The sampling Package

August 14, 2007

Version 0.8

Date 2007-08-14

Title Survey Sampling

Author Yves Tillé <yves.tille@unine.ch>, Alina Matei <alina.matei@unine.ch>

Maintainer Alina Matei <alina.matei@unine.ch>

Description Functions for drawing and calibrating samples.

Imports lpSolve

Depends MASS, lpSolve

License GPL version 2 or later.

LazyLoad no

Encoding latin1

R topics documented:

HTestimator .......................................................... 2
MU284 ................................................................. 3
UPbrewer ............................................................. 4
UPmaxentropy .......................................................... 5
UPmidzuno .............................................................. 7
UPmidzunopi2 ........................................................ 8
UPminimalsupport ..................................................... 9
UPmultinomial ......................................................... 10
UPpips ................................................................. 11
UPpivotal ............................................................... 12
UPpoisson ............................................................. 13
UPrandompivotal ...................................................... 14
UPrandomsystematic ................................................. 15
UPsampford ........................................................... 16
UPsystematic ......................................................... 17
**HTestimator**

*The Horvitz-Thompson estimator*

**Description**

Computes the Horvitz-Thompson estimator of the population total.

**Usage**

\[
\text{HTestimator}(y, pik, s)
\]

**Arguments**

- **y** 
  vector of the variable of interest; its length is equal to N, the population size.
- **pik** 
  vector of the first-order inclusion probabilities.
- **s** 
  vector of size N, with 0/1 elements indicating the unit status: 1 - unit is selected in the sample, 0 - otherwise.
MU284

See Also

UPTille

Examples

```r
# Belgian municipalities data base
data(belgianmunicipalities)
# Computes the inclusion probabilities
pik=inclusionprobabilities(belgianmunicipalities$Tot04,200)
N=length(pik)
n=sum(pik)
# Defines the variable of interest
y=belgianmunicipalities$TaxableIncome
# Draws a Poisson sample of expected size 200
s=UPpoisson(pik)
# Computes the Horvitz-Thompson estimator
HTestimator(y,pik,s)
```

MU284  The MU284 population

Description

This data is from Särndal et al (1992), see Appendix B, p. 652.

Usage

data(MU284)

Format

A data frame with 284 observations on the following 11 variables.

- **LABEL**  identifier number from 1 to 284.
- **P85**  1985 population (in thousands).
- **P75**  1975 population (in thousands).
- **RMT85**  revenues from 1985 municipal taxation (in millions of kronor).
- **CS82**  number of Conservative seats in municipal council.
- **SS82**  number of Social-Democratic seats in municipal council.
- **S82**  total number of seats in municipal council.
- **ME84**  number of municipal employees in 1984.
- **REV84**  real estate values according to 1984 assessment (in millions of kronor).
- **REG**  geographic region indicator.
- **CL**  cluster indicator (a cluster consists of a set of neighboring).
Source

http://lib.stat.cmu.edu/datasets/mu284

References


Examples

data(MU284)
hist(MU284$RMT85)

UPbrewer

Brewer sampling

Description

Uses Brewer method to select a sample of units (unequal probabilities, without replacement, fixed sample size).

Usage

UPbrewer(pik, eps=1e-06)

Arguments

pik vector of prescribed inclusion probabilities.
eps the control value, by default equal to 1e-06.

Details

Returns a vector (with elements 0 and 1) of size N, the population size. Each element k of this vector indicates the status of unit k (1, unit k is selected in the sample; 0, otherwise). The value eps is used to control pik (pik>eps & pik < 1-eps).

References


See Also

UPsystematic
Examples

```r
pik = c(0.2, 0.7, 0.8, 0.5, 0.4, 0.4)
s = UPbrewer(pik)
(1:length(pik))[s == 1]
```

Description

Maximum entropy sampling with fixed sample size and unequal probabilities (or Conditional Poisson sampling) is implemented by means of a sequential method.

Usage

```
UPmaxentropy(pik)
UPmaxentropypi2(pik)
UPMEqfromw(w, n)
UPMEpikfromq(q)
UPMEpiktildefrompik(pik, eps = 1e-6)
UPMEsfromq(q)
UPMEpik2frompikw(pik, w)
```

Arguments

- `n`: sample size.
- `pik`: vector of prescribed inclusion probabilities.
- `eps`: tolerance in Newton’s method; by default is 1E-6.
- `q`: matrix of the conditional selection probabilities for the sequential algorithm.
- `w`: parameter vector of the maximum entropy design.

Details

The maximum entropy sampling maximizes the entropy criterion:

\[
I(p) = - \sum_s p(s) \log[p(s)]
\]

The main procedure is `UPmaxentropy` which selects a sample (a vector of 0 and 1) from a given vector of inclusion probabilities. The procedure `UPmaxentropypi2` returns the matrix of joint inclusion probabilities from the first-order inclusion probability vector. The other procedures are intermediate steps. They can be useful to run simulations as shown in the examples below. The procedure `UPMEpiktildefrompik` computes the vector of the inclusion probabilities (denoted
pikt) of a Poisson sampling from the vector of the inclusion probabilities of the maximum entropy sampling. The maximum entropy sampling is the conditional design given the fixed sample size. The vector \( w \) can be easily obtained by \( w = \frac{\text{pikt}}{1 - \text{pikt}} \). Once \( \text{pikt} \) and \( w \) are deduced from \( \text{pik} \), a matrix of selection probabilities \( q \) can be derived from the sample size \( n \) and the vector \( w \) via \( \text{UPMEqfromw} \). Next, a sample can be selected from \( q \) using \( \text{UPMEsfromq} \). In order to generate several samples, it is more efficient to compute the matrix \( q \) (which needs some calculation), and then to use the procedure \( \text{UPMEsfromq} \). The vector of the inclusion probabilities can be recomputed from \( q \) using \( \text{UPMEpikfromq} \), which also checks the numerical precision of the algorithm. The procedure \( \text{UPMEpik2frompikw} \) computes the matrix of the joint inclusion probabilities from \( q \) and \( w \).

References


Examples

```r
### Example 1
## First method
UPmaxentropy(pik)
## Second method by using the intermediate procedures
n=sum(pik)
pikt=UPMEpiktildefrompik(pik)
w=pikt/(1-pikt)
q=UPMEqfromw(w,n)
UPMEsfromq(q)

### Example 2
## Sample of Belgian municipalities
data(belgianmunicipalities)
attach(belgianmunicipalities)
n=200
pik=inclusionprobabilities(average_income,n)
s=UPmaxentropy(pik)
```
UPmidzuno

# the sample is
as.character(Commune[s==1])
# the joint inclusion probabilities
pi2=UPmaxentropypi2(pik)
rowSums(pi2)/pik/n

### Example 3
###
# Selection of 200 samples of Belgian municipalities
# Once matrix q is computed, the selection of a sample is very quick.
# Simulations are thus possible.
data(belgianmunicipalities)
attach(belgianmunicipalities)
pik=inclusionprobabilities(averageincome,200)
pik=pik[pik!=1]
n=sum(pik)
pikt=UPMEpiktildefrompik(pik)
w=pikt/(1-pikt)
q=UPMEqfromw(w,n)
N=length(pik)
tt=rep(0,times=N)
# number of simulations
sim=200
for(i in 1:sim) tt = tt+UPMEsfromq(q)
tt=tt/sim
sum(abs(tt-pik))

UPmidzuno

Midzuno sampling

Description

Uses Midzuno method to select a sample of units (unequal probabilities, without replacement, fixed sample size).

Usage

UPmidzuno(pik)

Arguments

pik vector of prescribed inclusion probabilities.

Details

Returns a vector (with elements 0 and 1) of size N, the population size. Each element k of this vector indicates the status of unit k (1, unit k is selected in the sample; 0, otherwise).
References


See Also

UPtille

Examples

```r
#define the prescribed inclusion probabilities
pik=c(0.2,0.7,0.8,0.5,0.4,0.4)
#select a sample
s=UPmidzuno(pik)
#the sample is
(1:length(pik))[s==1]
```

UPmidzunopi2  
*Joint inclusion probabilities for Midzuno sampling*

Description

Computes the joint (second-order) inclusion probabilities for Midzuno sampling.

Usage

UPmidzunopi2(pik)

Arguments

pik  
the vector of the first-order inclusion probabilities.

Details

Returns a NxN matrix of the following form: the main diagonal contains the first-order inclusion probabilities for each unit k in the population; elements (k,l) are the joint inclusion probabilities of units k and l, with k not equal to l. N is the population size.

References


See Also

UPmidzuno
Examples

```r
#define the prescribed inclusion probabilities
pik=c(0.2,0.7,0.8,0.5,0.4,0.4)
#matrix of the joint inclusion probabilities
UPmidzunopi2(pik)
```

**Description**

Uses minimal support method to select a sample of units (unequal probabilities, without replacement, fixed sample size).

**Usage**

```r
UPminimalsupport(pik)
```

**Arguments**

- `pik`: vector of prescribed inclusion probabilities.

**Details**

Returns a vector (with elements 0 and 1) of size N, the population size. Each element k of this vector indicates the status of unit k (1, unit k is selected in the sample; 0, otherwise).

**References**


**Examples**

```
# define the prescribed inclusion probabilities
pik=c(0.2,0.7,0.8,0.5,0.4,0.4)
# select a sample
s=UPminimalsupport(pik)
# the sample is
(1:length(pik))[s==1]
```

```
# Example 2
```

```
data(belgianmunicipalities)
Tot=belgianmunicipalities$Tot04
```
UPmultinomial

Description

Uses Hansen-Hurwitz method to select a sample of units (unequal probabilities, with replacement, fixed sample size).

Usage

UPmultinomial(pik)

Arguments

pik

the vector of the prescribed inclusion probabilities.

Details

Returns a vector (with elements 0 and 1) of size N, the population size. Each element k of this vector indicates the status of unit k (1, unit k is selected in the sample; 0, otherwise).

References


See Also

rmultinom

Examples

#define the prescribed inclusion probabilities
pik=c(0.2,0.7,0.8,0.5,0.4,0.4)
#select a sample
s=UPmultinomial(pik)
#the sample is
(1:length(pik))[s==1]
UPopips

Order \( \pi \)ps sampling

Description

Implements order \( \pi \)ps sampling (unequal probabilities, without replacement, fixed sample size).

Usage

```
UPopips(lambda, type=c("pareto", "uniform", "exponential"))
```

Arguments

- `lambda` vector of working inclusion probabilities or target ones.
- `type` the type of order sampling (pareto, uniform, exponential).

Details

Returns a vector of selected units of size \( n \), the sample size.

References


See Also

`inclusionprobabilities`

Examples

```
#define the working inclusion probabilities
lambda=c(0.2,0.7,0.8,0.5,0.4,0.4)
#draw a Pareto sample
s=UPopips(lambda, type="pareto")
#the sample is
s
```
**UPpivotal**  

_Pivotal sampling_

**Description**

Selects an unequal probability sample using the pivotal method (unequal probabilities, without replacement, fixed sample size).

**Usage**

`UPpivotal(pik, eps=1e-6)`

**Arguments**

- `pik`: vector of prescribed inclusion probabilities.
- `eps`: the control value, by default equal to 1e-6.

**Details**

Returns a vector (with elements 0 and 1) of size N, the population size. Each element k of this vector indicates the status of unit k (1, unit k is selected in the sample; 0, otherwise). The value eps is used to control pik (pik>eps & pik < 1-eps).

**References**


**See Also**

`UPrandompivotal`

**Examples**

```r
# define the prescribed inclusion probabilities
pik=c(0.2,0.7,0.8,0.5,0.4,0.4)
# select a sample
s=UPpivotal(pik)
# the sample is
(1:length(pik))[s==1]
```
UPpoisson

Poisson sampling

Description

Draws a Poisson sample using a prescribed vector of first-order inclusion probabilities (unequal probabilities, without replacement, random sample size).

Usage

UPpoisson(pik)

Arguments

pik vector of the first-order inclusion probabilities.

Details

Returns a vector (with elements 0 and 1) of size N, the population size. Each element k of this vector indicates the status of unit k (1, unit k is selected in the sample; 0, otherwise).

See Also

inclusionprobabilities

Examples

############
## Example 1
############
# definition of pik
pik=c(1/3,1/3,1/3)
# select a sample
s=UPpoisson(pik)
# the sample is
(1:length(pik))[s==1]
############
## Example 2
############
data(belgianmunicipalities)
Tot=belgianmunicipalities$Tot04
name=belgianmunicipalities$Commune
n=200
pik=inclusionprobabilities(Tot,n)
# select a sample
s=UPpoisson(pik)
# the sample is
as.vector(name[s==1])
Description

Selects a sample using the pivotal method, when the order of the population units is random (unequal probabilities, without replacement, fixed sample size).

Usage

UPrandompivotal(pik)

Arguments

pik vector of prescribed inclusion probabilities.

Details

Returns a vector (with elements 0 and 1) of size N, the population size. Each element k of this vector indicates the status of unit k (1, unit k is selected in the sample; 0, otherwise).

References


See Also

UPpivotal

Examples

```r
# define the prescribed inclusion probabilities
pik = c(0.2, 0.7, 0.8, 0.5, 0.4, 0.4)
# select a sample
s = UPrandompivotal(pik)
# the sample is
(1:length(pik))[s==1]
```
UPrandomsystematic  Random systematic sampling

Description

Selects a sample using the systematic method, when the order of the population units is random (unequal probabilities, without replacement, fixed sample size).

Usage

UPrandomsystematic(pik)

Arguments

pik  vector of prescribed inclusion probabilities.

Details

Returns a vector (with elements 0 and 1) of size N, the population size. Each element k of this vector indicates the status of unit k (1, unit k is selected in the sample; 0, otherwise).

References


See Also

UPsystematic

Examples

#define the prescribed inclusion probabilities
pik=c(0.2,0.7,0.8,0.5,0.4,0.4)
#select a sample
s=UPrandomsystematic(pik)
#the sample is
(1:length(pik))[s==1]
Description

Uses Sampford method to select a sample of units (unequal probabilities, without replacement, fixed sample size).

Usage

```R
USampford(pik, eps=1e-6)
```

Arguments

- `pik`: vector of prescribed inclusion probabilities.
- `eps`: the control value, by default equal to 1e-6.

Details

Returns a vector (with elements 0 and 1) of size N, the population size. Each element k of this vector indicates the status of unit k (1, unit k is selected in the sample; 0, otherwise). The value eps is used to control pik (pik>eps & pik < 1-eps). The sample size must be small with respect to the population size; otherwise, the selection time can be very long.

References


See Also

- `UPsystematic`

Examples

```R
#define the prescribed inclusion probabilities
pik=c(0.2,0.7,0.8,0.5,0.4,0.4)
s=USampford(pik)
#the sample is
(1:length(pik))[s==1]
```
Systematic sampling

Description

Uses the systematic method to select a sample of units (unequal probabilities, without replacement, fixed sample size).

Usage

UPsystematic(pik)

Arguments

pik vector of prescribed inclusion probabilities.

Details

Returns a vector (with elements 0 and 1) of size N, the population size. Each element k of this vector indicates the status of unit k (1, unit k is selected in the sample; 0, otherwise).

References


See Also

inclusionprobabilities, UPrandomsystematic

Examples

###########
## Example 1
###########
# define the prescribed inclusion probabilities
pik=c(0.2,0.7,0.8,0.5,0.4,0.4)
# select a sample
s=UPsystematic(pik)
# the sample is
{s[length(pik)][s==1]}

###########
## Example 2
###########
data(belgianmunicipalities)
Tot=belgianmunicipalities$Tot04
name=belgianmunicipalities$Commune
pik=inclusionprobabilities(Tot,200)
# select a sample
UPsystematicpi2

Joint inclusion probabilities for systematic sampling

Description

Computes the joint (second-order) inclusion probabilities for systematic sampling.

Usage

UPsystematicpi2(pik)

Arguments

pik 

the vector of the first-order inclusion probabilities.

Details

Returns a NxN matrix of the following form: the main diagonal contains the first-order inclusion probabilities for each unit k in the population; elements (k,l) are the joint inclusion probabilities of units k and l, with k not equal to l. N is the population size.

References


See Also

UPsystematic

Examples

#define the prescribed inclusion probabilities
pik=c(0.2,0.7,0.8,0.5,0.4,0.4)
#matrix of the joint inclusion probabilities
UPsystematicpi2(pik)
**Description**

Uses the Tillé method to select a sample of units (unequal probabilities, without replacement, fixed sample size).

**Usage**

```
UPtille(pik, eps=1e-6)
```

**Arguments**

- `pik` the vector of the prescribed inclusion probabilities.
- `eps` the control value, by default equal to 1e-6.

**Details**

Returns a vector (with elements 0 and 1) of size N, the population size. Each element k of this vector indicates the status of unit k (1, unit k is selected in the sample; 0, otherwise). The value eps is used to control pik (pik>eps & pik < 1-eps).

**References**


**See Also**

`UPsystematic`

**Examples**

```
############
## Example 1
############
# define the prescribed inclusion probabilities
pik = c(0.2, 0.7, 0.8, 0.5, 0.4, 0.4)
# select a sample
s = UPtille(pik)
# the sample is
(1:length(pik))[s == 1]

############
## Example 2
############
```
# Selection of samples of municipalities
# with equal or unequal probabilities.
# Comparison of the accuracy by boxplots.
b=data(belgianmunicipalities)
pik=inclusionprobabilities(belgianmunicipalities$Tot04,200)
N=length(pik)
n=sum(pik)
# number of simulations; for an accurate result, increase this value
sim=10
ss=array(0,c(sim,8))
# the interest variable
y=belgianmunicipalities$TaxableIncome
# simulation and computation of the Horvitz-Thompson estimator
for(i in 1:sim)
{
  cat("Step ",i,"\n")
  ss[i,]=ss[i,]+c(
    HTestimator(y,pik,UPpoisson(pik)),
    HTestimator(y,pik,UPrandomsystematic(pik)),
    HTestimator(y,pik,UPrandompivotal(pik)),
    HTestimator(y,pik,UPtille(pik)),
    HTestimator(y,pik,UPmidzuno(pik)),
    HTestimator(y,pik,UPsystematic(pik)),
    HTestimator(y,pik,UPpivotal(pik)),
    HTestimator(y,rep(n/N,N),srswor(n,N)))
}
# boxplots of the estimators
colnames(ss) <-
c("poisson","rsyst","rpivotal","tille","midzuno","syst","pivotal","srswor")
boxplot(data.frame(ss), las=3)
# The results of the simulations can be interpreted.
# Simple random sampling
# and Poisson sampling are not accurate.
# All unequal probability sampling methods seem
# to have the same accuracy, except systematic sampling and pivotal sampling
# that have variances which depend on the order of the units in the file.

UPtillepi2

Joint inclusion probabilities for Tillé sampling

Description

Computes the joint (second-order) inclusion probabilities for Tillé sampling.

Usage

UPtillepi2(pik,eps=1e-6)
Arguments

pik the vector of the first-order inclusion probabilities.
eps the control value, by default equal to 1e-6.

Details

Returns a NxN matrix of the following form: the main diagonal contains the first-order inclusion probabilities for each unit k in the population; elements (k,l) are the joint inclusion probabilities of units k and l, with k not equal to l. N is the population size. The value eps is used to control pik (pik>eps & pik < 1-eps).

References


See Also

UPtille

Examples

#define the prescribed inclusion probabilities
pik=c(0.2,0.7,0.8,0.5,0.4,0.4)
pik_joint=UPtillepi2(pik)
#the joint inclusion probabilities
pik_joint

balancedcluster

Description

Selects a balanced cluster sample and returns a matrix containing the vector of inclusion probabilities and the selected sample.

Usage

balancedcluster(X,m,cluster,selection=1,comment=TRUE,method=1)

Arguments

X matrix of auxiliary variables on which the sample must be balanced.
m number of clusters to be selected.
cluster vector of integers that defines the clusters.
selection 1, selection of the clusters with probabilities proportional to size,
           2, selection of the clusters with equal probabilities.
comment a comment is written during the execution if comment is equal to TRUE.
method the used method in the function samplecube.

See Also

samplecube, fastflightcube, landingcube

Examples

```
###
# Example 1
###
# definition of the clusters; there are 15 units in 3 clusters
cluster=c(1,1,1,1,1,2,2,2,2,2,3,3,3,3,3)
# Matrix of balancing variables
X=cbind(c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15))
# Selection of 2 clusters
s=balancedcluster(X,2,cluster,2,TRUE)
# the sample of clusters with the inclusion probabilities of the clusters
s
# the selected clusters
unique(cluster[s[,1]==1])
# the selected units
(1:length(cluster))[s[,1]==1]
# with the probabilities
s[s[,1]==1,2]
###
# Example 2
###
data(MU284)
X=cbind(MU284$P75,MU284$CS82,MU284$SS82,MU284$S82,MU284$ME84)
s=balancedcluster(X,10,MU284$CL,1,TRUE)
cluster=MU284$CL
# the selected clusters
unique(cluster[s[,1]==1])
# the selected units
(1:length(cluster))[s[,1]==1]
# with the probabilities
s[s[,1]==1,2]
```

balancedstratification

Balanced stratification

Description

Selects a stratified balanced sample (a vector of 0 and 1). Firstly, the flight phase is applied in each stratum. Secondly, the strata are aggregated and the flight phase is applied on the whole population. Finally, the landing phase is applied on the whole population.
Usage

balancedstratification(X, strat, pik, comment=TRUE, method=1)

Arguments

X matrix of auxiliary variables on which the sample must be balanced.
strat vector of integers that specifies the stratification.
pik vector of inclusion probabilities.
comment a comment is written during the execution if comment is equal to TRUE.
method the used method in the function samplecube.

References


See Also

camplecube, fastflightcube, landingcube

Examples

############
## Example 1
############
# variable of stratification (3 strata)
strat=c(1,1,1,1,2,2,2,2,2,2,3,3,3,3,3)
# matrix of balancing variables
X=cbind(c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15))
# Vector of inclusion probabilities.
# The sample has its size equal to 9.
pik=rep(3/5,times=15)
# Selection of a stratified sample
s=balancedstratification(X, strat, pik, comment=TRUE)
# The sample is (1:length(pik))[s==1]
############
## Example 2
############
data(MU284)
X=cbind(MU284$P75,MU284$CS82,MU284$SS82,MU284$S82,MU284$ME84)
strat=MU284$REG
balancedtwostage

Balanced two-stage sampling

Description

Selects a balanced two-stage sample. The function returns a matrix whose columns are the following five vectors: the selected second-stage sampling units (0 - unselected, 1 - selected), the final inclusion probabilities, the selected primary sampling units, the inclusion probabilities of the first stage, the inclusion probabilities of the second stage.

Usage

balancedtwostage(X, selection, m, n, PU, comment=TRUE, method=1)

Arguments

- **X**
  - matrix of auxiliary variables on which the sample must be balanced.

- **selection**
  - 1, for simple random sampling without replacement at each stage,
  - 2, for self-weighting two-stage selection.

- **m**
  - number of primary sampling units to be selected.

- **n**
  - number of second-stage sampling units to be selected.

- **PU**
  - vector of integers that defines the primary sampling units.

- **comment**
  - a comment is written during the execution if comment is equal to TRUE.

- **method**
  - the used method in the function samplecube.
belgianmunicipalities

See Also

samplecube, fastflightcube, landingcube, balancedstratification, balancedcluster

Examples

###########
## Example 1
###########
# definition of the primary units (3 primary units)
PU=c(1,1,1,1,2,2,2,2,3,3,3,3,3)
# matrix of balancing variables
X=cbind(c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15))
# Selection of 2 primary sampling units and 4 second-stage sampling units
s=balancedtwostage(X,1,2,4,PU,comment=TRUE)
# The samples and the inclusion probabilities
s

###########
## Example 2
###########
data(MU284)
X=cbind(MU284$P75,MU284$CS82,MU284$SS82,MU284$ME84)
N=dim(X)[1]
PU=MU284$CL
m=20
n=60
res=balancedtwostage(X,1,m,n,PU,TRUE)
# The samples and the inclusion probabilities
res

---

belgianmunicipalities

The Belgian municipalities population

Description

This data provides information about the Belgian population of July 1, 2004 compared to that of July 1, 2003, and some financial information about the municipality incomes at the end of 2001.

Usage

data(belgianmunicipalities)

Format

A data frame with 589 observations on the following 17 variables:

Commune municipality name.

INS ‘Institut National de statistique’ code.
boundedrakingratio

Province  province number.
Arrondiss  administrative division number.
Men04  number of men on July 1, 2004.
Women04  number of women on July 1, 2004.
Tot04  total population on July 1, 2004.
Men03  number of men on July 1, 2003.
Women03  number of women on July 1, 2003.
Tot03  total population on July 1, 2003.
Diffmen  number of men on July 1, 2004 minus the number of men on July 1, 2003.
Diffwom  number of women on July 1, 2004 minus the number of women on July 1, 2003.
DiffTOT  difference between the total population on July 1, 2004 and on July 1, 2003.
TaxableIncome  total taxable income in euros in 2001.
Totaltaxation  total taxation in euros in 2001.
averageincome  average of the income-tax return in euros in 2001.
medianincome  median of the income-tax return in euros in 2001.

Source


Examples

data(belgianmunicipalities)
hist(belgianmunicipalities$medianincome)

boundedrakingratio  g-weights of the raking ratio estimator

Description

Computes the g-weights of the raking ratio estimator. The g-weights should lie in the specified bounds.

Usage

boundedrakingratio(Xs,piks,t,q=rep(1,length(piks)),LOW=0,UP=10)
boundedrakingratio

Arguments

Xs       matrix of calibration variables.
piks    vector of inclusion probabilities.
t       vector of population totals.
q       vector of weights for the distance. The variation of the g-weights is reduced for the small values of q. When a component of q is null, the corresponding g-weight remains equal to 1.
LOW     smallest value for the g-weights.
UP      largest value for the g-weights.

References


See Also

regressionestimator, boundedregressionestimator, rakingratio, checkcalibration

Examples

########################
## Example 1
########################
# matrix of auxiliary variables
Xs=cbind(c(1,1,1,1,0,0,0,0,0,0), c(0,0,0,0,1,1,1,1,1,1), c(1,2,3,4,5,6,7,8,9,10))
# inclusion probabilities
piks=rep(0.2,times=10)
# vector of totals
T=c(24,26,290)
# the Horvitz-Thompson estimator
THT=t(1/piks)%*%Xs
# the g-weights
G=boundedrakingratio(Xs,piks,T,LOW=0.8,UP=1.2)
# the calibrated estimator is equal to T
Tcal=t(G/piks)%*%Xs
# the g-weights are between LOW and UP bounds
G
########################
## Example 2
########################
Example of raking ratio, regression, bounded regression and bounded raking ratio estimators, with the data of Belgian municipalities.

Firstly, a sample is selected by means of Poisson sampling. Secondly, the g-weights and the calibration estimators are calculated.

The database of Belgian municipalities
data(belgianmunicipalities)attach(belgianmunicipalities)X=cbind(Men03/mean(Men03),Women03/mean(Women03),Diffmen,Diffwom,TaxableIncome/mean(TaxableIncome),Totaltaxation/mean(Totaltaxation),averageincome/mean(averageincome),medianincome/mean(medianincome))

# selection of a sample with expectation size equal to 200 by means of Poisson sampling
# the inclusion probabilities are proportional to the average income
pik=inclusionprobabilities(averageincome,200)
N=length(pik) # population size
s=UPpoisson(pik) # sample
Xs=X[s==1,] # matrix of calibration variables of the sample
piks=pik[s==1] # inclusion probabilities of the sample
n=length(piks) # sample size

t=c(t(rep(1,times=N))%*%X) # The Horvitz-Thompson estimators of auxiliary variables
c((1/piks) %*% Xs) # The true total

g-weights of the regression estimator
boundedregressionestimator

Description

Computes the g-weights of the regression estimator. The g-weights should lie in the specified bounds.

Usage

boundedregressionestimator(Xs,piks,t,q=rep(1,times=length(piks)),LOW=0,UP=10)

Arguments

- **Xs**: matrix of calibration variables.
- **piks**: vector of inclusion probabilities.
- **t**: vector of population totals.
- **q**: vector of weights for the distance. The variation of the g-weights is reduced for small values of q. When a component of q is null, the corresponding g-weight remains equal to 1.
- **LOW**: smallest value for the g-weights.
- **UP**: largest value for the g-weights.

References


See Also

rakingratio, boundedregressionestimator, regressionestimator, checkcalibration

Examples

```
###
## Example 1
###
# matrix of auxiliary variables
Xs=cbind(
  c(1,1,1,1,0,0,0,0,0,0),
  c(0,0,0,0,1,1,1,1,1,1),
  c(1,2,3,4,5,6,7,8,9,10)
)
# inclusion probabilities
piks=rep(0.2,times=10)
# vector of totals
t=c(24,26,290)
# the Horvitz-Thompson estimator
```
tHT = t(1/piks) * Xs
# the g-weights
q = boundedregressionestimator(Xs, piks, t, LOW=0.75, UP=1.2)
# the regression estimator is equal to t
tcal = t(q/piks) * Xs
# the g-weights are between LOW and UP bounds
q

-------------
## Example 2
-------------
# Example of rakingratio, regression, boundedregression and boundedrakingratio estimators,
# with the data of Belgian municipalities.
# Firstly, a sample is selected by means of Poisson sampling.
# Secondly, the g-weights and the calibration estimators are calculated.
# the database of Belgian municipalities
data(belgianmunicipalities)
attach(belgianmunicipalities)
X = cbind(
  Men03/mean(Men03),
  Women03/mean(Women03),
  Diffmen,
  Diffwom,
  TaxableIncome/mean(TaxableIncome),
  Totaltaxation/mean(Totaltaxation),
  averageincome/mean(averageincome),
  medianincome/mean(medianincome))
# selection of a sample with expectation size equal to 200
# by means of Poisson sampling
# the inclusion probabilities are proportional to the average income
pik = inclusionprobabilities(averageincome, 200)
N = length(pik) # population size
s = UPpoisson(pik) # sample
Xs = X[s == 1,] # matrix of calibration variables of the sample
piks = pik[s == 1] # inclusion probabilities of the sample
n = length(piks) # sample size
# vector of population totals of the auxiliary variables
t = c(t(rep(1, times=N)) * X)
# The Horvitz-Thompson estimator of auxiliary variables
c((1/piks) * Xs)
# The true total
t
# Computation of the g-weights
# by means of different calibration methods.
g1 = regressionestimator(Xs, piks, t)
g2 = rakingratio(Xs, piks, t)
g3 = boundedregressionestimator(Xs, piks, t, LOW=0.5, UP=1.5)
g4 = boundedrakingratio(Xs, piks, t, LOW=0.5, UP=1.5)
# In some cases, the calibration does not exist
# particularly when bounds are used.
# If the calibration is possible, the calibration estimators are the following
if (checkcalibration(Xs, piks, t, g1)) c((g1/piks) * Xs) else print("error")
if (checkcalibration(Xs, piks, t, g2)) c((g2/piks) * Xs) else print("error")
if (checkcalibration(Xs, piks, t, g3)) c((g3/piks) * Xs) else print("error")
if(checkcalibration(Xs,piks,t,g4)) c((g4/piks) %*% Xs) else print("error")

---

### Description

Checks the validity of the calibration. In some cases, the regression or the raking ratio estimators do not exist, and the g-weights do not allow calibration. The function returns TRUE or FALSE.

### Usage

```r
checkcalibration(Xs, pik, t, g)
```

### Arguments

- **Xs**: matrix of calibration variables.
- **pik**: vector of inclusion probabilities.
- **t**: vector of population totals.
- **g**: vector of g-weights.

### See Also

`regressionestimator`, `rakingratio`, `boundedrakingratio`, `boundedregressionestimator`

### Examples

```r
# matrix of auxiliary variables
Xs=cbind(c(1,1,1,1,0,0,0,0,0,0),c(0,0,0,0,1,1,1,1,1,1),c(1,2,3,4,5,6,7,8,9,10))
# inclusion probabilities
pik=rep(0.2,times=10)
# vector of totals
t=c(24,26,280)
# the g-weights
f=rakingratio(Xs,pik,t)
# the calibration estimator is equal to t
checkcalibration(Xs,pik,t,f)
# We change the vector t
f=c(24,26,1000)
checkcalibration(Xs,pik,t,f)
# with this vector t, the calibration is impossible
checkcalibration(Xs,pik,t,f)
```
**cleanstrata**

*Clean strata*

**Description**

Renumbers a variable of stratification (qualitative variable). The strata receive a number from 1 to the last stratum number. The empty strata are suppressed. This function is used in 'balancedstratification'.

**Usage**

```
cleanstrata(strata)
```

**Arguments**

`strata` vector of stratum numbers.

**See Also**

`balancedstratification`

**Examples**

```r
# definition of the stratification variable
strata=c(-2,3,-2,3,4,4,-2,-2,3,4,0,0,0)
# renumber the strata
cleanstrata(strata)
```

---

**cluster**

*Cluster sampling*

**Description**

Cluster sampling with equal/unequal probabilities.

**Usage**

```
cluster(data, clustername, size, method=c("srswor","srswr","poisson","systematic"), pik, description=FALSE)
```
Arguments

data data frame or data matrix; its number of rows is N, the population size.
clustername the name of the clustering variable.
size sample size.
method method to select clusters; the following methods are implemented: simple random sampling without replacement (srswor), simple random sampling with replacement (srswr), Poisson sampling (poisson), systematic sampling (systematic); by default, the method is "srswor".
pik vector of selection probabilities or auxiliary information used to compute them; this argument is needed only for unequal probability sampling (Poisson, systematic). If an auxiliary information is provided, the function uses the inclusionprobabilities function for computing these probabilities. If the method is "srswr" and the sample size is larger than the population size, this vector is normalized to one.
description a message is printed if the value is TRUE; the message gives the number of selected clusters, the number of units in the population and the number of selected units. By default, the value is FALSE.

Details

The cluster object contains the following information: the selected clusters, the identifier of the units in the selected clusters, the final inclusion probabilities for the units (they are equal for the units coming from the same cluster). If method is "srswr", the number of replicates is also given.

See Also

mstage, strata, getdata

Examples

#### Example 1
# Uses the swissmunicipalities data to draw a sample of clusters
data(swissmunicipalities)
# the variable 'REG' has 7 categories in the population; it is used as clustering variable
# the sample size is 3; the method is simple random sampling without replacement
cl=cluster(swissmunicipalities, clusternamewidth=4, size=3, method="srswor")
# extracts the observed data
# the order of the columns is different from the order in the swissmunicipalities database
getdata(swissmunicipalities, cl)
#### Example 2
# the same data as in Example 1
# the sample size is 3; the method is systematic sampling
# the pik vector is randomly generated using the U(0,1) distribution
cl_sys=cluster(swissmunicipalities, clusternamewidth=4, size=3, method="systematic", pik=runif(length(cl$clustername), 0, 1))
# extracts the observed data
**disjunctive**  
*Disjunctive combination*

**Description**

Transforms a categorical variable into a matrix of indicators. The values of the categorical variable are integer numbers (positive or negative).

**Usage**

```r
disjunctive(strata)
```

**Arguments**

- `strata` vector of integer numbers.

**See Also**

`balancedstratification`

**Examples**

```r
# definition of the variable of stratification
strata=c(-2,3,-2,3,4,4,-2,-2,3,4,0,0,0)
# computation of the matrix
disjunctive(strata)
```

---

**fastflightcube**  
*Fast flight phase for the cube method*

**Description**

Executes the fast flight phase of the cube method (algorithm of Chauvet and Tillé, 2005, 2006). The data are sorted following the argument order. Inclusion probabilities equal to 0 or 1 are tolerated.

**Usage**

```r
fastflightcube(X,pik,order=1,comment=TRUE)
```
getdata

Arguments

X  matrix of auxiliary variables on which the sample must be balanced.
pik vector of inclusion probabilities.
order 1, the data are randomly arranged, 2, no change in data order, 3, the data are sorted in decreasing order.
comment a comment is written during the execution if comment is equal to TRUE.

References


See Also

samplecube

Examples

# Matrix of balancing variables
X=cbind(c(1,1,1,1,1,1,1,1,1),c(1,2,3,4,5,6,7,8,9))
# Vector of inclusion probabilities.
# The sample size is 3.
pik=c(1/3,1/3,1/3,1/3,1/3,1/3,1/3,1/3,1/3)
# pikstar is almost a balanced sample and is ready for the landing phase
pikstar=fastflightcube(X,pik,order=1,comment=TRUE)
pikstar

---

**getdata**

Get data

Description

Extracts the observed data from the data frame. The function is used after a sample is drawn.

Usage

getdata(data, m)
inclusionprobabilities

**Arguments**

- `data` : the data frame or data matrix; its number of rows is N, the population size.
- `m` : the vector of the selected units or the sampled data set.

**See Also**

`srswor, UPsystematic, strata, cluster, mstage`

**Examples**

```
# Example 1
# Generates artificial data (a 235X3 matrix with 3 columns: state, region, income).
# The variable 'state' has 2 categories (nc and sc);
# the variable 'region' has 3 categories (1, 2 and 3);
# the variable 'income' is generated using U(0,1) distribution.
data=rbind(matrix(rep("nc",165),165,1,byrow=TRUE),
matrix(rep("sc",70),70,1,byrow=TRUE))
data=cbind.data.frame(data,c(rep(1,100), rep(2,50), rep(3,15), rep(1,30), rep(2,40)),1000*runif(235))
names(data)=c("state","region","income")
# the inclusion probabilities are computed using the variable 'income'
pik=inclusionprobabilities(data$income,20)
# draw a sample s using systematic sampling (sample size is 20)
s=UPsystematic(pik)
# extracts the observed data
getdata(data,s)
```

```
# Example 2
# see other examples in 'strata', 'cluster', 'mstage' help
```

---

**inclusionprobabilities**

**Inclusion probabilities**

**Description**

Computes the first-order inclusion probabilities from a vector of positive numbers (for probability proportional-to-size sampling).

**Usage**

`inclusionprobabilities(a,n)`

**Arguments**

- `a` : vector of positive numbers.
- `n` : sample size.
**inclusionprobastrata**

### Examples

```
############
## Example 1
############
# a vector of positive numbers
a=1:20
# computation of the inclusion probabilities for a sample size n=12
pik=inclusionprobabilities(a,12)
pik

############
## Example 2
############
# Computation of the inclusion probabilities proportional to the number
# of inhabitants in each municipality of the Belgian database.
data(belgianmunicipalities)
pik=inclusionprobabilities(belgianmunicipalities$Tot04,200)
# the first-order inclusion probabilities for each municipality
data.frame(pik=pik,name=belgianmunicipalities$Commune)
# the inclusion probability sum is equal to sample size
sum(pik)
```

---

**inclusionprobastrata**

*Strata inclusion probabilities*

### Description

Computes the inclusion probabilities for a stratified design. The inclusion probabilities are equal in each strata.

### Usage

```
inclusionprobastrata(strat,nh)
```

### Arguments

- **strat**: vector that defines the strata.
- **nh**: vector with the number of units to be selected in each strata.

### See Also

`balancedstratification`
Examples

```r
# the strata
strata=c(1,1,1,1,2,2,2,2,3,3,3,3)
# sample size in each stratum
nh=c(2,3)
inclusionprobastrata(strata,nh)
```

---

**landingcube**  
*Landing phase for the cube method*

---

**Description**

Landing phase of the cube method using linear programming.

**Usage**

```r
landingcube(X, pikstar, pik, comment=TRUE)
```

**Arguments**

- `X`: matrix of auxiliary variables on which the sample must be balanced.
- `pikstar`: vector obtained at the end of the flight phase.
- `pik`: vector of inclusion probabilities.
- `comment`: a comment is written during the execution if `comment` is equal to TRUE.

**References**


**See Also**

`samplecube, fastflightcube`
**Examples**

```r
# matrix of balancing variables
X <- cbind(c(1, 1, 1, 1, 1, 1, 1, 1, 1), c(1.1, 2.2, 3.1, 4.2, 5.1, 6.3, 7.1, 8.1, 9.1))
# Vector of inclusion probabilities
# The sample has the size equal to 3.
pik <- c(1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3)
# pikstar is almost a balanced sample and is ready for the landing phase
pikstar <- fastflightcube(X, pik, order=1, comment=TRUE)
# selection of the sample s
s <- landingcube(X, pikstar, pik, comment=TRUE)
round(s)
```

**Description**

Implements multistage sampling with equal/unequal probabilities.

**Usage**

```r
mstage(data, stage=c("stratified","cluster",""), varnames, size, method=c("srswor","srswr","poisson","systematic"), pik, description=FALSE)
```

**Arguments**

- `data` the data frame or data matrix; its number of rows is N, the population size.
- `stage` list of sampling type at each stage; the possible values are: "stratified", "clustering" and "". For multistage element sampling, this argument is not needed.
- `varnames` list of stratification or clustering variables.
- `size` list of sample sizes (in the order in which the samples appear in the multistage sampling).
- `method` method to select units; the following methods are implemented: simple random sampling without replacement (srswor), simple random sampling with replacement (srswr), Poisson sampling (poisson), systematic sampling (systematic); by default, the method is "srswor".
- `pik` list of selection probabilities or auxiliary information used to compute them; this argument is needed only for unequal probability sampling (Poisson, systematic). If an auxiliary information is provided, the function uses the inclusionprobabilities function for computing these probabilities. If the method is "srswr" and the sample size is larger than the population size, the selection probabilities are normalized to one.
- `description` a message is printed if the value is TRUE; the message gives the number of selected units and the number of the units in the population. By default, its value is FALSE.
Details

The function produces an object, which contains the following information: the selected units at each stage, the identifier of the units, the unit inclusion probabilities at each stage, the final inclusion probabilities of the units. If the method is "srswr", the number of replicates is also given.

See Also

cluster, strata, getdata

Examples

############
## Example 1
############
# Two-stage cluster sampling
# Uses the 'swissmunicipalities' data for drawing a sample of units
data(swissmunicipalities)
# the variable 'REG' (region) has 7 categories;
# it is used as clustering variable in the first-stage sample
# the variable 'CT' (canton) has 26 categories;
# it is used as clustering variable in the second-stage sample
# 4 clusters (regions) are selected in the first-stage
# 1 canton is selected in the second-stage from each sample cluster
# the method is simple random sampling without replacement (equal probability, without replacement)
m=mstage(swissmunicipalities,stage=list("cluster","cluster"), varnames=list("REG","CT"), size=list(4,c(1,1,1,1)), method="srswor")
# extracts the observed data
# the order of the columns is different from the order in the swissmunicipalities database
getdata(swissmunicipalities, m)

############
## Example 2
############
# Two-stage element sampling
# Example from An and Watts (New SAS procedures for Analysis of Sample Survey Data)
# Generates artificial data (a 235x3 matrix with 3 columns: state, region, income).
# The variable "state" has 2 categories ('nc' and 'sc').
# The variable "region" has 3 categories (1, 2 and 3).
# The variable "income" is generated using U(0,1) distribution.
data=rbind(matrix(rep("nc",165),165,1,byrow=TRUE),matrix(rep("sc",70),70,1,byrow=TRUE))
data=cbind.data.frame(data,c(rep(1,100),rep(2,50),rep(3,15),rep(1,30),rep(2,40)),100*runif(235))
names(data)=c("state","region","income")
# the method is simple random sampling with replacement
# 25 units are drawn in the first-stage
# in the second-stage, 10 units are drawn from the already 25 selected units
m=mstage(data,size=list(25,10), method="srswr")
# extracts the observed data
getdata(data,m)

############
## Example 3
############
# One-stage stratified cluster sampling
# The same data as in Example 2
The variable 'state' is used as stratification variable. 20 states are drawn in the first stratum and 10 states in the second stratum. The variable 'region' is used as clustering variable. 1 cluster (region) is drawn in each stratum.

```r
m = mstage(data, stage = list("stratified","cluster"), varnames = list("state","region"), size = list(c(20,10),c(1,1)), method = "srswor")
```

# extracts the observed data
getdata(data,m)

# Example 4

## Two-stage cluster sampling

The same data as in Example 1

```r
data(swissmunicipalities)
```

The method is systematic sampling (unequal probabilities, without replacement). 4 clusters are drawn in the first-stage. 1 cluster from each cluster sample in the second-stage. ls is the list of sizes

```r
ls = list(4,c(1,1,1,1))
```

In the first-stage, the clustering variable is 'REG' (region) with 7 categories. Each region is selected with the probability 1/7.

In the second-stage, the clustering variable is 'CT' (canton) with 26 categories.

In the region 1, there are 3 cantons and each canton is selected with the probability 1/3.

In the region 2, there are 5 cantons and each canton is selected with the probability 1/5.

In the region 3, there are 3 cantons and each canton is selected with the probability 1/3.

In the region 4, there is 1 canton, which it is selected with the probability 1.

In the region 5, there are 7 cantons and each canton is selected with the probability 1/7.

In the region 6, there are 6 cantons and each canton is selected with the probability 1/6.

In the region 7, there is 1 canton, which it is selected with the probability 1.

It is necessary to use a list of selection probabilities at each stage.

```r
prob = list(rep(4/7,7),list(rep(1/3,3),rep(1/5,5),rep(1/3,3),rep(1,1),rep(1/7,7),rep(1/6,6),rep(1,1)))
```

```r
m = mstage(swissmunicipalities,stage = list("cluster","cluster"), varnames = list("REG","CT"), size = list(c(4,1,1,1,1,1,1)), method = "systematic", pik = prob)
```

# extracts the observed data
getdata(swissmunicipalities,m)

---

**rakingratio**

**g-weights of the raking ratio estimator**

**Description**

Computes the g-weights of the raking ratio estimator.

**Usage**

```r
rakingratio(Xs,piks,t,q=rep(1,times=length(piks)))
```
Arguments

- **Xs**: matrix of calibration variables.
- **piks**: vector of inclusion probabilities.
- **t**: vector of population totals.
- **q**: vector of weights for the distance. The variation of the g-weights is reduced for the small values of q. When a component of q is null, the corresponding g-weight remains equal to 1.

References


See Also

- `regressionestimator`, `boundedrakingratio`, `boundedregressionestimator`, `checkcalibration`

Examples

```r
# Example 1
# matrix of auxiliary variables
Xs=cbind(c(1,1,1,1,0,0,0,0,0,0),c(0,0,0,0,1,1,1,1,1,1),c(1,2,3,4,5,6,7,8,9,10))
# inclusion probabilities
piks=rep(0.2,times=10)
# vector of totals
t=c(24,26,280)
# the Horvitz-Thompson estimator
tHT=t(1/piks)%*%Xs
# the g-weights
q=rakingratio(Xs,piks,t)
# the calibration estimator is equal to t
ncal=t(q/piks)%*%Xs
```

```r
# Example 2
# Example of rakingratio, regression, boundedregression and boundedrakingratio estimators,
# with the data of Belgian municipalities.
# Firstly, a sample is selected by means of Poisson sampling.
# Secondly, the g-weights and the calibration estimators are calculated.
# the database of Belgian municipalities
data(belgianmunicipalities)
attach(belgianmunicipalities)
```
X = cbind(
  Men03/mean(Men03),
  Women03/mean(Women03),
  Diffmen,
  Diffwom,
  TaxableIncome/mean(TaxableIncome),
  Totaltaxation/mean(Totaltaxation),
  averageincome/mean(averageincome),
  medianincome/mean(medianincome))

# draw a sample with expected size 200
# using Poisson sampling
# the inclusion probabilities are proportional to the average income
pik = inclusionprobabilities(averageincome, 200)
N = length(pik)  # population size
s = UPpoisson(pik)  # sample selection
Xs = X[s == 1,]  # matrix of calibration variables of the sample
piks = pik[s == 1]  # inclusion probabilities of the sample
n = length(piks)  # sample size

# vector of population totals of the auxiliary variables
t = c(t(rep(1, times = N)) %*% X)
# The Horvitz-Thompson estimators of auxiliary variables
c((1/piks) %*% Xs)
# The true total
t
# Computation of the g-weights
# by means of different calibration methods.
g1 = regressionestimator(Xs, piks, t)
g2 = rakingratio(Xs, piks, t)
g3 = boundedregressionestimator(Xs, piks, t, LOW = 0.5, UP = 1.5)
g4 = boundedrakingratio(Xs, piks, t, LOW = 0.5, UP = 1.5)

# In some cases, the calibration does not exist
# particularly when bounds are used.
# If the calibration is possible, the calibration estimators are the following
if (checkcalibration(Xs, piks, t, g1)) c((g1/piks) %*% Xs) else print("error")
if (checkcalibration(Xs, piks, t, g2)) c((g2/piks) %*% Xs) else print("error")
if (checkcalibration(Xs, piks, t, g3)) c((g3/piks) %*% Xs) else print("error")
if (checkcalibration(Xs, piks, t, g4)) c((g4/piks) %*% Xs) else print("error")

regressionestimator

Description

Computes the g-weights of the regression estimator.

Usage

regressionestimator(Xs, piks, tot, q = rep(1, times = length(piks)))
**regressionestimator**

**Arguments**

- **Xs**: matrix of calibration variables.
- **piks**: vector of inclusion probabilities.
- **tot**: vector of population totals.
- **q**: vector of weights for the distance. The variation of the g-weights is reduced for the small values of q. When a component of q is null, the corresponding g-weight remains equal to 1.

**References**


**See Also**

- rakingratio
- boundedrakingratio
- boundedregressionestimator
- checkcalibration

**Examples**

```
########
## Example 1
########
# matrix of auxiliary variables
Xs=cbind(
  c(1,1,1,1,0,0,0,0,0,0),
  c(0,0,0,0,1,1,1,1,1,1),
  c(1,2,3,4,5,6,7,8,9,10))
# inclusion probabilities
piks=rep(0.2,times=10)
# vector of totals
t=c(24,26,280)
# the Horvitz-Thompson estimator
tHT=t(1/piks)%*%Xs
# the g-weights
g=regressionestimator(Xs,piks,t)
# the calibration estimator is equal to t
tcal=t(g/piks)%*%Xs

########
## Example 2
########
# Example of rakingratio, regression, boundedregression and boundedrakingratio estimators,
# with the data of Belgian municipalities.
# Firstly, a sample is selected by means of Poisson sampling.
# Secondly, the g-weights and the calibration estimators are calculated.
# the database of Belgian municipalities
```
data(belgianmunicipalities)
attach(belgianmunicipalities)
X=cbind(
    Men03/mean(Men03),
    Women03/mean(Women03),
    Diffmen,
    Diffwom,
    TaxableIncome/mean(TaxableIncome),
    Totaltaxation/mean(Totaltaxation),
    averageincome/mean(averageincome),
    medianincome/mean(medianincome)
)
# draw a sample with expected size 200
# using Poisson sampling
# the inclusion probabilities are proportional to the average income
pik=inclusionprobabilities(averageincome,200)
N=length(pik) # population size
s=UPpoisson(pik) # sample
Xs=X[s==1,] # matrix of calibration variables of the sample
piks=pik[s==1] # inclusion probabilities of the sample
n=length(piks) # sample size
# vector of population totals of the auxiliary variables
t=c(t(rep(1,times=N))%*%X)
# The Horvitz-Thompson estimators of auxiliary variables
c((1/piks) %*% Xs)
# The true total

# Computation of the g-weights
# by means of different calibration methods.
g1=regressionestimator(Xs,piks,t)
g2=rakingratio(Xs,piks,t)
g3=boundedregressionestimator(Xs,piks,t,LOW=0.5,UP=1.5)
g4=boundedrakingratio(Xs,piks,t,LOW=0.5,UP=1.5)
# In some cases, the calibration does not exist
# particularly when bounds are used.
if(checkcalibration(Xs,piks,t,g1)) c((g1/piks) %*% Xs) else print("error")
if(checkcalibration(Xs,piks,t,g2)) c((g2/piks) %*% Xs) else print("error")
if(checkcalibration(Xs,piks,t,g3)) c((g3/piks) %*% Xs) else print("error")
if(checkcalibration(Xs,piks,t,g4)) c((g4/piks) %*% Xs) else print("error")

#### Example 3
####
#### Computation of the ratio estimator as
#### a particular case of the regression estimator.
#### The q-weights must be the inverse of the calibration variable.

# data(belgianmunicipalities)
attach(belgianmunicipalities)
pik=inclusionprobabilities(averageincome,200)
X=cbind(
    Men03/mean(Men03),
    Women03/mean(Women03),
    Diffmen,
**samplecube**  

Sample cube method

**Description**

Select a balanced sample (a vector of 0 and 1) or an almost balanced sample. Firstly, the flight phase is applied. Next, if needed, the landing phase is applied on the result of the flight phase.

**Usage**

```r
samplecube(X,pik,order=1,comment=TRUE,method=1)
```

**Arguments**

- `X` matrix of auxiliary variables on which the sample must be balanced.
- `pik` vector of inclusion probabilities.
- `order` 1, the data are randomly arranged, 2, no change in data order, 3, the data are sorted in decreasing order.
- `comment` a comment is written during the execution if comment is equal to TRUE.
- `method` 1, for a landing phase by linear programming, 2, for a landing phase by suppression of variables.

**References**


See Also

landingcube, fastflightcube

Examples

```
### Example 1
# matrix of balancing variables
X=cbind(c(1,1,1,1,1,1,1,1,1),c(1,1,2,2,3,1,4,2,5,1,6,3,7,1,8,1,9,1))
# Vector of inclusion probabilities.
# The sample size is 3.
pik=c(1/3,1/3,1/3,1/3,1/3,1/3,1/3,1/3,1/3)
# Selection of the sample
s=samplecube(X,pik,order=1,comment=TRUE)
# The selected sample
(1:length(pik))[s==1]

### Example 2
# 2 strata and 2 auxiliary variables
# we verify the values of the inclusion probabilities by simulations
X=rbind(c(1,0,1,2),c(1,0,2,5),c(1,0,3,7),c(1,0,4,9),
c(1,0,5,1),c(1,0,6,5),c(1,0,7,7),c(1,0,8,6),c(1,0,9,9),
c(1,0,10,3),c(1,0,11,3),c(1,0,12,2),c(1,0,13,3),
c(0,1,14,6),c(0,1,15,8),c(0,1,16,9),c(0,1,17,1),
c(0,1,18,2),c(0,1,19,3),c(0,1,20,4))
pik=rep(1/2,times=20)
ppp=rep(0,times=20)
sim=100
for(i in 1:sim)
  ppp=ppp+samplecube(X,pik,1,FALSE)
ppp=ppp/sim
print(ppp)
print(pik)

### Example 3
# unequal probability sampling by cube method
# one auxiliary variable equal to the inclusion probability
N=200
pik=runif(N)
pikfin=samplecube(array(pik,c(N,1)),pik,1,TRUE)

### Example 4
# p auxiliary variables generated randomly
```
```r
N=1000
p=7
x=rnorm(N*p,10,3)
# random inclusion probabilities
pik= runif(N)
X=cbind(array(x,c(N,p)),pik)
pikfin=samplecube(X,pik,1,TRUE)

############
## Example 5
############
# strata and an auxiliary variable
N=5000
a=rep(1,times=N)
b=rep(0,times=N)
V1=c(a,b,b)
V2=c(b,a,b)
V3=c(b,b,a)
X=cbind(V1,V2,V3)
pik=rep(2/10,times=3*N)
pikfin=samplecube(X,pik,1,TRUE)

############
## Example 6
############
# Selection of a balanced sample using the MU284 population,
# simulation and comparison of the variance with
# unequal probability sampling of fixed sample size.
############
data(MU284)
# Computation of the inclusion probabilities
pik=inclusionprobabilities(MU284$P75,50)
# Definition of the matrix of balancing variables
X=cbind(MU284$P75,MU284$CS82,MU284$SS82,MU284$S82,MU284$ME84,MU284$REV84)
# Computation of the Horvitz-Thompson estimator for a balanced sample
HTestimator(MU284$RMT85,pik,samplecube(X,pik,1,FALSE))
# Computation of the Horvitz-Thompson estimator for an unequal probability sample
HTestimator(MU284$RMT85,pik,samplecube(matrix(pik),pik,1,FALSE))
# simulations; for a better accuracy, let sim=50
sim=8
res1=rep(0,times=sim)
res2=rep(0,times=sim)
for(i in 1:sim)
{
  cat("Simulation number ",i,"\n")
  res1[i]=HTestimator(MU284$RMT85,pik,samplecube(X,pik,1,FALSE))
  res2[i]=HTestimator(MU284$RMT85,pik,samplecube(matrix(pik),pik,1,FALSE))
}
# summary and boxplots
summary(res1)
summary(res2)
ss=cbind(res1,res2)
colnames(ss) = c("balanced sampling","uneq prob sampling")
boxplot(data.frame(ss), las=1)
```
**srswor**

*Simple random sampling without replacement*

**Description**

Draws a simple random sampling without replacement of size n (equal probabilities, fixed sample size, without replacement).

**Usage**

```
srswor(n, N)
```

**Arguments**

- `n` sample size.
- `N` population size.

**Details**

Returns a vector (with elements 0 and 1) of size N, the population size. Each element k of this vector indicates the status of unit k (1, unit k is selected in the sample; 0, otherwise).

**See Also**

`sample, srswr`

**Examples**

```
# Example 1
s = srswor(3, 10)
(1:10)[s == 1]

# Example 2
n = 200
s = srswor(n, length(Tot))
as.vector(name[s == 1])
```
srswor1  

Selection-rejection method

Description

Draws a simple random sampling without replacement of size \( n \) using the selection-rejection method.

Usage

srswor1(n, N)

Arguments

\( n \)  

sample size.

\( N \)  

population size.

Details

Returns a vector (with elements 0 and 1) of size \( N \), the population size. Each element \( k \) of this vector indicates the status of unit \( k \) (1, unit \( k \) is selected in the sample; 0, otherwise).

References


See Also

srswor

Examples

\[
\begin{align*}
s &= \text{srswor1}(3, 10) \\
\text{#the sample is} \\
(1:10)[s==1]
\end{align*}
\]
srswr

Simple random sampling with replacement

Description

Draws a simple random sampling with replacement of size \( n \) (equal probabilities, fixed sample size, with replacement).

Usage

\( \text{srswr}(n, N) \)

Arguments

\( n \) sample size.
\( N \) population size.

Details

Returns a vector of size \( N \), population size. Each element \( k \) of this vector indicates the number of replicates for unit \( k \) in the sample.

See Also

rmultinom, UPmultinomial

Examples

\[ s = \text{srswr}(3, 10) \]
#the selected units are
\( (1:10)[s != 0] \)
#with the number of replicates
\( s[s != 0] \)

strata

Stratified sampling

Description

Stratified sampling with equal/unequal probabilities.

Usage

\( \text{strata}(\text{data, stratanames=NULL, size, method=c("srswor","srswr","poisson","systematic", pik,description=FALSE}) \)
Arguments

data
stratnames
size
method
pik
description

Details

The function produces an object, which contains the following information: the selected units within strata, the identifier of the units, the final inclusion probabilities of the units. If the method is "srswr", the number of replicates is also given.

See Also

getdata, mstage

Examples

############
## Example 1
############
# Example from An and Watts (New SAS procedures for Analysis of Sample Survey Data)
# Generates artificial data (a 235X3 matrix with 3 columns: state, region, income).
# The variable "state" has 2 categories (nc and sc).
# The variable "region" has 3 categories (1, 2 and 3).
# The sampling frame is stratified by region within state.
data=rbind(matrix(rep("nc",165),165,1,byrow=TRUE),matrix(rep("sc",70),70,1,byrow=TRUE))
data=cbind.data.frame(data,c(rep(1,100), rep(2,50), rep(3,15), rep(1,30),rep(2,40)),1000*runif(235))
names(data)=c("state","region","income")
data=table(data$region,data$state)
# do not run
# nc  sc
# 1 100 30
# 2 50  40
# 3 15  0
# there are 5 cells with non-zero values; one draws 5 samples (1 sample in each stratum)
# the sample stratum sizes are 10,5,10,4,6, respectively
# the method is 'srswor' (equal probability, without replacement)
s=strata(data,c("region","state"),size=c(10,5,10,4,6), method="srswor")
# extracts the observed data
getdata(data,s)

#######
## Example 2
#######
# The same data as in Example 1
# the method is 'systematic' (unequal probability, without replacement)
# the selection probabilities are computed using the variable 'income'
s=strata(data,c("region","state"),size=c(10,5,10,4,6), method="systematic", pik=data$income)
# extracts the observed data
getdata(data,s)

#######
## Example 3
#######
# Uses the 'swissmunicipalities' data for drawing a sample of units
data(swissmunicipalities)
# the variable 'REG' has 7 categories in the population; it is used as stratification variable
# Computes the population stratum sizes
table(swissmunicipalities$REG)
# do not run
# 1  2  3  4  5  6  7
# 589 913 321 171 471 186 245
# the sample stratum sizes are given by size=c(30,20,45,15,20,11,44)
# the method is simple random sampling without replacement (equal probability, without replacement)
st=strata(swissmunicipalities,stratanames=c("REG"),size=c(30,20,45,15,20,11,44), method="srswor")
# extracts the observed data
# the order of the columns is different from the order in the swissmunicipalities database
getdata(swissmunicipalities, st)

---

**swissmunicipalities**  
*The Swiss municipalities population*

**Description**

This population provides information about the Swiss municipalities in 2003.

**Usage**

`data(swissmunicipalities)`

**Format**

A data frame with 2896 observations on the following 22 variables.

**CT**  Swiss canton.
REG   Swiss region.
COM   municipality number.
Nom   municipality name.
HApoly municipality area.
Surfacesbois wood area.
Surfacescult area under cultivation.
Alp   mountain pasture area.
Airbat area with buildings.
Airind industrial area.
P00BMTOT number of men.
P00BWTOT number of women.
Pop020 number of men and women aged between 0 and 20.
Pop2040 number of men and women aged between 20 and 40.
Pop4065 number of men and women aged between 40 and 65.
Pop65P number of men and women aged between 65 and over.
H00PTOT number of households.
H00P01 number of households with 1 person.
H00P02 number of households with 2 persons.
H00P03 number of households with 3 persons.
H00P04 number of households with 4 persons.
POPTOT total population.

Source
Swiss Federal Statistical Office.

Examples

data(swissmunicipalities)
hist(swissmunicipalities$POPTOT)

writesample  All possible samples of fixed size

Description
Gives a matrix whose rows are the vectors (0 or 1) of all samples of fixed size.

Usage
writesample(n,N)
writesample

Arguments

n sample size.
N population size.

See Also

`landingcube`

Examples

```r
# all samples of size 4
# from a population of size 10.
w = writesample(4, 10)
# the samples are
t(apply(w, 1, function(x) (1:ncol(w))[x==1]))
```
Index

*Topic datasets
  belgianmunicipalities, 24
  MU284, 2
  swissmunicipalities, 52

*Topic survey
  balancedcluster, 20
  balancedstratification, 21
  balancedtwostage, 23
  boundedrakingratio, 25
  boundedregressionestimator, 27
  checkcalibration, 30
  cleanstrata, 31
  cluster, 31
  disjunctive, 33
  fastflightcube, 33
  getdata, 34
  HTestimator, 1
  inclusionprobabilities, 35
  inclusionprobastrata, 36
  landingcube, 37
  mstage, 38
  rakingratio, 40
  regressionestimator, 42
  samplecube, 45
  srswor, 48
  srswor1, 49
  srswr, 50
  strata, 50
  UPbrewer, 3
  UPmaxentropy, 4
  UPmidzuno, 6
  UPmidzunopi2, 7
  UPminimalsupport, 8
  UPmultinomial, 9
  UPpivotal, 10
  UPpoisson, 12
  UPrandompivotal, 13
  UPrandomsystematic, 14
  UPSampford, 15
  UPsystematic, 16
  UPsystematicpi2, 17
  UPTtille, 18
  UPTtillepi2, 19
  writesample, 53

balancedcluster, 20, 24
balancedstratification, 21, 24, 31, 33, 36
balancedtwostage, 23
belgianmunicipalities, 24
boundedrakingratio, 25, 30, 41, 43
boundedregressionestimator, 27, 30, 41, 43
checkcalibration, 30, 41, 43
cleanstrata, 31
cluster, 31, 35, 39
disjunctive, 33
fastflightcube, 21, 22, 24, 33, 37, 46
getdata, 32, 34, 39, 51
HTestimator, 1
inclusionprobabilities, 10, 12, 16, 32, 35, 38, 51
inclusionprobastrata, 36, 36
landingcube, 21, 22, 24, 37, 46, 54
mstage, 32, 35, 38, 51
MU284, 2
rakingratio, 30, 40, 43
regressionestimator, 30, 41, 42
rmultinom, 9, 50
INDEX

sample, 48
samplecube, 21, 22, 24, 34, 37, 45
srswor, 35, 48, 49
srswor1, 49
srswr, 48, 50
strata, 32, 35, 39, 50
swissmunicipalities, 52

UPbrewer, 3
UPmaxentropy, 4
UPmaxentropypi2 (UPmaxentropy), 4
UPMEpik2frompikw (UPmaxentropy), 4
UPMEpikfromq (UPmaxentropy), 4
UPMEpiktildefrompik
  (UPmaxentropy), 4
UPMEqfromw (UPmaxentropy), 4
UPMEsfromq (UPmaxentropy), 4
UPmidzuno, 6, 7
UPmidzunopi2, 7
UPminimalsupport, 8
UPmultinomial, 9, 50
UPopips, 10
UPpivotal, 11, 13
UPpoisson, 12
UPrandompivotal, 11, 13
UPrandomsystematic, 14, 16
UPsampford, 15
UPsystematic, 4, 14, 15, 16, 17, 18, 35
UPsystematicpi2, 17
UPtille, 2, 7, 18, 20
UPtillepi2, 19

writesample, 53