable;
- sample: sample size returned in the given inner loop;
- cv_eta(I): expected CV for Ith combination between driving variables by domain. Note that these expected CVs hold when the algorithm converges to an optimal solution. Otherwise they are theoretical values useful for understanding the trend of convergence. Each CV has the following generic name cv_eta.v.m:d, where: v indicates the variables; m indicates the partition (margin); d indicates the domain of the given partition;

`hist_stat` matrix containing information on the convergence process. See hist_expected_cv for the variables out_iter, in_iter and sample. The remaing variables are:

- difsample: sample size difference between the current and the previous iteration;
- n_pos: number of domains of estimate where the relative difference between the current CV and the CV threshold is positive;
- n_pos1: number of domains of estimate where the relative difference between the current CV and the CV threshold is larger than 1%;
- mean_difpos: mean of the positive relative differences between the current CVs and the CV thresholds;
- median_difpos: median of the positive relative differences between the current CVs and the CV thresholds;
- q3_difpos: 3th quartile of the positive relative differences between the current CVs and the CV thresholds;
- max_difpos: max of the positive relative differences between the current CVs and the CV thresholds.

`convergence` matrix containing convergence parameters:

- maxiter: max number of the inner loop iteration in a outer loop;
- convergence: convergence threshold. The value is compared with difsample size. If difsample<convergence in two contiguous inner loop the algorithm initiates a new outer loop. If difsample<convergence in two contiguous outer loop the algorithm stops. The convergence is achieved.

Note

The instrumental CV thresholds in the errors data frame and hist_constraints matrix are based on the variance thresholds shown in formula (5.2) in Falorsi and Righi (2015). The CV indices cumulate the combination of the driving variables by the domain of estimates. When the CV combination specific thresholds are taken into account (see for example errors.grad), each index corresponds to the index given in the data frame of CV thresholds. When the CV driving variable by partition specific thresholds are taken into account (see for example errors.grad.matrix) the indices are built automatically. **Example**: let us consider two driving variables and two domain partitions, the indices are applied according to the following order; first driving variable (YPRED1) by domains of the first partition (MARG1); first driving variable (YPRED1) by domains of the second partition (MARG2); the second driving variable (YPRED2) by domains of the first partition (MARG1); second driving variable (YPRED2) by domains of the second partition (MARG2).

Dom variable in the errors data frame is an instrumental variable.

Author(s)

Paolo Righi

References

Falorsi, P.D. and Righi P.(2015). Generalized framework for defining the optimal inclusion probabilities of one-stage sampling designs for multivariate and multi-domain surveys, *Survey Methodology*, 41, n.1, 215-236.

See Also

`parameters`, `errors.grad`, `errors.grad.matrix`.

Examples

```
## Not run:
#The below examples work
####################################
## Example 1: graduates allocation
##            allocation in one step
####################################
data(graduates)
data(errors.grad)
par<-parameters(graduates, "id_rek", "prof.size", "start.allocation", "dom", "y", "vary",
errors.grad, "cvt", vars = 2, nmarg = 2)
output_allocation<-allocation(par, output_mw, maxiter = 5, minouter=100, maxouter=107,
minprof=0, integ.solution="NO ", convergence = 0.01)

#Results
#Data frame with the inclusion probabilities
head(output_allocation$output)
#Statistics on the convergence process
head(output_allocation$hist_stat)
head(output_allocation$hist_expected_cv)
head(output_allocation$hist_constraints)


####################################
## Example 2: graduates allocation
##            allocation in two steps
####################################


#First step - one outer loop iteration. Set the allocation function:
output_mw1<-allocation(par, maxiter = 5, minouter=100, maxouter=100, convergence = 0.01)

#Save the partial output
#setwd(".......")
save(output_mw1, file= "output_mw1.RData")

# If the second step allocation is performed in a new R session
#setwd("...")
#load("...\output_mw1.RData")

#Second Step allocation
output_mwf<-allocation(par, output_mw1, maxiter = 5, 101, 110, convergence = 0.01)

#Results
#Data frame with the inclusion probabilities
head(output_mwf$output)
#Statistics on the convergence process
head(output_mwf$hist_stat)
head(output_mwf$hist_expected_cv)
head(output_mwf$hist_constraints)


################################################
## Example 3: Allocation defined on two partitions
##            data frame ict.enterprises
################################################
```

```
#Note the variables DOM1, DOM2 and DOM3 identify three partitions of the population.
#Objective: allocate the sample taking into account DOM2 and DOM3. One solution is two
#delete DOM1 and rename DOM2 as DOM1 and DOM3 as DOM2.
#Otherwise the user can act on the CV thresholds as follows:

data(ict.enterprises)
data(ict.errors_2marg)
par.ict<-parameters(ict.enterprises, "STRATO", "N", "initial.size", "DOM", "M", "V",
ict.errors_2marg, "CV", vars = 2, nmarg = 3)
ict.all<-allocation(par.ict, output_mw, maxiter = 5, minouter=100, maxouter=107,
minprof=0, integ.solution="NO ", convergence = 0.10)

#The use of CVs equal to 999 in the first row related to DOM1 excludes such partition in the
# definition of the allocation


## End(Not run)
```

`errors.grad` *CV thresholds by specific combination for graduate example*

Description

Data frame containing the CVs for each combination between the driving variables and the domains of the partitions (margins).

Usage

```
data(errors.grad)
```

Format

A data frame with 1 observations on the following 70 variables.

`cvt1` CV threshold of the first combination

`...`

`cvt70` CV threshold of the 70th combination

Warning

Attention when defining the CV indices. The algorithm assumes the order given by the driving variables by the partitions (margins) according to the example in Note of allocation function.

See Also

parameters

Examples

```
data(errors.grad)
errors.grad
```

errors.grad.matrix *CV thresholds of the driving variable by partition*

Description

Data frame containing the homogeneus CVs for each combination of the driving variables by partitions (margins).

Usage

```
data(errors.grad.matrix)
```

Format

A data frame with 2 observations on the following 2 variables.

`cv_y1` CV threshold for the first driving variable

`cv_y2` CV threshold for the second driving variable

Details

Each row corresponds to a given partition (margin). This type of CV thresholds fixes for each driving variable homogeneus CV thresholds for all the domains of a given partiton.

Warning

Attention when defining the CV indices. The algorithm assumes the order given by the driving variables by the partitions (margins) according to the example in Note of allocation function.

Examples

```
data(errors.grad.matrix)
errors.grad.matrix
```

graduates *Graduates*

Description

Sub-population of the 2007 Italian graduates (Master Degree). Each observation is a profile. A profile identifies a set of graduates with the same values of the YPRED variables and MARG variables (see parameters function). The total number of graduates is 3427.

Usage

```
data (graduates)
```

Format

A data frame with 1679 observations on the following 10 variables.

`y1` working probability of graduates in the profile

`y2` looking for a job probability of graduates in the profile

`dom1` code identifying the domain of the first partition (margin). Factor with levels `1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20`

`dom2` code identifying the domain of the second partition (margin). Factor with levels `1 2 3 4 5 6 7 8 9 10 11 12 13 14 15`

`vary1` model variance of y1

`vary2` model variance of y2

`id_rek` code identifying the 1679 profiles

`prof.size` number of graduates in the profile

`start.allocation` starting sample size in the profile

Details

The sample alloction algorithm assumes the driving variables are unknown and predicted by a super population model (Sarndal *et al.*, 1992). The values of the variables vary1 and vary2 give the model uncertanty of the predicted values according to expression (4.1) in Falorsi and Righi (2015).

Note, the variables dom1, dom2 and id_rek can be numeric as well.

Source

Italian National Statistical Institute

References

Falorsi, P.D. and Righi P.(2015). Generalized framework for defining the optimal inclusion probabilities of one-stage sampling designs for multivariate and multi-domain surveys, *Survey Methodology*, 41, n.1, 215-236.

Sarndal, C.E., Swensson, B., and Wretman, J. (1992). *Model Assisted Survey Sampling*. New York: Springer-Verlag.

Examples

```
data(graduates)
head(graduates)
```

`ict.enterprises` *Enterprises in Computer and related economic activities*

Description

Italian Enterprises from 1 to 99 employees (year 1999) belonging to the Computer and related economic activities (2-digits of the NACE rev.1 classification) - Information and Communication Technology (ICT). Several domain partitions (margins) are taken into account. Each observation is a profile. A profile identifies a set of enterprises with the same values of the YPRED variables and MARG variables (see parameters function). The total number of enterprises is 10329.

Usage

```
data(ict.enterprises)
```

Format

A data frame with 1867 observations on the following 13 variables.

`STRATO` code identifying the record

`N` number of enterprises in the profile

`initial.size` starting sample size in the profile

`M1` predicted labour cost

`M2` predicted value added

`V1` model variance of ypred1

`V2` model variance of ypred2

`DOM1` Country (1 domain)

`DOM2` geographical region (20 domains)

`DOM3` economic activity group (6 groups) by Size class, number of persons employed 1-4; 5-9; 10-19;20-99 (24 domains)

`w1` economic activity group (6 domains)

`w2` size class, number of persons employed 1-4; 5-9; 10-19;20-99 (4 domains)

`w3` geographical region (20 domains)

Details

The sample alloction algorithm assumes the driving variables are unknown and predicted by the values in M1 and M2 according to a super population model (Sarndal *et al.*, 1992). The values of the variables V1 and V2 give the model uncertanty (model variances) of the predicted values according to expression (4.1) in Falorsi and Righi (2015).

In the example the starting sample size is fixed to 0.99 by N for all the population units. The optimal solution is not affected by the starting sample size but the number of loops can change.

If the records of the data frame were the singol enterprises then N=1. Furthermore, the starting sample size should be the starting inclusion probability.

The data frame contains two sets of partitions: DOM1, DOM2, DOM3; w1,w2,w3. Application can consider only one set of partition.

Source

Based on data of Italian National Statistical Institute.

References

Falorsi, P.D. and Righi P.(2015). Generalized framework for defining the optimal inclusion probabilities of one-stage sampling designs for multivariate and multi-domain surveys, *Survey Methodology*, 41, n.1, 215-236.

Falorsi, P.D. and Righi P.(2008). A balanced sampling approach for multi-way stratification designs for small area estimation, *Survey Methodology*, 34, n.2, 223-234.

Sarndal, C.E., Swensson, B., and Wretman, J. (1992). *Model Assisted Survey Sampling*. New York: Springer-Verlag.

Examples

```
data(ict.enterprises)
head(ict.enterprises)
## maybe str(ict.enterprises) ; plot(ict.enterprises) ...
```

`ict.errors_2marg` *CV thresholds of the driving variable by partition for ict.enterprise example. Two partitions are considered.*

Description

Data frame containing the homogeneus CVs for each combination of the driving variables by partitions (margins).

Usage

```
data(ict.errors_2marg)
```

Format

A data frame with 3 observations on the following 2 variables.

`CV1` CV thresholds for the first driving variable related to three partitions.

`CV2` CV thresholds for the second driving variable related to three partitions.

Details

Each row corresponds to a given partition (margin). This type of CV thresholds fixes for each driving variable homogeneus CV thresholds for all the domains of a given partiton.

Examples

```
data(ict.errors_2marg)
##############################
#The CV values of the first row are set to 999. Large CVs like these excludes the
#corresponding partiton to the definition of the allocation
##############################
ict.errors_2marg
```

`ict.errors_3marg` *CV thresholds of the driving variable by partition for ict.enterprise example. Three partitions are considered.*

Description

Data frame containing the homogeneus CVs for each combination of the driving variables by partitions (margins).

Usage

```
data(ict.errors_3marg)
```

Format

A data frame with 3 observations on the following 3 variables.

`CV1` CV thresholds for the first driving variable related to three partitions.

`CV2` CV thresholds for the second driving variable related to three partitions.

Details

Each row corresponds to a given partition (margin). This type of CV thresholds fixes for each driving variable homogeneus CV thresholds for all the domains of a given partiton.

Examples

```
data(ict.errors_3marg)
ict.errors_3marg
```

---

parameters *Definition of the parameters for the allocation procedure*

Description

Defines the parameters for the allocation procedure. Imports the input data frames and sets the number of driving variables and domain partitions (margins)

Usage

```
parameters(namedata, id, pop, start_size, MARG, YPRED, RES, err, CV, vars = 2,
           nmarg = 2)
```

Arguments

`namedata`	data frame with input values for the allocation procedure; its number of rows is the number of profiles
`id`	name of the variable identifying the profile
`pop`	name of the variable identifying the profile population size
`start_size`	name of the variable identifying the initial profile sample size; typically set equal to 0.5*pop
`MARG`	the name of the prefix specifying the domain partition (margin)
`YPRED`	the name of the prefix specifying the predicted values of the allocation driving variables
`RES`	the name of the prefix specifying the prediction variance of the allocation driving variables
`err`	data frame with the coefficient of variation thresholds
`CV`	the name of the prefix specifying the the coefficient of variation thresholds in the err data frame
`vars`	an integer specifying the number of driving variables; by default, its value is 2
`nmarg`	an integer specifying the number of domain partitions (margins); by default, its value is 2

Details

The records of the namedata data frame identify the profiles of the population. A profile is given by the values of the YPRED variables and MARG variables. Population units with the same values of this set of variables define a unique record (profile) in the data frame. The function aims to store the metadata describing the sampling allocation and the matrices required by the allocation function. Depending on the dimension of the allocation problem (number of profiles, number of driving variables and number of margins), the function could be time spending.

The metadata and the matrices are the input for the allocation function.

Value

The function returns a LIST with several objects required to run the allocation function.

`namedata`	the input data frame
`modelmatrix`	matrix (dimensions: number profiles - by - number of domains of interest) whose columns identify the domains of interest. Each column is the membership indicator variable (1 - record belongs to the domain, 0 - otherwise). The column are ordered as follows: ascending ordered domain partitions by ascending ordered domains in a given partition. Zero value occurs when the colomn identifies a domain not including the record.
`matrixres`	matrix (dimensions: number profiles - by - number of driving variables by number of domains of interest) whose columns have the prediction variances or 0 values. The column are ordered as follows:ascending ordered driving variables by ascending ordered domain partitions by ascending ordered domains in a given partition. Zero value occurs when the colomn identifies a domain not including the record.
`cvt_err`	data frame (dimension: 1 - by - number of driving variables by number of domains of interest) whose columns have the coefficient of variation.
`meta`	LIST containing meta information for the allocation function. Most of the objects are described in the argument section (id, pop, start_size, MARG,vars, nmarg). Furthrmore: dim_dom is the data frame including the variable MARG (number of domains for each partition) and MARGT the overall number of domains; domvar is a numeric value counting the number od domains by the number of driving variables; fpc is the finite population correction factor (Falorsi and Righi, 2015); POP is the profile population size; STRATA is the data frame required in the modified Chromy algorithm (Falorsi and Righi, 2015).

Author(s)

Paolo Righi

References

Falorsi, P.D. and Righi P.(2015). Generalized framework for defining the optimal inclusion probabilities of one-stage sampling designs for multivariate and multi-domain surveys, *Survey Methodology*, 41, n.1, 215-236.

See Also

`allocation`

Examples

```
## Not run:
#The below examples work
#########################################################################
## Example 1: graduates with specific variable by domains CV thresholds
#########################################################################
# Set up the meta information for the graduates example. Data frame graduates contains
#1679 records (profiles) for 3427 graduates.

data(graduates)
head(graduates)
data(errors.grad)
errors.grad
#Set up the function: two driving variables (vars=2) and two domain partitions (nmarg=2)
par<-parameters(graduates, "id_rek", "prof.size", "start.allocation", "dom", "y", "vary",
errors.grad, "cvt", vars = 2, nmarg = 2)

# Metadata stored in the par LIST containing the following objects
str(par)
# Input  data frame
head(par$namedata)
# Matrix modelmatrix
dim(par$modelmatrix)
head (par$modelmatrix)
# Matrix matrixres
dim(par$matrixres)
head (par$matrixres)
#Data frame cvt_err
head(par$cvt_err)
# List including meta parameters
str(par$meta)

#########################################################################
## Example 2: graduates with driving variables by partitions CV thresholds
#########################################################################
#Note the CV thresholds are the same of example 1 but they arranged in a different block way.
data(graduates)
head(graduates)
data(errors.grad.matrix)
#Set up the function: two driving variables (vars=2) and two domain partitions (nmarg=2)
par<-parameters(graduates, "id_rek", "prof.size", "start.allocation", "dom", "y", "vary",
errors.grad.matrix, "cv_y", vars = 2, nmarg = 2)

# Metadata stored in the par LIST containing the following objects
str(par)
# Input  data frame
head(par$namedata)
# Matrix modelmatrix
dim(par$modelmatrix)
head (par$modelmatrix)
# Matrix matrixres
dim(par$matrixres)
head (par$matrixres)
#Data frame cvt_err. The order of the CVs follows the rule given in Note of allocation function.
#The order is the same of the Example 1.
head(par$cvt_err)
```

```
# List including meta parameters
str(par$meta)

## End(Not run)
```

Index