

Package: TeachingSampling (via r-universe)

October 19, 2026

Type Package

Title Selection of Samples and Parameter Estimation in Finite Population

Version 4.2.0

Date 2026-06-15

Description Allows the user to draw probabilistic samples and make inferences from a finite population based on several sampling designs, including simple random, systematic, Bernoulli, Poisson, PPS, stratified, and cluster sampling. Provides Horvitz-Thompson, Hansen-Hurwitz, and generalised regression (GREG) estimators of totals, means, ratios, regression coefficients, and quantiles, along with exact and approximate variance estimators.

License GPL (>= 2)

Depends R (>= 3.5), dplyr, magrittr

Encoding UTF-8

NeedsCompilation no

URL <https://github.com/psirusteam/TeachingSampling>

BugReports <https://github.com/psirusteam/TeachingSampling/issues>

Config/roxygen2/version 8.0.0

Repository <https://psirusteam.r-universe.dev>

Date/Publication 2026-07-21 15:24:43 UTC

RemoteUrl <https://github.com/psirusteam/teachingsampling>

RemoteRef HEAD

RemoteSha 93d41a140ec067b80749e840907b7a75e8ba82a1

Contents

BigCity	3
BigLucy	4

Deltakl	6
Domains	7
E.1SI	8
E.2SI	10
E.BE	13
E.Beta	15
E.piPS	17
E.PO	19
E.PPS	20
E.Quantile	21
E.SI	23
E.STpiPS	25
E.STPPS	27
E.STSI	28
E.SY	29
E.Trim	31
E.UC	32
E.WR	36
GREG.SI	37
HH	41
HT	44
Ik	50
IkRS	51
IkWR	52
IPFP	53
kish_allocation	54
Lucy	56
nk	57
OrderWR	58
p.WR	60
Pik	61
PikHol	63
Pikl	64
PikPPS	66
PikSTPPS	67
S.BE	69
S.piPS	71
S.PO	72
S.PPS	74
S.SI	75
S.STpiPS	77
S.STPPS	78
S.STSI	80
S.SY	81
S.WR	83
Support	84
SupportRS	85
SupportWR	86

<i>BigCity</i>	3
----------------	---

T.SIC	88
VarHT	89
VarSYGHT	90
Wk	92

Index	97
--------------	-----------

BigCity	<i>Full Person-level Population Database</i>
---------	--

Description

A data frame corresponding to some socioeconomic variables from 150,266 people of a city in a particular year.

Usage

```
data(BigCity)
```

Format

A data frame with the following variables:

HHID The identifier of the household. It corresponds to an alphanumeric sequence (four letters and five digits).

PersonID The identifier of the person within the household. Note: it is not a unique identifier for the whole population.

Stratum Households are located in geographic strata. There are 119 strata across the city.

PSU Households are clustered in cartographic segments defined as primary sampling units (PSU). There are 1,664 PSU nested within strata.

Zone Segments within strata can be located in urban or rural areas across the city.

Sex Sex of the person.

Age Age of the person.

Income Per capita monthly income.

Expenditure Per capita monthly expenditure.

Employment A person's employment status.

Poverty Indicates whether the person is poor or not, based on income.

MaritalST Marital status of the person.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro@gmail.com>

References

Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[Lucy](#), [BigLucy](#)

Examples

```
data(BigCity)
attach(BigCity)
estima <- data.frame(Income, Expenditure)
# The population totals
colSums(estima)
# Some parameters of interest
table(Poverty, Zone)
xtabs(Income ~ Poverty + Zone)
# Correlations among characteristics of interest
cor(estima)
# Some useful histograms
hist(Income)
hist(Expenditure)
# Some useful plots
boxplot(Income ~ Poverty)
barplot(table(Employment))
pie(table(MaritalST))
```

BigLucy

Full Business Population Database (Extended)

Description

A data frame corresponding to some financial variables of 85,396 industrial companies of a city in a particular fiscal year.

Usage

```
data(BigLucy)
```

Format

A data frame with the following variables:

ID The identifier of the company. It corresponds to an alphanumeric sequence (two letters and three digits).

Ubication The address of the principal office of the company in the city.

Level The industrial companies are discriminated according to the income declared. There are small, medium and big companies.

Zone The country is divided by counties. A company belongs to a particular zone according to its cartographic location.

Income The total amount of a company's earnings in the previous fiscal year.

Employees The total number of persons working for the company in the previous fiscal year.

Taxes The total amount of a company's income tax.

SPAM Indicates if the company uses the Internet and webmail options to make self-propaganda.

ISO Indicates if the company is certified by the International Organization for Standardization.

Years The age of the company.

Segments Cartographic segments by county. A segment comprises on average 10 companies located close to each other.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro@gmail.com>

References

Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[Lucy](#), [BigCity](#)

Examples

```
data(BigLucy)
attach(BigLucy)
# The variables of interest are: Income, Employees and Taxes
estima <- data.frame(Income, Employees, Taxes)
# The population totals
colSums(estima)
# Some parameters of interest
table(SPAM, Level)
xtabs(Income ~ Level + SPAM)
# Correlations among characteristics of interest
cor(estima)
# Some useful histograms
hist(Income)
hist(Taxes)
hist(Employees)
# Some useful plots
boxplot(Income ~ Level)
barplot(table(Level))
pie(table(SPAM))
```

Deltakl	<i>Variance-Covariance Matrix of the Sample Membership Indicators for Fixed Size Without Replacement Sampling Designs</i>
---------	---

Description

Computes the matrix $\Delta_{kl} = \pi_{kl} - \pi_k \pi_l$ for all pairs of units in a finite population. This matrix appears in the exact Horvitz-Thompson variance formula.

Usage

Deltakl(N, n, p)

Arguments

N	Population size. Recommended $N \leq 15$.
n	Sample size.
p	Vector of probabilities for each possible sample in the support. Must sum to 1.

Details

The matrix Δ is central to the Horvitz-Thompson variance estimator:

$$V(\hat{t}_{y,\pi}) = \sum_k \sum_l \Delta_{kl} \frac{y_k}{\pi_k} \frac{y_l}{\pi_l}$$

It requires computing both first-order ([Pik](#)) and second-order ([Pikl](#)) inclusion probabilities, so it is only feasible for small populations.

Value

An $N \times N$ matrix where entry (k, l) equals $\pi_{kl} - \pi_k \pi_l$. Diagonal entries equal $\pi_k(1 - \pi_k)$.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[Pik](#), [Pikl](#), [VarHT](#)

Examples

```

U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
N <- length(U)
n <- 2
p <- c(0.13, 0.2, 0.15, 0.1, 0.15, 0.04, 0.02, 0.06, 0.07, 0.08)
sum(p)
# Variance-Covariance matrix of the sample membership indicators
Deltakl(N, n, p)

```

Domains	<i>Domain Indicator Matrix</i>
---------	--------------------------------

Description

Creates a binary indicator matrix that identifies the domain membership of each unit in the sample. Each column corresponds to one domain (level of y) and each row to one unit.

Usage

```
Domains(y)
```

Arguments

y A vector (factor or coercible to factor) identifying the domain membership of each unit in the sample.

Details

This function is useful for domain estimation, where population totals or means must be estimated for subgroups of the population. The indicator matrix can be multiplied element-wise with the variable of interest to restrict estimation to each domain.

Value

A binary matrix of dimension $n \times D$, where D is the number of domains (levels of y). Entry $(k, d) = 1$ if unit k belongs to domain d , and 0 otherwise. Column names are the domain labels.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also[E.SI](#), [E.STSI](#)**Examples**

```
#####
## Example 1
#####
# This domain contains only two categories: "yes" and "no"
x <- as.factor(c("yes","yes","yes","no","no","no","no","yes","yes"))
Domains(x)

#####
## Example 2
#####
# Uses the Lucy data to draw a random sample of units according
# to a SI design
data(Lucy)
attach(Lucy)

N <- dim(Lucy)[1]
n <- 400
sam <- sample(N,n)
# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
attach(data)
names(data)
# The variable SPAM is a domain of interest
Doma <- Domains(SPAM)
Doma
# HT estimation of the absolute domain size for every category in the domain
# of interest
E.SI(N,n,Doma)

#####
## Example 3
#####
# Following with Example 2...
# The variables of interest are: Income, Employees and Taxes
# This function allows to estimate the population total of this variables for every
# category in the domain of interest SPAM
estima <- data.frame(Income, Employees, Taxes)
SPAM.no <- estima*Doma[,1]
SPAM.yes <- estima*Doma[,2]
E.SI(N,n,SPAM.no)
E.SI(N,n,SPAM.yes)
```

Description

This function computes the Horvitz-Thompson estimator of the population total according to a single stage sampling design.

Usage

```
E.1SI(NI, nI, y, PSU)
```

Arguments

NI	Population size of Primary Sampling Units (PSUs).
nI	Sample size of Primary Sampling Units (PSUs).
y	Vector, matrix or data frame containing the values of the variables of interest for every unit in the selected sample.
PSU	Vector identifying the PSU membership of each unit in the sample.

Details

The estimator aggregates unit values within each selected PSU into cluster totals, then applies the expansion estimator across PSUs. The variance is estimated from the variability of the cluster totals under a simple random sampling framework at the PSU level.

Value

A matrix with four rows and one column per variable of interest:

- Estimation: Estimated population total.
- Standard Error: Estimated standard error of the total.
- CVE: Estimated coefficient of variation (in percentage).
- DEFF: Design effect with respect to simple random sampling.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas

See Also

[E.2SI](#)

Examples

```
data('BigCity')
Households <- BigCity %>% group_by(HHID) %>%
  summarise(Stratum = unique(Stratum),
            PSU = unique(PSU),
            Persons = n(),
            Income = sum(Income),
            Expenditure = sum(Expenditure))

attach(Households)
UI <- levels(as.factor(Households$PSU))
NI <- length(UI)
nI <- 100

samI <- S.SI(NI, nI)
sampleI <- UI[samI]

CityI <- Households[which(Households$PSU %in% sampleI), ]
attach(CityI)
area <- as.factor(CityI$PSU)
estima <- data.frame(CityI$Persons, CityI$Income, CityI$Expenditure)

E.1SI(NI, nI, estima, area)
```

E.2SI	<i>Estimation of the Population Total under Two Stage Simple Random Sampling Without Replacement</i>
-------	--

Description

Computes the Horvitz-Thompson estimator of the population total under a two-stage simple random sampling without replacement design, where both Primary Sampling Units (PSUs) and Secondary Sampling Units (SSUs) are selected by simple random sampling without replacement.

Usage

```
E.2SI(NI, nI, Ni, ni, y, PSU)
```

Arguments

NI	Population size of Primary Sampling Units (PSUs).
nI	Sample size of Primary Sampling Units (PSUs).
Ni	Vector of population sizes of Secondary Sampling Units within each selected PSU.
ni	Vector of sample sizes of Secondary Sampling Units within each selected PSU.
y	Vector, matrix or data frame containing the values of the variables of interest for every unit in the selected sample.
PSU	Vector identifying the PSU membership of each unit in the sample.

Details

The variance estimator decomposes into two components: the between-PSU component and the within-PSU component, following the classical two-stage variance decomposition of Sarndal et al. (1992).

Value

A matrix with four rows and one column per variable of interest:

- Estimation: Estimated population total.
- Standard Error: Estimated standard error of the total.
- CVE: Estimated coefficient of variation (in percentage).
- DEFF: Design effect with respect to simple random sampling.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[E.1SI](#), [E.UC](#)

Examples

```
#####
## Example 1
#####
# Uses Lucy data to draw a twostage simple random sample
# accordind to a 2SI design. Zone is the clustering variable
data(Lucy)
attach(Lucy)
summary(Zone)
# The population of clusters or Primary Sampling Units
UI<-c("A","B","C","D","E")
NI <- length(UI)
# The sample size is nI=3
nI <- 3
# Selects the sample of PSUs
samI<-S.SI(NI,nI)
dataI<-UI[samI]
dataI
# The sampling frame of Secondary Sampling Unit is saved in Lucy1 ... Lucy3
Lucy1<-Lucy[which(Zone==dataI[1]),]
Lucy2<-Lucy[which(Zone==dataI[2]),]
```

```

Lucy3<-Lucy[which(Zone==dataI[3]),]
# The size of every single PSU
N1<-dim(Lucy1)[1]
N2<-dim(Lucy2)[1]
N3<-dim(Lucy3)[1]
Ni<-c(N1,N2,N3)
# The sample size in every PSI is 135 Secondary Sampling Units
n1<-135
n2<-135
n3<-135
ni<-c(n1,n2,n3)
# Selects a sample of Secondary Sampling Units inside the PSUs
sam1<-S.SI(N1,n1)
sam2<-S.SI(N2,n2)
sam3<-S.SI(N3,n3)
# The information about each Secondary Sampling Unit in the PSUs
# is saved in data1 ... data3
data1<-Lucy1[sam1,]
data2<-Lucy2[sam2,]
data3<-Lucy3[sam3,]
# The information about each unit in the final selected sample is saved in data
data<-rbind(data1, data2, data3)
attach(data)
# The clustering variable is Zone
Cluster <- as.factor(as.integer(Zone))
# The variables of interest are: Income, Employees and Taxes
# This information is stored in a data frame called estima
estima <- data.frame(Income, Employees, Taxes)
# Estimation of the Population total
E.2SI(NI,nI,Ni,ni,estima,Cluster)

#####
## Example 2 Total Census to the entire population
#####
# Uses Lucy data to draw a cluster random sample
# according to a SI design ...
# Zone is the clustering variable
data(Lucy)
attach(Lucy)
summary(Zone)
# The population of clusters
UI<-c("A","B","C","D","E")
NI <- length(UI)
# The sample size equals to the population size of PSU
nI <- NI
# Selects every single PSU
samI<-S.SI(NI,nI)
dataI<-UI[samI]
dataI
# The sampling frame of Secondary Sampling Unit is saved in Lucy1 ... Lucy5
Lucy1<-Lucy[which(Zone==dataI[1]),]
Lucy2<-Lucy[which(Zone==dataI[2]),]
Lucy3<-Lucy[which(Zone==dataI[3]),]

```

```

Lucy4<-Lucy[which(Zone==dataI[4]),]
Lucy5<-Lucy[which(Zone==dataI[5]),]
# The size of every single PSU
N1<-dim(Lucy1)[1]
N2<-dim(Lucy2)[1]
N3<-dim(Lucy3)[1]
N4<-dim(Lucy4)[1]
N5<-dim(Lucy5)[1]
Ni<-c(N1,N2,N3,N4,N5)
# The sample size of Secondary Sampling Units equals to the size of each PSU
n1<-N1
n2<-N2
n3<-N3
n4<-N4
n5<-N5
ni<-c(n1,n2,n3,n4,n5)
# Selects every single Secondary Sampling Unit inside the PSU
sam1<-S.SI(N1,n1)
sam2<-S.SI(N2,n2)
sam3<-S.SI(N3,n3)
sam4<-S.SI(N4,n4)
sam5<-S.SI(N5,n5)
# The information about each unit in the cluster is saved in Lucy1 ... Lucy5
data1<-Lucy1[sam1,]
data2<-Lucy2[sam2,]
data3<-Lucy3[sam3,]
data4<-Lucy4[sam4,]
data5<-Lucy5[sam5,]
# The information about each Secondary Sampling Unit
# in the sample (census) is saved in data
data<-rbind(data1, data2, data3, data4, data5)
attach(data)
# The clustering variable is Zone
Cluster <- as.factor(as.integer(Zone))
# The variables of interest are: Income, Employees and Taxes
# This information is stored in a data frame called estima
estima <- data.frame(Income, Employees, Taxes)
# Estimation of the Population total
E.2SI(NI,nI,Ni,ni,estima,Cluster)
# Sampling error is null

```

E.BE

Estimation of the Population Total under Bernoulli Sampling Without Replacement

Description

Computes the Horvitz-Thompson estimator of the population total under a Bernoulli sampling design, where each unit in the population is independently selected with the same inclusion probability.

Usage

```
E.BE(y, prob)
```

Arguments

y	Vector, matrix or data frame containing the values of the variables of interest for every unit in the selected sample.
prob	Scalar. The (constant) inclusion probability used in the Bernoulli sampling design. Must satisfy $0 < \text{prob} \leq 1$.

Details

Under Bernoulli sampling, the sample size is random. The inclusion probability is constant and equal to prob for all units. The variance estimator accounts for the randomness of the sample size.

Value

A matrix with four rows and one column per variable of interest:

- Estimation: Estimated population total.
- Standard Error: Estimated standard error of the total.
- CVE: Estimated coefficient of variation (in percentage).
- DEFF: Design effect with respect to simple random sampling.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*.
 Editorial Universidad Santo Tomas.

See Also

[S.BE](#), [E.SI](#)

Examples

```
data('Lucy')
attach(Lucy)
N <- nrow(Lucy)
prob <- 0.1
sam <- S.BE(N, prob)
sam <- sam[sam != 0]
y <- data.frame(Income = Income[sam], Employees = Employees[sam])
E.BE(y, prob)
```

E.Beta	<i>Estimation of Regression Coefficients under Simple Random Sampling Without Replacement</i>
--------	---

Description

Computes the weighted least squares estimator of regression coefficients for a finite population under simple random sampling without replacement. Both the estimated coefficients and their estimated standard errors are returned.

Usage

```
E.Beta(N, n, y, x, ck = 1, b0 = FALSE)
```

Arguments

N	Population size.
n	Sample size.
y	Vector, matrix or data frame of variables of interest (response).
x	Vector, matrix or data frame of auxiliary variables (predictors).
ck	Optional variance-stabilising constant. Default is 1 (homoscedastic model).
b0	Logical. If TRUE, an intercept column of ones is prepended to x. Default is FALSE.

Details

The estimator uses a working model with weights $V = 1/(\pi_k c_k)$, where $\pi_k = n/N$ under simple random sampling and c_k is an optional variance-stabilising constant. The variance is estimated using the residual-based sandwich approach of Sarndal et al. (1992).

Value

A three-dimensional array with dimensions $[3, P, Q]$, where P is the number of auxiliary variables and Q is the number of variables of interest. The three rows correspond to:

- Beta estimation: Estimated regression coefficient.
- Standard Error: Estimated standard error.
- CVE: Estimated coefficient of variation (in percentage).

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseño de encuestas y estimación de parámetros*. Editorial Universidad Santo Tomas.

See Also

[GREG.SI](#), [E.SI](#)

Examples

```
#####
## Example 1: Linear models involving continuous auxiliary information
#####

# Draws a simple random sample without replacement
data(Lucy)
attach(Lucy)

N <- dim(Lucy)[1]
n <- 400
sam <- S.SI(N, n)
# The information about the units in the sample
# is stored in an object called data
data <- Lucy[sam,]
attach(data)
names(data)

##### common mean model

estima<-data.frame(Income, Employees, Taxes)
x <- rep(1,n)
E.Beta(N, n, estima,x,ck=1,b0=FALSE)

##### common ratio model

estima<-data.frame(Income)
x <- data.frame(Employees)
E.Beta(N, n, estima,x,ck=x,b0=FALSE)

##### Simple regression model without intercept

estima<-data.frame(Income, Employees)
x <- data.frame(Taxes)
E.Beta(N, n, estima,x,ck=1,b0=FALSE)

##### Multiple regression model without intercept

estima<-data.frame(Income)
x <- data.frame(Employees, Taxes)
E.Beta(N, n, estima,x,ck=1,b0=FALSE)
```

```
##### Simple regression model with intercept

estima<-data.frame(Income, Employees)
x <- data.frame(Taxes)
E.Beta(N, n, estima,x,ck=1,b0=TRUE)

##### Multiple regression model with intercept

estima<-data.frame(Income)
x <- data.frame(Employees, Taxes)
E.Beta(N, n, estima,x,ck=1,b0=TRUE)

#####
## Example 2: Linear models with discrete auxiliary information
#####

# Draws a simple random sample without replacement
data(Lucy)
attach(Lucy)

N <- dim(Lucy)[1]
n <- 400
sam <- S.SI(N,n)
# The information about the sample units is stored in an object called data
data <- Lucy[sam,]
attach(data)
names(data)
# The auxiliary information
Doma<-Domains(Level)

##### Poststratified common mean model

estima<-data.frame(Income, Employees, Taxes)
E.Beta(N, n, estima,Doma,ck=1,b0=FALSE)

##### Poststratified common ratio model

estima<-data.frame(Income, Employees)
x<-Doma*Taxes
E.Beta(N, n, estima,x,ck=1,b0=FALSE)
```

E.piPS

Estimation of the Population Total under Probability-Proportional-to-Size Sampling Without Replacement

Description

Computes the Horvitz-Thompson estimator of the population total under a without-replacement probability proportional to size (piPS) sampling design. The variance is estimated using the Horvitz-Thompson variance approximation based on first-order inclusion probabilities.

Usage

```
E.piPS(y, Pik)
```

Arguments

y	Vector, matrix or data frame containing the values of the variables of interest for every unit in the selected sample.
Pik	Vector of first-order inclusion probabilities for each unit in the sample.

Details

When all inclusion probabilities are equal (i.e. $\text{sum}(\text{Pik}) == n$), the variance is set to zero, reflecting an equal-probability design.

Value

A matrix with four rows and one column per variable of interest:

- Estimation: Estimated population total.
- Standard Error: Estimated standard error of the total.
- CVE: Estimated coefficient of variation (in percentage).
- DEFF: Design effect with respect to simple random sampling.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseño de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[S.piPS](#), [PikPPS](#), [E.PO](#)

Examples

```
data('Lucy')
attach(Lucy)
N <- nrow(Lucy)
n <- 400
x <- Employees
res <- S.piPS(n, x)
sam <- res[, 1]
Pik <- res[, 2]
y <- data.frame(Income = Income[sam], Taxes = Taxes[sam])
E.piPS(y, Pik)
```

E.PO	<i>Estimation of the Population Total under Poisson Sampling Without Replacement</i>
------	--

Description

Computes the Horvitz-Thompson estimator of the population total under a Poisson sampling design, where each unit is independently selected with its own inclusion probability.

Usage

E.PO(y, Pik)

Arguments

y	Vector, matrix or data frame containing the values of the variables of interest for every unit in the selected sample.
Pik	Vector of first-order inclusion probabilities for each unit in the sample.

Details

Under Poisson sampling, units are selected independently, so the exact variance of the Horvitz-Thompson estimator has a simple closed form: $V(\hat{t}) = \sum_k (1 - \pi_k)(y_k/\pi_k)^2$.

Value

A matrix with four rows and one column per variable of interest:

- Estimation: Estimated population total.
- Standard Error: Estimated standard error of the total.
- CVE: Estimated coefficient of variation (in percentage).
- DEFF: Design effect with respect to simple random sampling.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[S.PO](#), [E.piPS](#)

Examples

```
data('Lucy')
attach(Lucy)
N <- nrow(Lucy)
n <- 400
Pik <- PikPPS(n, Employees)
sam <- S.PO(N, Pik)
sam <- sam[sam != 0]
y <- data.frame(Income = Income[sam], Taxes = Taxes[sam])
E.PO(y, Pik[sam])
```

E.PPS

Estimation of the Population Total under Probability-Proportional-to-Size Sampling With Replacement

Description

Computes the Hansen-Hurwitz estimator of the population total under a probability proportional to size with-replacement (PPS-WR) sampling design.

Usage

```
E.PPS(y, pk)
```

Arguments

y	Vector, matrix or data frame containing the values of the variables of interest for every selected unit (with possible repetitions).
pk	Vector of selection probabilities for each draw in the sample.

Details

The Hansen-Hurwitz estimator is $\hat{t} = (1/m) \sum_{i=1}^m y_i/p_i$, where p_i is the selection probability of the i -th draw and m is the number of draws.

Value

A matrix with four rows and one column per variable of interest:

- Estimation: Estimated population total.
- Standard Error: Estimated standard error of the total.
- CVE: Estimated coefficient of variation (in percentage).
- DEFF: Design effect with respect to simple random sampling.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[S.PPS](#), [HH](#), [E.piPS](#)

Examples

```
data('Lucy')
attach(Lucy)
m <- 400
res <- S.PPS(m, Employees)
sam <- res[, 1]
pk <- res[, 2]
y <- data.frame(Income = Income[sam], Taxes = Taxes[sam])
E.PPS(y, pk)
```

E.Quantile

Estimation of a Population Quantile

Description

Computes a weighted quantile estimator for finite populations. When inclusion probabilities are provided, the estimator uses the Horvitz-Thompson weights $d_k = 1/\pi_k$; otherwise, equal weights are assumed (simple random sampling).

Usage

```
E.Quantile(y, Qn, Pik)
```

Arguments

y	Vector, matrix or data frame containing the values of the variables of interest for every unit in the selected sample.
Qn	Scalar in (0, 1). The desired quantile level (e.g. 0.5 for the median, 0.25 for the first quartile).
Pik	Optional vector of first-order inclusion probabilities. If omitted, equal probabilities are assumed.

Details

The estimator is based on the weighted empirical cumulative distribution function. For each variable, units are sorted by their observed value, cumulative weights are computed, and the quantile is located by interpolation.

Value

A numeric vector of length equal to the number of variables in y, containing the estimated quantile for each variable.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[E.SI](#), [E.piPS](#)

Examples

```
#####
## Example 1
#####
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
y <- c(32, 34, 46, 89, 35)
x <- c(52, 60, 75, 100, 50)
z <- cbind(y, x)
Pik <- c(0.58, 0.34, 0.48, 0.33, 0.27)
E.Quantile(y, 0.5)
E.Quantile(x, 0.25)
E.Quantile(z, 0.75)
E.Quantile(z, 0.5, Pik)
#####
## Example 2
#####
data(Lucy)
attach(Lucy)
m <- 400
res <- S.PPS(m, Income)
sam <- res[, 1]
pk.s <- res[, 2]
Pik.s <- 1 - (1 - pk.s)^m
data <- Lucy[sam, ]
attach(data)
estima <- data.frame(Income, Employees, Taxes)
E.Quantile(estima, 0.5, Pik.s)
```

E.SI	<i>Estimation of the Population Total under Simple Random Sampling Without Replacement</i>
------	--

Description

Computes the Horvitz-Thompson estimator of the population total under a simple random sampling without replacement (SI) design.

Usage

E.SI(N, n, y)

Arguments

N	Population size.
n	Sample size.
y	Vector, matrix or data frame containing the values of the variables of interest for every unit in the selected sample.

Details

Under simple random sampling without replacement, the Horvitz-Thompson estimator reduces to $\hat{t}_y = N\bar{y}_s$, the expansion estimator. The design effect is always 1 because SI is the reference design.

Value

A matrix with four rows and one column per variable of interest:

- Estimation: Estimated population total.
- Standard Error: Estimated standard error of the total.
- CVE: Estimated coefficient of variation (in percentage).
- DEFF: Design effect (always 1 under SI, included for consistency with other estimators).

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[S.SI](#), [E.STSI](#), [GREG.SI](#)

Examples

```
#####
## Example 1
#####
# Uses the Lucy data to draw a random sample of units according to a SI design
data(Lucy)
attach(Lucy)

N <- dim(Lucy)[1]
n <- 400
sam <- S.SI(N,n)
# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
attach(data)
names(data)
# The variables of interest are: Income, Employees and Taxes
# This information is stored in a data frame called estima
estima <- data.frame(Income, Employees, Taxes)
E.SI(N,n,estima)

#####
## Example 2
#####
# Following with Example 1. The variable SPAM is a domain of interest
Doma <- Domains(SPAM)
# This function allows to estimate the size of each domain in SPAM
estima <- data.frame(Doma)
E.SI(N,n,Doma)

#####
## Example 3
#####
# Following with Example 1. The variable SPAM is a domain of interest
Doma <- Domains(SPAM)
# This function allows to estimate the parameters of the variables of interest
# for every category in the domain SPAM
estima <- data.frame(Income, Employees, Taxes)
SPAM.no <- cbind(Doma[,1], estima*Doma[,1])
SPAM.yes <- cbind(Doma[,1], estima*Doma[,2])
# Before running the following lines, notice that:
# The first column always indicates the population size
# The second column is an estimate of the size of the category in the domain SPAM
# The remaining columns estimates the parameters of interest
# within the corresponding category in the domain SPAM
E.SI(N,n,SPAM.no)
E.SI(N,n,SPAM.yes)

#####
## Example 4
#####
# Following with Example 1. The variable SPAM is a domain of interest
# and the variable ISO is a populational subgroup of interest
```

```

Doma <- Domains(SPAM)
estima <- Domains(Zone)
# Before running the following lines, notice that:
# The first column indicates wheter the unit
# belongs to the first category of SPAM or not
# The remaining columns indicates wheter the unit
# belogns to the categories of Zone
SPAM.no <- data.frame(SpamNO=Doma[,1], Zones=estima*Doma[,1])
# Before running the following lines, notice that:
# The first column indicates wheter the unit
# belongs to the second category of SPAM or not
# The remaining columns indicates wheter the unit
# belogns to the categories of Zone
SPAM.yes <- data.frame(SpamYES=Doma[,2], Zones=estima*Doma[,2])
# Before running the following lines, notice that:
# The first column always indicates the population size
# The second column is an estimate of the size of the
# first category in the domain SPAM
# The remaining columns estimates the size of the categories
# of Zone within the corresponding category of SPAM
# Finnaly, note that the sum of the point estimates of the last
# two columns gives exactly the point estimate in the second column
E.SI(N,n,SPAM.no)
# Before running the following lines, notice that:
# The first column always indicates the population size
# The second column is an estimate of the size of the
# second category in the domain SPAM
# The remaining columns estimates the size of the categories
# of Zone within the corresponding category of SPAM
# Finnaly, note that the sum of the point estimates of the last two
# columns gives exactly the point estimate in the second column
E.SI(N,n,SPAM.yes)

```

E.STpiPS

Estimation of the Population Total under Stratified Probability Proportional to Size Sampling Without Replacement

Description

Computes the Horvitz-Thompson estimator of the population total under a stratified without-replacement probability proportional to size (piPS) sampling design.

Usage

```
E.STpiPS(y, Pik, S)
```

Arguments

y	Vector, matrix or data frame of variables of interest.
Pik	Vector of first-order inclusion probabilities for each unit in the sample.
S	Vector identifying the stratum membership of each unit in the sample.

Value

A matrix with four rows and one column per variable of interest:

- Estimation: Estimated population total.
- Standard Error: Estimated standard error.
- CVE: Estimated coefficient of variation (in percentage).
- DEFF: Design effect with respect to simple random sampling.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseño de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[S.STpiPS](#), [E.piPS](#), [E.STSI](#)

Examples

```
# Uses the Lucy data to draw a stratified random sample
# according to a piPS design in each stratum
data(Lucy)
attach(Lucy)
N1 <- summary(Level)[[1]]
N2 <- summary(Level)[[2]]
N3 <- summary(Level)[[3]]
nh <- c(N1, 100, 200)
S <- Level
x <- Employees
res <- S.STpiPS(S, x, nh)
sam <- res[, 1]
pik <- res[, 2]
data <- Lucy[sam, ]
attach(data)
estima <- data.frame(Income, Employees, Taxes)
E.STpiPS(estima, pik, Level)
```

E.STPPS*Estimation of the Population Total under Stratified Probability Proportional to Size Sampling With Replacement*

Description

Computes the Hansen-Hurwitz estimator of the population total under a stratified PPS with-replacement (STPPS) sampling design.

Usage

E.STPPS(y, pk, mh, S)

Arguments

y	Vector, matrix or data frame of variables of interest.
pk	Vector of selection probabilities for each draw in the sample.
mh	Integer vector with the number of draws within each stratum.
S	Vector identifying the stratum membership of each unit in the sample.

Value

A matrix with four rows and one column per variable of interest:

- Estimation: Estimated population total.
- Standard Error: Estimated standard error.
- CVE: Estimated coefficient of variation (in percentage).
- DEFF: Design effect with respect to simple random sampling.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[S.STPPS](#), [E.PPS](#), [E.STpiPS](#)

Examples

```
# Uses the Lucy data to draw a stratified random sample
# according to a PPS design in each stratum
data(Lucy)
attach(Lucy)
m1 <- 83; m2 <- 100; m3 <- 200
mh <- c(m1, m2, m3)
res <- S.STPPS(Level, Income, mh)
sam <- res[, 1]
pk <- res[, 2]
data <- Lucy[sam, ]
attach(data)
estima <- data.frame(Income, Employees, Taxes)
E.STPPS(estima, pk, mh, Level)
```

E.STSI

Estimation of the Population Total under Stratified Simple Random Sampling Without Replacement

Description

Computes the Horvitz-Thompson estimator of the population total under a stratified simple random sampling without replacement (STSI) design.

Usage

```
E.STSI(S, Nh, nh, y)
```

Arguments

S	Vector identifying the stratum membership of each unit in the sample.
Nh	Integer vector with the population size of each stratum.
nh	Integer vector with the sample size of each stratum.
y	Vector, matrix or data frame of variables of interest.

Value

A matrix with four rows and one column per variable of interest:

- Estimation: Estimated population total.
- Standard Error: Estimated standard error.
- CVE: Estimated coefficient of variation (in percentage).
- DEFF: Design effect with respect to simple random sampling.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseño de encuestas y estimacion de parametros*.
 Editorial Universidad Santo Tomas.

See Also

[S.STSI](#), [E.SI](#), [E.STpiPS](#)

Examples

```
#####
## Example 1
#####
data(Lucy)
attach(Lucy)
N1 <- summary(Level)[[1]]
N2 <- summary(Level)[[2]]
N3 <- summary(Level)[[3]]
Nh <- c(N1, N2, N3)
n1 <- N1; n2 <- 100; n3 <- 200
nh <- c(n1, n2, n3)
sam <- S.STSI(Level, Nh, nh)
data <- Lucy[sam, ]
attach(data)
estima <- data.frame(Income, Employees, Taxes)
E.STSI(Level, Nh, nh, estima)
#####
## Example 2
#####
# The variable SPAM is a domain of interest
Doma <- Domains(SPAM)
SPAM.no <- estima * Doma[, 1]
SPAM.yes <- estima * Doma[, 2]
E.STSI(Level, Nh, nh, Doma)
E.STSI(Level, Nh, nh, SPAM.no)
E.STSI(Level, Nh, nh, SPAM.yes)
```

E.SY

Estimation of the Population Total under Systematic Sampling Without Replacement

Description

Computes the Horvitz-Thompson estimator of the population total under a systematic sampling design with sampling interval a .

Usage

E.SY(N, a, y)

Arguments

N	Population size.
a	Sampling interval (skip). The expected sample size is N/a .
y	Vector, matrix or data frame containing the values of the variables of interest for every unit in the selected sample.

Details

Under systematic sampling the sample size is $n = N/a$. Because only one systematic sample is observed, the variance cannot be estimated without assumptions. Here, the variance is approximated by treating the systematic sample as a simple random sample of the same size, which is a common conservative approximation.

Value

A matrix with four rows and one column per variable of interest:

- Estimation: Estimated population total.
- Standard Error: Estimated standard error of the total.
- CVE: Estimated coefficient of variation (in percentage).
- DEFF: Design effect with respect to simple random sampling.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[S.SY](#), [E.SI](#)

Examples

```
data('Lucy')
attach(Lucy)
N <- nrow(Lucy)
a <- 10
sam <- S.SY(N, a)
y <- data.frame(Income = Income[sam], Taxes = Taxes[sam])
E.SY(N, a, y)
```

E.Trim*Weight Trimming and Redistribution*

Description

This function performs a method of trimming sampling weights based on the evenly redistribution of the net ammount of weight loss among units whose weights were not trimmed. This way, the sum of the timmed sampling weights remains the same as the original weights.

Usage

```
E.Trim(dk, L, U)
```

Arguments

dk	Vector of original sampling weights.
L	Lower bound for weights.
U	Upper bound for weights.

Details

Weights below L or above U are replaced by the respective bound. The net weight lost by trimming is then redistributed evenly among all units whose weights were not trimmed. The process iterates until no weight falls outside [L, U], ensuring that the sum of the trimmed weights equals the sum of the original weights.

Value

A numeric vector of trimmed sampling weights with the same length as dk and the same sum as the original weights.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com> with contributions from Javier Nunez <javier_nunez at inec.gob.ec>

References

Valliant, R. et. al. (2013), *Practical Tools for Designing and Weighing Survey Samples*. Springer.
Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

Examples

```
# Example 1
dk <- c(1, 1, 1, 10)
summary(dk)
L <- 1
U <- 3.5 * median(dk)
dkTrim <- E.Trim(dk, L, U)
sum(dk)
sum(dkTrim)

# Example 2
dk <- rnorm(1000, 10, 10)
L <- 1
U <- 3.5 * median(dk)
dkTrim <- E.Trim(dk, L, U)
sum(dk)
sum(dkTrim)
summary(dk)
summary(dkTrim)
hist(dk)
hist(dkTrim)
```

E.UC

Estimation of the Population Total and its variance using the Ultimate Cluster technique

Description

This function computes a weighted estimator of the population total and estimates its variance by using the Ultimate Cluster technique. This approximation performs well in many sampling designs. The user specifically needs to declare the variables of interest, the primary sampling units, the strata, and the sampling weights for every singlt unit in the sample.

Usage

```
E.UC(S, PSU, dk, y)
```

Arguments

S	Vector identifying the stratum membership of each unit in the selected sample.
PSU	Vector identifying the PSU membership of each unit in the selected sample.
dk	Sampling weights of the units in the sample.
y	Vector, matrix or data frame containing the values of the variables of interest for every unit in the selected sample.

Details

The Ultimate Cluster variance approximation collapses the variance estimation to the first stage of selection. For each stratum, the weighted total contribution of each PSU is computed, and the variance is estimated from the variability among those PSU contributions. This approximation is conservative but robust across many multi-stage designs.

Value

A matrix with four rows and one column per variable of interest:

- Estimation: Estimated population total.
- Standard Error: Estimated standard error using the Ultimate Cluster approximation.
- CVE: Estimated coefficient of variation (in percentage).
- DEFF: Design effect with respect to simple random sampling.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*.
 Editorial Universidad Santo Tomas

See Also

[E.2SI](#)

Examples

```
#####
## Example 1:          ##
## Stratified Two-stage SI ##
#####

data('BigCity')
FrameI <- BigCity %>% group_by(PSU) %>%
  summarise(Stratum = unique(Stratum),
            Persons = n(),
            Income = sum(Income),
            Expenditure = sum(Expenditure))

attach(FrameI)

sizes = FrameI %>% group_by(Stratum) %>%
  summarise(NIh = n(),
            nIh = 2,
            dI = NIh/nIh)
```

```

NIh <- sizes$NIh
nIh <- sizes$nIh

samI <- S.STSI(Stratum, NIh, nIh)
UI <- levels(as.factor(FrameI$PSU))
sampleI <- UI[samI]

FrameII <- left_join(sizes, BigCity[which(BigCity$PSU %in% sampleI), ])
attach(FrameII)

HHdb <- FrameII %>%
  group_by(PSU) %>%
  summarise(Ni = length(unique(HHID)))

Ni <- as.numeric(HHdb$Ni)
ni <- ceiling(Ni * 0.1)
ni
sum(ni)

sam = S.SI(Ni[1], ni[1])
clusterII = FrameII[which(FrameII$PSU == sampleI[1]), ]
sam.HH <- data.frame(HHID = unique(clusterII$HHID)[sam])
clusterHH <- left_join(sam.HH, clusterII, by = "HHID")
clusterHH$dki <- Ni[1]/ni[1]
clusterHH$dk <- clusterHH$dI * clusterHH$dki
data = clusterHH
for (i in 2:length(Ni)) {
  sam = S.SI(Ni[i], ni[i])
  clusterII = FrameII[which(FrameII$PSU == sampleI[i]), ]
  sam.HH <- data.frame(HHID = unique(clusterII$HHID)[sam])
  clusterHH <- left_join(sam.HH, clusterII, by = "HHID")
  clusterHH$dki <- Ni[i]/ni[i]
  clusterHH$dk <- clusterHH$dI * clusterHH$dki
  data1 = clusterHH
  data = rbind(data, data1)
}

sum(data$dk)
attach(data)
estima <- data.frame(Income, Expenditure)
area <- as.factor(PSU)
stratum <- as.factor(Stratum)

E.UC(stratum, area, dk, estima)

#####
## Example 2: ##
## Self weighted Two-stage SI ##
#####

data('BigCity')
FrameI <- BigCity %>% group_by(PSU) %>%
  summarise(Stratum = unique(Stratum),

```

```

        Households = length(unique(HHID)),
        Income = sum(Income),
        Expenditure = sum(Expenditure))

attach(FrameI)

sizes = FrameI %>% group_by(Stratum) %>%
  summarise(NIh = n(),
            nIh = 2)

NIh <- sizes$NIh
nIh <- sizes$nIh

resI <- S.STpiPS(Stratum, Households, nIh)
head(resI)
samI <- resI[, 1]
piI <- resI[, 2]
UI <- levels(as.factor(FrameI$PSU))
sampleI <- data.frame(PSU = UI[samI], dI = 1/piI)

FrameII <- left_join(sampleI,
                    BigCity[which(BigCity$PSU %in% sampleI[,1]), ], )

attach(FrameII)

HHdb <- FrameII %>%
  group_by(PSU) %>%
  summarise(Ni = length(unique(HHID)))
Ni <- as.numeric(HHdb$Ni)
ni <- 5

sam = S.SI(Ni[1], ni)
clusterII = FrameII[which(FrameII$PSU == sampleI$PSU[1]), ]
sam.HH <- data.frame(HHID = unique(clusterII$HHID)[sam])
clusterHH <- left_join(sam.HH, clusterII, by = "HHID")
clusterHH$dkI <- Ni[1]/ni
clusterHH$dk <- clusterHH$dI * clusterHH$dkI
data = clusterHH
for (i in 2:length(Ni)) {
  sam = S.SI(Ni[i], ni)
  clusterII = FrameII[which(FrameII$PSU == sampleI$PSU[i]), ]
  sam.HH <- data.frame(HHID = unique(clusterII$HHID)[sam])
  clusterHH <- left_join(sam.HH, clusterII, by = "HHID")
  clusterHH$dkI <- Ni[i]/ni
  clusterHH$dk <- clusterHH$dI * clusterHH$dkI
  data1 = clusterHH
  data = rbind(data, data1)
}

sum(data$dk)
attach(data)
estima <- data.frame(Income, Expenditure)
area <- as.factor(PSU)

```

```
stratum <- as.factor(Stratum)
E.UC(stratum, area, dk, estima)
```

E.WR	<i>Estimation of the Population Total under Simple Random Sampling With Replacement</i>
------	---

Description

Computes the Hansen-Hurwitz estimator of the population total under a simple random sampling with replacement (WR) design.

Usage

```
E.WR(N, m, y)
```

Arguments

N	Population size.
m	Number of draws (sample size with replacement).
y	Vector, matrix or data frame containing the values of the variables of interest for every draw in the sample (repetitions allowed).

Details

Under simple random sampling with replacement with m draws, the Hansen-Hurwitz estimator is $\hat{t} = (N/m) \sum_{i=1}^m y_i$.

Value

A matrix with four rows and one column per variable of interest:

- Estimation: Estimated population total.
- Standard Error: Estimated standard error of the total.
- CVE: Estimated coefficient of variation (in percentage).
- DEFF: Design effect with respect to simple random sampling without replacement.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also[S.WR](#), [E.SI](#)**Examples**

```
data('Lucy')
attach(Lucy)
N <- nrow(Lucy)
m <- 400
sam <- S.WR(N, m)
y <- data.frame(Income = Income[sam], Taxes = Taxes[sam])
E.WR(N, m, y)
```

GREG.SI

*Generalised Regression Estimator under Simple Random Sampling***Description**

Computes the Generalised Regression (GREG) estimator of the population total under simple random sampling without replacement, using known population totals of auxiliary variables to improve efficiency.

Usage

```
GREG.SI(N, n, y, x, tx, b, b0 = FALSE)
```

Arguments

N	Population size.
n	Sample size.
y	Vector, matrix or data frame of variables of interest.
x	Vector, matrix or data frame of auxiliary variables observed in the sample.
tx	Vector of known population totals for the auxiliary variables.
b	Matrix of regression coefficients (e.g. from E.Beta).
b0	Logical. If TRUE, an intercept column is prepended to x. Default is FALSE.

Details

The GREG estimator is:

$$\hat{t}_{GREG} = \hat{t}_{HT} + (\mathbf{t}_x - \hat{\mathbf{t}}_{x,HT})^T \hat{\boldsymbol{\beta}}$$

where $\hat{\boldsymbol{\beta}}$ are the regression coefficients estimated from the sample, $\mathbf{t}_x$ are the known population totals, and variance is estimated from the residuals.

Value

A matrix with three rows and one column per variable of interest:

- Estimation: GREG estimated population total.
- Standard Error: Estimated standard error.
- CVE: Estimated coefficient of variation (in percentage).

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*.
 Editorial Universidad Santo Tomas.

See Also

[E.Beta](#), [E.SI](#), [Wk](#)

Examples

```
#####
## Example 1: Linear models involving continuous auxiliary information
#####

# Draws a simple random sample without replacement
data(Lucy)
attach(Lucy)

N <- dim(Lucy)[1]
n <- 400
sam <- S.SI(N,n)
# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
attach(data)
names(data)

##### common mean model

estima<-data.frame(Income, Employees, Taxes)
x <- rep(1,n)
model <- E.Beta(N, n, estima, x, ck=1,b0=FALSE)
b <- t(as.matrix(model[1,,]))
tx <- c(N)
GREG.SI(N,n,estima,x,tx, b, b0=FALSE)

##### common ratio model

estima<-data.frame(Income)
```

```

x <- data.frame(Employees)
model <- E.Beta(N, n, estima, x, ck=x,b0=FALSE)
b <- t(as.matrix(model[1,,]))
tx <- sum(Lucy$Employees)
GREG.SI(N,n,estima,x,tx, b, b0=FALSE)

##### Simple regression model without intercept

estima<-data.frame(Income, Employees)
x <- data.frame(Taxes)
model <- E.Beta(N, n, estima, x, ck=1,b0=FALSE)
b <- t(as.matrix(model[1,,]))
tx <- sum(Lucy$Taxes)
GREG.SI(N,n,estima,x,tx, b, b0=FALSE)

##### Multiple regression model without intercept

estima<-data.frame(Income)
x <- data.frame(Employees, Taxes)
model <- E.Beta(N, n, estima, x, ck=1, b0=FALSE)
b <- as.matrix(model[1,,])
tx <- c(sum(Lucy$Employees), sum(Lucy$Taxes))
GREG.SI(N,n,estima,x,tx, b, b0=FALSE)

##### Simple regression model with intercept

estima<-data.frame(Income, Employees)
x <- data.frame(Taxes)
model <- E.Beta(N, n, estima, x, ck=1,b0=TRUE)
b <- as.matrix(model[1,,])
tx <- c(N, sum(Lucy$Taxes))
GREG.SI(N,n,estima,x,tx, b, b0=TRUE)

##### Multiple regression model with intercept

estima<-data.frame(Income)
x <- data.frame(Employees, Taxes)
model <- E.Beta(N, n, estima, x, ck=1,b0=TRUE)
b <- as.matrix(model[1,,])
tx <- c(N, sum(Lucy$Employees), sum(Lucy$Taxes))
GREG.SI(N,n,estima,x,tx, b, b0=TRUE)

#####
## Example 2: Linear models with discrete auxiliary information
#####

# Draws a simple random sample without replacement
data(Lucy)

N <- dim(Lucy)[1]
n <- 400
sam <- S.SI(N,n)
# The information about the units in the sample is stored in an object called data

```

```

data <- Lucy[sam,]
attach(data)
names(data)

# The auxiliary information is discrete type
Doma<-Domains(Level)

##### Poststratified common mean model

estima<-data.frame(Income, Employees, Taxes)
model <- E.Beta(N, n, estima, Doma, ck=1,b0=FALSE)
b <- t(as.matrix(model[1,,]))
tx <- colSums(Domains(Lucy$Level))
GREG.SI(N,n,estima,Doma,tx, b, b0=FALSE)

##### Poststratified common ratio model

estima<-data.frame(Income, Employees)
x <- Doma*Taxes
model <- E.Beta(N, n, estima, x ,ck=1,b0=FALSE)
b <- as.matrix(model[1,,])
tx <- colSums(Domains(Lucy$Level)*Lucy$Taxes)
GREG.SI(N,n,estima,x,tx, b, b0=FALSE)

#####
## Example 3: Domains estimation through the postestratified estimator
#####

# Draws a simple random sample without replacement
data(Lucy)

N <- dim(Lucy)[1]
n <- 400
sam <- S.SI(N,n)
# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
attach(data)
names(data)

# The auxiliary information is discrete type
Doma<-Domains(Level)

##### Poststratified common mean model for the
# Income total in each poststratum #####

estima<-Doma*Income
model <- E.Beta(N, n, estima, Doma, ck=1, b0=FALSE)
b <- t(as.matrix(model[1,,]))
tx <- colSums(Domains(Lucy$Level))
GREG.SI(N,n,estima,Doma,tx, b, b0=FALSE)

##### Poststratified common mean model for the
# Employees total in each poststratum #####

```

```

estima<-Doma*Employees
model <- E.Beta(N, n, estima, Doma, ck=1,b0=FALSE)
b <- t(as.matrix(model[1,,]))
tx <- colSums(Domains(Lucy$Level))
GREG.SI(N,n,estima,Doma,tx, b, b0=FALSE)

##### Poststratified common mean model for the
# Taxes total in each poststratum #####

estima<-Doma*Taxes
model <- E.Beta(N, n, estima, Doma, ck=1, b0=FALSE)
b <- t(as.matrix(model[1,,]))
tx <- colSums(Domains(Lucy$Level))
GREG.SI(N,n,estima,Doma,tx, b, b0=FALSE)

```

HH

*Hansen-Hurwitz Estimator of the Population Total***Description**

Computes the Hansen-Hurwitz (HH) estimator of the population total under a with-replacement sampling design, given the sample observations and their selection probabilities.

Usage

```
HH(y, pk)
```

Arguments

y	Vector or matrix of values of the variable(s) of interest for units in the sample (with possible repetitions).
pk	Vector of selection probabilities for each draw in the sample.

Details

The Hansen-Hurwitz estimator is:

$$\hat{t}_{HH} = \frac{1}{m} \sum_{i=1}^m \frac{y_i}{p_i}$$

where p_i is the selection probability of the i -th draw and m is the number of draws. This estimator is design-unbiased under any with-replacement sampling design.

Value

A numeric vector or matrix with the estimated total for each variable of interest.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

- Hansen, M.H. and Hurwitz, W.N. (1943). On the theory of sampling from finite populations. *Annals of Mathematical Statistics*, 14, 333-362.
- Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.

See Also

[E.PPS](#), [HT](#), [S.PPS](#)

Examples

```
#####
## Example 1
#####
# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# Vectors y1 and y2 give the values of the variables of interest
y1<-c(32, 34, 46, 89, 35)
y2<-c(1,1,1,0,0)
y3<-cbind(y1,y2)
# The population size is N=5
N <- length(U)
# The sample size is m=2
m <- 2
# pk is the probability of selection of every single unit
pk <- c(0.35, 0.225, 0.175, 0.125, 0.125)
# Selection of a random sample with replacement
sam <- sample(5,2, replace=TRUE, prob=pk)
# The selected sample is
U[sam]
# The values of the variables of interest for the units in the sample
y1[sam]
y2[sam]
y3[sam,]
# The Hansen-Hurwitz estimator
HH(y1[sam],pk[sam])
HH(y2[sam],pk[sam])
HH(y3[sam,],pk[sam])

#####
## Example 2
#####
# Uses the Lucy data to draw a simple random sample with replacement
data(Lucy)
attach(Lucy)

N <- dim(Lucy)[1]
m <- 400
sam <- sample(N,m,replace=TRUE)
# The vector of selection probabilities of units in the sample
pk <- rep(1/N,m)
```

```

# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
attach(data)
names(data)
# The variables of interest are: Income, Employees and Taxes
# This information is stored in a data frame called estima
estima <- data.frame(Income, Employees, Taxes)
HH(estima, pk)

#####
## Example 3 HH is unbiased for with replacement sampling designs
#####

# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# Vector y1 and y2 are the values of the variables of interest
y<-c(32, 34, 46, 89, 35)
# The population size is N=5
N <- length(U)
# The sample size is m=2
m <- 2
# pk is the probability of selection of every single unit
pk <- c(0.35, 0.225, 0.175, 0.125, 0.125)
# p is the probability of selection of every possible sample
p <- p.WR(N,m,pk)
p
sum(p)
# The sample membership matrix for random size without replacement sampling designs
Ind <- nk(N,m)
Ind
# The support with the values of the elements
Qy <- SupportWR(N,m, ID=y)
Qy
# The support with the values of the elements
Qp <- SupportWR(N,m, ID=pk)
Qp
# The HT estimates for every single sample in the support
HH1 <- HH(Qy[1,], Qp[1,])[1,]
HH2 <- HH(Qy[2,], Qp[2,])[1,]
HH3 <- HH(Qy[3,], Qp[3,])[1,]
HH4 <- HH(Qy[4,], Qp[4,])[1,]
HH5 <- HH(Qy[5,], Qp[5,])[1,]
HH6 <- HH(Qy[6,], Qp[6,])[1,]
HH7 <- HH(Qy[7,], Qp[7,])[1,]
HH8 <- HH(Qy[8,], Qp[8,])[1,]
HH9 <- HH(Qy[9,], Qp[9,])[1,]
HH10 <- HH(Qy[10,], Qp[10,])[1,]
HH11 <- HH(Qy[11,], Qp[11,])[1,]
HH12 <- HH(Qy[12,], Qp[12,])[1,]
HH13 <- HH(Qy[13,], Qp[13,])[1,]
HH14 <- HH(Qy[14,], Qp[14,])[1,]
HH15 <- HH(Qy[15,], Qp[15,])[1,]
# The HT estimates arranged in a vector

```

```

Est <- c(HH1, HH2, HH3, HH4, HH5, HH6, HH7, HH8, HH9, HH10, HH11, HH12, HH13,
HH14, HH15)
Est
# The HT is actually design-unbiased
data.frame(Ind, Est, p)
sum(Est*p)
sum(y)

```

HT	<i>Horvitz-Thompson Estimator of the Population Total</i>
----	---

Description

Computes the Horvitz-Thompson (HT) estimator of the population total for one or more variables of interest, given the sample observations and their first-order inclusion probabilities.

Usage

```
HT(y, Pik)
```

Arguments

y	Vector or matrix of values of the variable(s) of interest for units in the sample.
Pik	Vector of first-order inclusion probabilities for each unit in the sample.

Details

The Horvitz-Thompson estimator is defined as:

$$\hat{t}_{y,\pi} = \sum_{k \in s} \frac{y_k}{\pi_k}$$

where π_k is the first-order inclusion probability of unit k . This estimator is design-unbiased for any fixed-size sampling design.

Value

A numeric vector or matrix with the estimated total for each variable of interest.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Horvitz, D.G. and Thompson, D.J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47, 663-685.
 Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.

See Also

[VarHT](#), [E.SI](#), [E.piPS](#)

Examples

```
#####
## Example 1
#####
# Uses the Lucy data to draw a simple random sample without replacement
data(Lucy)
attach(Lucy)

N <- dim(Lucy)[1]
n <- 400
sam <- sample(N,n)
# The vector of inclusion probabilities for each unit in the sample
pik <- rep(n/N,n)
# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
attach(data)
names(data)
# The variables of interest are: Income, Employees and Taxes
# This information is stored in a data frame called estima
estima <- data.frame(Income, Employees, Taxes)
HT(estima, pik)

#####
## Example 2
#####
# Uses the Lucy data to draw a simple random sample with replacement
data(Lucy)

N <- dim(Lucy)[1]
m <- 400
sam <- sample(N,m,replace=TRUE)
# The vector of selection probabilities of units in the sample
pk <- rep(1/N,m)
# Computation of the inclusion probabilities
pik <- 1-(1-pk)^m
# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
attach(data)
names(data)
# The variables of interest are: Income, Employees and Taxes
# This information is stored in a data frame called estima
estima <- data.frame(Income, Employees, Taxes)
HT(estima, pik)

#####
## Example 3
#####
# Without replacement sampling
```

```

# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# Vector y1 and y2 are the values of the variables of interest
y1<-c(32, 34, 46, 89, 35)
y2<-c(1,1,1,0,0)
y3<-cbind(y1,y2)
# The population size is N=5
N <- length(U)
# The sample size is n=2
n <- 2
# The sample membership matrix for fixed size without replacement sampling designs
Ind <- Ik(N,n)
# p is the probability of selection of every possible sample
p <- c(0.13, 0.2, 0.15, 0.1, 0.15, 0.04, 0.02, 0.06, 0.07, 0.08)
# Computation of the inclusion probabilities
inclusion <- Pik(p, Ind)
# Selection of a random sample
sam <- sample(5,2)
# The selected sample
U[sam]
# The inclusion probabilities for these two units
inclusion[sam]
# The values of the variables of interest for the units in the sample
y1[sam]
y2[sam]
y3[sam,]
# The Horvitz-Thompson estimator
HT(y1[sam],inclusion[sam])
HT(y2[sam],inclusion[sam])
HT(y3[sam,],inclusion[sam])

#####
## Example 4
#####
# Following Example 3... With replacement sampling
# The population size is N=5
N <- length(U)
# The sample size is m=2
m <- 2
# pk is the probability of selection of every single unit
pk <- c(0.9, 0.025, 0.025, 0.025, 0.025)
# Computation of the inclusion probabilities
pik <- 1-(1-pk)^m
# Selection of a random sample with replacement
sam <- sample(5,2, replace=TRUE, prob=pk)
# The selected sample
U[sam]
# The inclusion probabilities for these two units
inclusion[sam]
# The values of the variables of interest for the units in the sample
y1[sam]
y2[sam]
y3[sam,]

```

```

# The Horvitz-Thompson estimator
HT(y1[sam],inclusion[sam])
HT(y2[sam],inclusion[sam])
HT(y3[sam,],inclusion[sam])

#####
## Example 5 HT is unbiased for without replacement sampling designs
##               Fixed sample size
#####

# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# Vector y1 and y2 are the values of the variables of interest
y<-c(32, 34, 46, 89, 35)
# The population size is N=5
N <- length(U)
# The sample size is n=2
n <- 2
# The sample membership matrix for fixed size without replacement sampling designs
Ind <- Ik(N,n)
Ind
# p is the probability of selection of every possible sample
p <- c(0.13, 0.2, 0.15, 0.1, 0.15, 0.04, 0.02, 0.06, 0.07, 0.08)
sum(p)
# Computation of the inclusion probabilities
inclusion <- Pik(p, Ind)
inclusion
sum(inclusion)
# The support with the values of the elements
Qy <-Support(N,n,ID=y)
Qy
# The HT estimates for every single sample in the support
HT1<- HT(y[Ind[1,]==1], inclusion[Ind[1,]==1])
HT2<- HT(y[Ind[2,]==1], inclusion[Ind[2,]==1])
HT3<- HT(y[Ind[3,]==1], inclusion[Ind[3,]==1])
HT4<- HT(y[Ind[4,]==1], inclusion[Ind[4,]==1])
HT5<- HT(y[Ind[5,]==1], inclusion[Ind[5,]==1])
HT6<- HT(y[Ind[6,]==1], inclusion[Ind[6,]==1])
HT7<- HT(y[Ind[7,]==1], inclusion[Ind[7,]==1])
HT8<- HT(y[Ind[8,]==1], inclusion[Ind[8,]==1])
HT9<- HT(y[Ind[9,]==1], inclusion[Ind[9,]==1])
HT10<- HT(y[Ind[10,]==1], inclusion[Ind[10,]==1])
# The HT estimates arranged in a vector
Est <- c(HT1, HT2, HT3, HT4, HT5, HT6, HT7, HT8, HT9, HT10)
Est
# The HT is actually design-unbiased
data.frame(Ind, Est, p)
sum(Est*p)
sum(y)

#####
## Example 6 HT is unbiased for without replacement sampling designs
##               Random sample size

```

```
#####

# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# Vector y1 and y2 are the values of the variables of interest
y<-c(32, 34, 46, 89, 35)
# The population size is N=5
N <- length(U)
# The sample membership matrix for random size without replacement sampling designs
Ind <- IkRS(N)
Ind
# p is the probability of selection of every possible sample
p <- c(0.59049, 0.06561, 0.06561, 0.06561, 0.06561, 0.06561, 0.00729, 0.00729,
      0.00729, 0.00729, 0.00729, 0.00729, 0.00729, 0.00729, 0.00729, 0.00081,
      0.00081, 0.00081, 0.00081, 0.00081, 0.00081, 0.00081, 0.00081, 0.00081,
      0.00009, 0.00009, 0.00009, 0.00009, 0.00009, 0.00001)

sum(p)
# Computation of the inclusion probabilities
inclusion <- Pik(p, Ind)
inclusion
sum(inclusion)
# The support with the values of the elements
Qy <-SupportRS(N, ID=y)
Qy
# The HT estimates for every single sample in the support
HT1<- HT(y[Ind[1,]==1], inclusion[Ind[1,]==1])
HT2<- HT(y[Ind[2,]==1], inclusion[Ind[2,]==1])
HT3<- HT(y[Ind[3,]==1], inclusion[Ind[3,]==1])
HT4<- HT(y[Ind[4,]==1], inclusion[Ind[4,]==1])
HT5<- HT(y[Ind[5,]==1], inclusion[Ind[5,]==1])
HT6<- HT(y[Ind[6,]==1], inclusion[Ind[6,]==1])
HT7<- HT(y[Ind[7,]==1], inclusion[Ind[7,]==1])
HT8<- HT(y[Ind[8,]==1], inclusion[Ind[8,]==1])
HT9<- HT(y[Ind[9,]==1], inclusion[Ind[9,]==1])
HT10<- HT(y[Ind[10,]==1], inclusion[Ind[10,]==1])
HT11<- HT(y[Ind[11,]==1], inclusion[Ind[11,]==1])
HT12<- HT(y[Ind[12,]==1], inclusion[Ind[12,]==1])
HT13<- HT(y[Ind[13,]==1], inclusion[Ind[13,]==1])
HT14<- HT(y[Ind[14,]==1], inclusion[Ind[14,]==1])
HT15<- HT(y[Ind[15,]==1], inclusion[Ind[15,]==1])
HT16<- HT(y[Ind[16,]==1], inclusion[Ind[16,]==1])
HT17<- HT(y[Ind[17,]==1], inclusion[Ind[17,]==1])
HT18<- HT(y[Ind[18,]==1], inclusion[Ind[18,]==1])
HT19<- HT(y[Ind[19,]==1], inclusion[Ind[19,]==1])
HT20<- HT(y[Ind[20,]==1], inclusion[Ind[20,]==1])
HT21<- HT(y[Ind[21,]==1], inclusion[Ind[21,]==1])
HT22<- HT(y[Ind[22,]==1], inclusion[Ind[22,]==1])
HT23<- HT(y[Ind[23,]==1], inclusion[Ind[23,]==1])
HT24<- HT(y[Ind[24,]==1], inclusion[Ind[24,]==1])
HT25<- HT(y[Ind[25,]==1], inclusion[Ind[25,]==1])
HT26<- HT(y[Ind[26,]==1], inclusion[Ind[26,]==1])
HT27<- HT(y[Ind[27,]==1], inclusion[Ind[27,]==1])
HT28<- HT(y[Ind[28,]==1], inclusion[Ind[28,]==1])
```

```

HT29<- HT(y[Ind[29,]==1], inclusion[Ind[29,]==1])
HT30<- HT(y[Ind[30,]==1], inclusion[Ind[30,]==1])
HT31<- HT(y[Ind[31,]==1], inclusion[Ind[31,]==1])
HT32<- HT(y[Ind[32,]==1], inclusion[Ind[32,]==1])
# The HT estimates arranged in a vector
Est <- c(HT1, HT2, HT3, HT4, HT5, HT6, HT7, HT8, HT9, HT10, HT11, HT12, HT13,
        HT14, HT15, HT16, HT17, HT18, HT19, HT20, HT21, HT22, HT23, HT24, HT25, HT26,
        HT27, HT28, HT29, HT30, HT31, HT32)

Est
# The HT is actually design-unbiased
data.frame(Ind, Est, p)
sum(Est*p)
sum(y)

#####
## Example 7 HT is unbiased for with replacement sampling designs
#####

# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# Vector y1 and y2 are the values of the variables of interest
y<-c(32, 34, 46, 89, 35)
# The population size is N=5
N <- length(U)
# The sample size is m=2
m <- 2
# pk is the probability of selection of every single unit
pk <- c(0.35, 0.225, 0.175, 0.125, 0.125)
# p is the probability of selection of every possible sample
p <- p.WR(N,m,pk)
p
sum(p)
# The sample membership matrix for random size without replacement sampling designs
Ind <- IkWR(N,m)
Ind
# The support with the values of the elements
Qy <- SupportWR(N,m, ID=y)
Qy
# Computation of the inclusion probabilities
pik <- 1-(1-pk)^m
pik
# The HT estimates for every single sample in the support
HT1 <- HT(y[Ind[1,]==1], pik[Ind[1,]==1])
HT2 <- HT(y[Ind[2,]==1], pik[Ind[2,]==1])
HT3 <- HT(y[Ind[3,]==1], pik[Ind[3,]==1])
HT4 <- HT(y[Ind[4,]==1], pik[Ind[4,]==1])
HT5 <- HT(y[Ind[5,]==1], pik[Ind[5,]==1])
HT6 <- HT(y[Ind[6,]==1], pik[Ind[6,]==1])
HT7 <- HT(y[Ind[7,]==1], pik[Ind[7,]==1])
HT8 <- HT(y[Ind[8,]==1], pik[Ind[8,]==1])
HT9 <- HT(y[Ind[9,]==1], pik[Ind[9,]==1])
HT10 <- HT(y[Ind[10,]==1], pik[Ind[10,]==1])
HT11 <- HT(y[Ind[11,]==1], pik[Ind[11,]==1])

```

```

HT12 <- HT(y[Ind[12,]==1], pik[Ind[12,]==1])
HT13 <- HT(y[Ind[13,]==1], pik[Ind[13,]==1])
HT14 <- HT(y[Ind[14,]==1], pik[Ind[14,]==1])
HT15 <- HT(y[Ind[15,]==1], pik[Ind[15,]==1])
# The HT estimates arranged in a vector
Est <- c(HT1, HT2, HT3, HT4, HT5, HT6, HT7, HT8, HT9, HT10, HT11, HT12, HT13,
        HT14, HT15)
Est
# The HT is actually design-unbiased
data.frame(Ind, Est, p)
sum(Est*p)
sum(y)

```

Ik	<i>Sample Membership Indicator Matrix</i>
----	---

Description

Constructs the indicator matrix of the sampling support for a fixed-size without-replacement design. Each row corresponds to one possible sample and each column to one population unit.

Usage

```
Ik(N, n)
```

Arguments

N	Population size. Recommended $N \leq 15$.
n	Sample size.

Details

The full enumeration of all $\text{choose}(N, n)$ possible samples is computationally feasible only for small populations. For $N > 15$ this function will be very slow. It is intended primarily for theoretical illustrations and teaching purposes.

Value

A binary matrix of dimension $\text{choose}(N, n) \times N$, where entry $(s, k) = 1$ if unit k belongs to sample s , and 0 otherwise.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[Pik](#), [Pik1](#), [Support](#)

Examples

```
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
N <- length(U)
n <- 2
# The sample membership matrix
Ik(N, n)
# The first unit, Yves, belongs to the first four possible samples
```

IkRS

Sample Membership Indicator Matrix for All Possible Sample Sizes

Description

Constructs the indicator matrix of the complete sampling support, stacking the indicator matrices for all sample sizes from 1 to N. This covers every possible non-empty subset of the population.

Usage

```
IkRS(N)
```

Arguments

N Population size. Recommended $N \leq 10$.

Details

This function calls [Ik](#) for each possible sample size $n = 1, \dots, N$ and stacks the results. It is intended for small populations only ($N \leq 10$) due to the exponential growth of the support size.

Value

A binary matrix with 2^N rows (one per non-empty subset, including the empty set as the first row of zeros) and N columns. Entry $(s, k) = 1$ if unit k belongs to subset s .

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[Ik](#), [SupportRS](#)

Examples

```
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
N <- length(U)
# The sample membership matrix for all sample sizes
IkRS(N)
# The first sample is a null one and the last sample is a census
```

IkWR	<i>Sample Membership Indicator for with Replacements sampling designs</i>
------	---

Description

Constructs the indicator matrix of the with-replacement sampling support for a population of size N and m draws. Each row corresponds to one possible ordered outcome and each column to one population unit, with entry $(s, k) = 1$ if unit k was selected at least once in outcome s .

Usage

```
IkWR(N, m)
```

Arguments

N	Population size. Keep small due to combinatorial growth.
m	Number of draws (sample size with replacement).

Details

The with-replacement support is enumerated via [SupportWR](#). This function is intended for small populations and few draws only, as the support grows rapidly with N and m .

Value

A binary matrix of dimension $\text{choose}(N+m-1, m) \times N$, where entry $(s, k) = 1$ if unit k appears in the s -th outcome of the with-replacement support, and 0 otherwise.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[Ik](#), [SupportWR](#), [nk](#)

Examples

```
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
N <- length(U)
m <- 2
# The sample membership matrix for with-replacement sampling
IkWR(N, m)
```

IPFP

Iterative Proportional Fitting Procedure (Raking)

Description

Adjusts a contingency table so that its row and column marginals match known population totals, using the Iterative Proportional Fitting Procedure (IPFP), also known as raking or RAS algorithm.

Usage

```
IPFP(Table, Col.knw, Row.knw, tol = 1e-04)
```

Arguments

Table	A matrix or data frame of initial cell counts or weights to be adjusted.
Col.knw	Numeric vector of known column marginal totals.
Row.knw	Numeric vector of known row marginal totals.
tol	Convergence tolerance. The algorithm stops when the total absolute deviation between known and estimated marginals is below tol. Default is 0.0001.

Details

The algorithm alternates between row and column adjustments until convergence. At each step, cells in each row (or column) are multiplied by the ratio of the known marginal to the current estimated marginal. Convergence is assessed by the sum of absolute differences between known and estimated marginals.

Value

A matrix with `nrow(Table) + 1` rows and `ncol(Table) + 1` columns containing the adjusted cell counts, with an added row of estimated column totals and an added column of estimated row totals.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Deming, W.E. and Stephan, F.F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *Annals of Mathematical Statistics*, 11(4), 427-444.

Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[Domains](#), [Wk](#)

Examples

```
#####
## Example 1
#####
Table <- matrix(c(80, 90, 10, 170, 80, 80, 150, 210, 130), 3, 3)
rownames(Table) <- c("a1", "a2", "a3")
colnames(Table) <- c("b1", "b2", "b3")
Col.knw <- c(150, 300, 550)
Row.knw <- c(430, 360, 210)
IPFP(Table, Col.knw, Row.knw, tol = 0.0001)
#####
## Example 2
#####
data(Lucy)
attach(Lucy)
N <- dim(Lucy)[1]
n <- 400
sam <- sample(N, n)
data <- Lucy[sam, ]
attach(data)
Doma1 <- Domains(Level)
Doma2 <- Domains(SPAM)
SPAM.no <- Doma2[, 1] * Doma1
SPAM.yes <- Doma2[, 2] * Doma1
est1 <- E.SI(N, n, SPAM.no)[, 2:4]
est2 <- E.SI(N, n, SPAM.yes)[, 2:4]
Table <- cbind(est1[, ], est2[, ])
Col.knw <- colSums(Domains(Lucy$SPAM))
Row.knw <- colSums(Domains(Lucy$Level))
IPFP(Table, Col.knw, Row.knw, tol = 0.0001)
```

Description

Computes the optimal sample size allocation across strata using the Kish (1992) compromise allocation method, which interpolates between uniform and proportional allocation through a design effect parameter I.

Usage

```
kish_allocation(n, N_h, I = 0.5)
```

Arguments

- | | |
|-----|---|
| n | Integer. Total desired sample size. |
| N_h | Named numeric vector. Population sizes for each stratum $h = 1, \dots, H$. |
| I | Non-negative numeric. Intraclass correlation coefficient (ICC) or design effect parameter controlling the allocation: <ul style="list-style-type: none"> • $I = 0 \rightarrow$ Uniform allocation (equal sample per stratum). • $I = \text{Inf} \rightarrow$ Proportional allocation (proportional to N_h). • $0 < I < \text{Inf} \rightarrow$ Compromise between uniform and proportional. • Recommended value: $I = 0.5$ (Kish, 1992). |

Details

The Kish compromise allocation assigns sample sizes as:

$$n_h = n \cdot \frac{\sqrt{I W_h^2 + H^{-2}}}{\sum_{h=1}^H \sqrt{I W_h^2 + H^{-2}}}$$

where $W_h = N_h/N$ is the stratum weight and H is the number of strata. This formulation nests two classical allocations as limiting cases: when $I = 0$ the numerator reduces to $1/H$ (uniform), and as $I \rightarrow \infty$ it is dominated by W_h (proportional).

Value

A named integer vector of length H with the allocated sample sizes per stratum. The values sum to approximately n (rounding may cause a difference of ± 1).

Author(s)

Yury Vanessa Ochoa Montes <yury.ochoa@urosario.edu.co>

References

Kish, L. (1992). Weighting for unequal P_i . *Journal of Official Statistics*, 8(2), 183–200.

See Also

[E.STSI](#) for estimation under stratified sampling, [S.STSI](#) for stratified simple random sampling.

Examples

```
N_h <- c(
  Corozal      = 41847,
  Orange_Walk = 48175,
  Belize       = 57658,
  Cayo         = 78473,
  Stann_Creek = 31347,
  Toledo       = 31711
)

# Uniform allocation (I = 0)
kish_allocation(n = 3096, N_h = N_h, I = 0)

# Proportional allocation (I -> Inf)
kish_allocation(n = 3096, N_h = N_h, I = 1e6)

# Kish recommended compromise (I = 0.5)
kish_allocation(n = 3096, N_h = N_h, I = 0.5)
```

Lucy

Full Business Population Database

Description

A data frame containing socioeconomic and financial variables from a population of 2,396 business units, used throughout the package to illustrate survey sampling designs and estimators.

Usage

```
data(Lucy)
```

Format

A data frame with 2,396 rows and 8 variables:

ID The identifier of the company. It corresponds to an alphanumeric sequence (two letters and ten digits).

Ubication The address of the principal office of the company.

Level The size level of the company discriminated according to the income declared. There are small, medium and big companies.

Zone The country is divided by counties. A company belongs to a particular zone according to its cartographic location.

Income The total amount of a company's earnings in the previous fiscal year.

Employees The total number of persons working for the company in the previous fiscal year.

Taxes The total amount of a company's income tax.

SPAM Indicates if the company uses the Internet and webmail options to make self-propaganda.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro@gmail.com>

References

Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[BigLucy](#), [BigCity](#)

Examples

```
data(Lucy)
attach(Lucy)
# The variables of interest are: Income, Employees and Taxes
estima <- data.frame(Income, Employees, Taxes)
# The population totals
colSums(estima)
# Some parameters of interest
table(SPAM, Level)
xtabs(Income ~ Level + SPAM)
# Correlations among characteristics of interest
cor(estima)
# Some useful histograms
hist(Income)
hist(Taxes)
hist(Employees)
# Some useful plots
boxplot(Income ~ Level)
barplot(table(Level))
pie(table(SPAM))
```

nk

Sample Selection Indicator for With Replacement Sampling Designs

Description

Constructs the frequency matrix of the with-replacement sampling support for a population of size N and m draws. Each row corresponds to one possible outcome and each column to one population unit, with entry (s, k) equal to the number of times unit k was selected in outcome s .

Usage

`nk(N, m)`

Arguments

N	Population size. Keep small due to combinatorial growth.
m	Number of draws (sample size with replacement).

Details

Unlike [IkWR](#), which records only whether a unit was selected, this function records how many times each unit was selected. This is needed for with-replacement estimators based on selection frequencies.

Value

An integer matrix of dimension $\text{choose}(N+m-1, m) \times N$, where entry (s, k) is the frequency of unit k in outcome s .

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[IkWR](#), [SupportWR](#), [p.WR](#)

Examples

```
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
N <- length(U)
m <- 2
# Frequency matrix for with-replacement sampling
nk(N, m)
```

OrderWR

Ordered With-Replacement Sampling Support

Description

Enumerates all ordered sequences of m draws from a population of size N with replacement. Unlike [SupportWR](#), this function considers order, so sequences that differ only in draw order are treated as distinct outcomes.

Usage

```
OrderWR(N, m, ID = FALSE)
```

Arguments

N	Population size.
m	Number of draws.
ID	Optional vector of population labels of length N. If provided, labels are substituted for integer indices in the output. If FALSE (default), integer indices are returned.

Details

The total number of ordered with-replacement sequences of size m from N units is N^m . This grows rapidly and the function should only be used for small N and m .

Value

A matrix with N^m rows and m columns, where each row is one ordered sequence of draws. If ID is provided, population labels are substituted for indices.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[SupportWR](#), [IkWR](#)

Examples

```
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
N <- length(U)
# Five possible ordered samples of size m=1
OrderWR(N, 1)
OrderWR(N, 1, ID = U)
# 25 possible ordered samples of size m=2
OrderWR(N, 2)
OrderWR(N, 2, ID = U)
# Note: ordered samples differ from unordered (SupportWR)
OrderWR(N, 2)
SupportWR(N, 2)
```

p.WR

*Sample Probabilities under With-Replacement Sampling***Description**

Computes the probability of each possible outcome in the with-replacement sampling support, given unit selection probabilities p_k .

Usage

p.WR(N, m, pk)

Arguments

N	Population size.
m	Number of draws (sample size with replacement).
pk	Vector of length N with selection probabilities for each unit. Must sum to 1.

Details

For each distinct unordered outcome (multiset) in the support enumerated by [nk](#), the probability is computed as a multinomial probability:

$$p(s) = \frac{m!}{\prod_k n_k!} \prod_k p_k^{n_k}$$

where n_k is the number of times unit k appears in outcome s and p_k is the selection probability of unit k .

Value

A numeric vector of length $\text{choose}(N+m-1, m)$ with the probability of each distinct unordered outcome in the with-replacement support.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[nk](#), [SupportWR](#), [S.PPS](#)

Examples

```
#####
## Example 1
#####
# With replacement simple random sampling
# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# Vector pk is the selection probability of the units in the finite population
pk <- c(0.2, 0.2, 0.2, 0.2, 0.2)
sum(pk)
N <- length(pk)
m <- 3
# The sampling design
p <- p.WR(N, m, pk)
p
sum(p)

#####
## Example 2
#####
# With replacement PPS random sampling
# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# Vector x is the auxiliary information and y is the variables of interest
x<-c(32, 34, 46, 89, 35)
y<-c(52, 60, 75, 100, 50)
# Vector pk is the selection probability of the units in the finite population
pk <- x/sum(x)
sum(pk)
N <- length(pk)
m <- 3
# The sampling design
p <- p.WR(N, m, pk)
p
sum(p)
```

Pik

First-Order Inclusion Probabilities from a Sampling Design

Description

Computes the first-order inclusion probabilities for each unit in a finite population, given the probability of each possible sample and the indicator matrix of the sampling support.

Usage

Pik(p, Ind)

Arguments

p	Vector of probabilities for each possible sample in the support. Must sum to 1.
Ind	Indicator matrix of the sampling support, as returned by Ik . Rows are samples, columns are population units.

Details

The inclusion probability of unit k is computed as the sum of the probabilities of all samples that contain unit k :

$$\pi_k = \sum_{s \ni k} p(s)$$

The indicator matrix Ind (output of [Ik](#)) has one row per possible sample and one column per population unit, with entry 1 if unit k is in sample s and 0 otherwise.

Value

A row vector (1 x N matrix) of first-order inclusion probabilities $\pi_k = P(k \in s)$ for each unit k in the population.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[Ik](#), [Pikl](#), [PikPPS](#)

Examples

```
# Population of size N = 5, sample size n = 2
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
N <- length(U)
n <- 2
# Sample probabilities (one per possible sample)
p <- c(0.13, 0.2, 0.15, 0.1, 0.15, 0.04, 0.02, 0.06, 0.07, 0.08)
Ind <- Ik(N, n)
pik <- Pik(p, Ind)
pik
# Check: inclusion probabilities sum to n
sum(pik)
```

PikHol

*Optimal Inclusion Probabilities for Multiple Surveys (Holmberg)***Description**

Computes optimal first-order inclusion probabilities for a population that is surveyed on multiple occasions, minimising a measure of total variance across surveys. This implements the approach of Holmberg (2002) for coordinated sampling over time.

Usage

```
PikHol(n, sigma, e, Pi = NULL)
```

Arguments

n	Integer vector of length p with the desired sample size for each of the p surveys.
sigma	Matrix of dimension N x p where column <i>k</i> contains the auxiliary size variable for survey <i>k</i> .
e	Scalar. Relative tolerance parameter controlling the precision target across surveys.
Pi	Optional matrix of dimension N x p with initial inclusion probabilities for each survey. If omitted, PikPPS is used.

Details

For each survey *k*, the initial inclusion probabilities are computed via [PikPPS](#). An optimal composite size measure is then derived by combining the per-survey auxiliary variables through a weighted sum, and the final inclusion probabilities are computed proportional to the square root of this composite. The resulting sample size *n.st* is chosen to minimise total variance subject to a relative precision target *e*.

Value

A numeric vector of length N with the optimal inclusion probability for each unit in the population.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Holmberg, A. (2002). A multiparameter perspective on the choice of sampling design in surveys. *Statistics in Transition*, 5(6), 969-994.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[PikPPS](#), [PikSTPPS](#)

Examples

```
#####
## Example 1
#####
data(Lucy)
attach(Lucy)
N <- dim(Lucy)[1]
n <- c(350, 400)
sigy1 <- sqrt(Income^(1))
sigy2 <- sqrt(Income^(2))
sigma <- cbind(sigy1, sigy2)
Piks <- PikHol(n, sigma, 0.03)
n.opt <- round(sum(Piks))
res <- S.piPS(n.opt, Piks)
sam <- res[, 1]
Pik.s <- res[, 2]
estima <- data.frame(Lucy$Income[sam], Lucy$Employees[sam])
E.piPS(estima, Pik.s)
#####
## Example 2 - with custom inclusion probabilities
#####
data(Lucy)
attach(Lucy)
N <- dim(Lucy)[1]
n <- c(350, 400)
sigy1 <- sqrt(Income^(1))
sigy2 <- sqrt(Income^(2))
sigma <- cbind(sigy1, sigy2)
pikas <- cbind(rep(400/N, N), rep(400/N, N))
Piks <- PikHol(n, sigma, 0.03, pikas)
round(sum(Piks))
```

Pikl

Second-Order Inclusion Probabilities

Description

Computes the matrix of second-order inclusion probabilities $\pi_{kl} = P(k \in s \text{ and } l \in s)$ for all pairs of units in a finite population of size N under a fixed-size sampling design.

Usage

Pikl(N, n, p)

Arguments

N	Population size. Keep small (recommended $N \leq 15$) due to the combinatorial enumeration of all possible samples.
n	Sample size.
p	Vector of probabilities for each possible sample in the support. Must sum to 1.

Details

The second-order inclusion probabilities are needed to compute the exact Horvitz-Thompson variance estimator and the Sen-Yates-Grundy variance estimator. This function enumerates the full sampling support via [Ik](#) and is therefore only feasible for small populations ($N \leq 15$).

Value

An $N \times N$ matrix where entry (k, l) is the probability that both units k and l are included in the same sample. Diagonal entries (k, k) equal the first-order inclusion probability π_k .

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[Pik](#), [Deltakl](#), [VarHT](#)

Examples

```
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
N <- length(U)
n <- 2
p <- c(0.13, 0.2, 0.15, 0.1, 0.15, 0.04, 0.02, 0.06, 0.07, 0.08)
sum(p)
# Second-order inclusion probabilities
Pikl(N, n, p)
```

PikPPS

*Inclusion Probabilities Proportional to Size***Description**

Computes first-order inclusion probabilities proportional to an auxiliary size variable x for a without-replacement sample of size n . A sequential truncation algorithm ensures all probabilities are at most 1.

Usage

```
PikPPS(n, x)
```

Arguments

n	Desired sample size.
x	Vector of length N with positive auxiliary size values for each unit in the population.

Details

The initial probabilities $\pi_k = nx_k / \sum x$ may exceed 1 for large units. The algorithm iteratively sets those probabilities to 1 and redistributes the remaining sample size among the other units until all probabilities are valid. The result satisfies $\sum \pi_k = n$.

Value

A numeric vector of length N with the first-order inclusion probability for each unit in the population. Values are in $(0, 1]$.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[S.piPS](#), [PikSTPPS](#), [PikHol](#)

Examples

```
#####
## Example 1
#####
x <- c(30,41,50,170,43,200)
n <- 3
# Two elements yields values bigger than one
n*x/sum(x)
# With this functions, all of the values are between zero and one
PikPPS(n,x)
# The sum is equal to the sample size
sum(PikPPS(n,x))

#####
## Example 2
#####
# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# The auxiliary information
x <- c(52, 60, 75, 100, 50)
# Gives the inclusion probabilities for the population according to a
# proportional to size design without replacement of size n=4
pik <- PikPPS(4,x)
pik
# The selected sample is
sum(pik)

#####
## Example 3
#####
# Uses the Lucy data to compute the vector of inclusion probabilities
# according to a piPS without replacement design
data(Lucy)
attach(Lucy)
# The sample size
n=400
# The selection probability of each unit is proportional to the variable Income
pik <- PikPPS(n,Income)
# The inclusion probabilities of the units in the sample
pik
# The sum of the values in pik is equal to the sample size
sum(pik)
# According to the design some elements must be selected
# They are called forced inclusion units
which(pik==1)
```

Description

For a given sample size, in each stratum, this function returns a vector of first order inclusion probabilities for an stratified sampling design proportional to an auxiliary variable.

Usage

```
PikSTPPS(S, x, nh)
```

Arguments

S	Vector identifying the membership to the strata of each unit in the population.
x	Vector of auxiliary information for each unit in the population.
nh	The vector defining the sample size in each stratum.

Details

is not always less than unity. A sequential algorithm must be used in order to ensure that for every unit in the population the inclusion probability gives a proper value; i.e. less or equal to unity.

Value

A vector of inclusion probabilities in a stratified finite population.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas Sarndal, C-E. and Swensson, B. and Wretman, J. (2003), *Model Assisted Survey Sampling*. Springer.

See Also

[PikHol](#), [PikPPS](#), [S.STpiPS](#)

Examples

```
#####
## Example 1
#####
# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# The auxiliary information
x <- c(52, 60, 75, 100, 50)
# Vector Strata contains an indicator variable of stratum membership
Strata <- c("A", "A", "A", "B", "B")
# The sample size in each stratum
nh <- c(2,2)
```

```

# The vector of inclusion probabilities for a stratified piPS sample
# without replacement of size two within each stratum
Pik <- PikSTPPS(Strata, x, nh)
Pik

# Some checks
sum(Pik)
sum(nh)

#####
## Example 2
#####
# Uses the Lucy data to compute the vector of inclusion probabilities
# for a stratified random sample according to a piPS design in each stratum

data(Lucy)
attach(Lucy)
# Level is the stratifying variable
summary(Level)

# Defines the size of each stratum
N1<-summary(Level)[[1]]
N2<-summary(Level)[[2]]
N3<-summary(Level)[[3]]
N1;N2;N3

# Defines the sample size at each stratum
n1<-70
n2<-100
n3<-200
nh<-c(n1,n2,n3)
nh

# Computes the inclusion probabilities for the stratified population
S <- Level
x <- Employees
Pik <- PikSTPPS(S, x, nh)

# Some checks
sum(Pik)
sum(nh)

```

Description

Draws a Bernoulli sample from a finite population of size N . Each unit is independently selected with the same inclusion probability prob.

Usage

S.BE(N, prob)

Arguments

N	Population size.
prob	Scalar. Inclusion probability, must satisfy $0 < \text{prob} \leq 1$.

Details

The sample size under Bernoulli sampling is random, following a Binomial(N, prob) distribution. To extract the selected indices, use `sam[sam != 0]`.

Value

A vector of length N where selected units contain their population index and non-selected units contain 0.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseño de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[E.BE](#), [S.PO](#), [S.SI](#)

Examples

```
#####
## Example 1
#####
# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# Draws a Bernoulli sample without replacement of expected size n=3
# The inclusion probability is 0.6 for each unit in the population
sam <- S.BE(5,0.6)
sam
# The selected sample is
U[sam]

#####
## Example 2
#####
# Uses the Lucy data to draw a Bernoulli sample
```

```

data(Lucy)
attach(Lucy)
N <- dim(Lucy)[1]
# The population size is 2396. If the expected sample size is 400
# then, the inclusion probability must be 400/2396=0.1669
sam <- S.BE(N,0.01669)
# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
data
dim(data)

```

S.piPS	<i>Probability Proportional to Size Without-Replacement Sampling (piPS)</i>
--------	---

Description

Draws a without-replacement sample of size n using a sequential algorithm that produces inclusion probabilities proportional to an auxiliary size variable x .

Usage

```
S.piPS(n, x, e = runif(length(x)))
```

Arguments

n	Sample size.
x	Vector of length N with positive auxiliary size values.
e	Optional vector of N uniform random variates in $(0, 1)$. If omitted, <code>runif(N)</code> is used.

Value

A matrix with n rows and two columns:

- Column 1: population indices of the selected units.
- Column 2: first-order inclusion probabilities of the selected units.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[E.piPS](#), [PikPPS](#), [S.STPPS](#)

Examples

```
#####
## Example 1
#####
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
x <- c(52, 60, 75, 100, 50)
# Draws a piPS sample without replacement of size n=3
res <- S.piPS(3, x)
res
sam <- res[, 1]
U[sam]
#####
## Example 2
#####
# Uses the Lucy data
data(Lucy)
attach(Lucy)
res <- S.piPS(400, Income)
sam <- res[, 1]
Pik.s <- res[, 2]
data <- Lucy[sam, ]
dim(data)
```

S.PO

Poisson Sampling

Description

Draws a Poisson sample from a finite population of size N . Each unit k is independently selected with its own inclusion probability π_k .

Usage

S.PO(N, Pik)

Arguments

N	Population size.
Pik	Vector of length N containing the first-order inclusion probability for each unit in the population. Values must be in $(0, 1]$.

Details

Poisson sampling is a generalisation of Bernoulli sampling that allows unequal inclusion probabilities. The sample size is random. To extract the selected indices, use `sam[sam != 0]`.

Value

A vector of length N where selected units contain their population index and non-selected units contain 0.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[E.PO](#), [PikPPS](#), [S.piPS](#)

Examples

```
#####
## Example 1
#####
# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# Draws a Bernoulli sample without replacement of expected size n=3
# "Erik" is drawn in every possible sample because its inclusion probability is one
Pik <- c(0.5, 0.2, 1, 0.9, 0.5)
sam <- S.PO(5,Pik)
sam
# The selected sample is
U[sam]

#####
## Example 2
#####
# Uses the Lucy data to draw a Poisson sample
data(Lucy)
attach(Lucy)
N <- dim(Lucy)[1]
n <- 400
Pik<-n*Income/sum(Income)
# None element of Pik bigger than one
which(Pik>1)
# The selected sample
sam <- S.PO(N,Pik)
# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
data
dim(data)
```

S.PPS

*Probability Proportional to Size With-Replacement Sampling***Description**

Draws a with-replacement sample of size m from a finite population using probabilities proportional to an auxiliary size variable x .

Usage

S.PPS(m , x)

Arguments

m	Number of draws (sample size with replacement).
x	Vector of length N containing positive auxiliary size values for each unit in the population.

Details

At each draw, unit k is selected with probability $p_k = x_k / \sum x$. Since sampling is with replacement, the same unit may appear more than once. Use [E.PPS](#) or [HH](#) to estimate population totals from this sample.

Value

A matrix with m rows and two columns:

- Column 1 (sam): population indices of the selected units.
- Column 2 (pk): selection probability of each draw.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[E.PPS](#), [HH](#), [S.piPS](#)

Examples

```
#####
## Example 1
#####
# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# The auxiliary information
x <- c(52, 60, 75, 100, 50)
# Draws a PPS sample with replacement of size m=3
res <- S.PPS(3,x)
sam <- res[,1]
# The selected sample is
U[sam]

#####
## Example 2
#####
# Uses the Lucy data to draw a random sample according to a
# PPS with replacement design
data(Lucy)
attach(Lucy)
# The selection probability of each unit is proportional to the variable Income
m <- 400
res<-S.PPS(400,Income)
# The selected sample
sam <- res[,1]
# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
data
dim(data)
```

S.SI

Simple Random Sampling Without Replacement

Description

Draws a simple random sample of size n without replacement from a finite population of size N using the sequential algorithm of Fan, Muller and Rezucha (1962).

Usage

```
S.SI(N, n, e = runif(N))
```

Arguments

N	Population size.
n	Sample size. Must satisfy $n \leq N$.
e	Optional vector of N uniform random variates in $(0, 1)$. If omitted, <code>runif(N)</code> is used. Useful for reproducibility or coordinated sampling.

Details

The sequential algorithm selects units one at a time by comparing a uniform random variate with the conditional inclusion probability at each step, ensuring exactly n units are selected. To extract the selected indices, filter out the zeros: `sam[sam != 0]`.

Value

A vector of length N where selected units contain their population index and non-selected units contain 0.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Fan, C.T., Muller, M.E. and Rezucha, I. (1962). Development of sampling plans by using sequential (item by item) selection techniques and digital computers. *Journal of the American Statistical Association*, 57(298), 387-402.

Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[E.SI](#), [S.STSI](#), [S.SY](#)

Examples

```
#####
## Example 1
#####
# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# Fixes the random numbers in order to select a sample
e <- c(0.4938, 0.7044, 0.4585, 0.6747, 0.0640)
# Draws a simple random sample without replacement of size n=3
sam <- S.SI(5, 3, e)
sam
# The selected sample is
U[sam]
#####
## Example 2
#####
# Uses the Lucy data to draw a random sample according to a SI design
data(Lucy)
attach(Lucy)
N <- dim(Lucy)[1]
n <- 400
sam <- S.SI(N, n)
# The information about the units in the sample
data <- Lucy[sam, ]
```

```
dim(data)
```

S.STpiPS	<i>Stratified Probability Proportional to Size Without-Replacement Sampling</i>
----------	---

Description

Draws a stratified sample where within each stratum units are selected using a probability proportional to size without-replacement (piPS) design.

Usage

```
S.STpiPS(S, x, nh)
```

Arguments

S	Vector of length N identifying the stratum membership of each unit in the population.
x	Vector of length N containing positive auxiliary size values for each unit in the population.
nh	Integer vector of length H specifying the sample size within each stratum.

Details

Within each stratum h , the function calls [S.piPS](#) to draw $nh[h]$ units with probabilities proportional to x . The global population indices are preserved in the output.

Value

A matrix with $\text{sum}(nh)$ rows and two columns, sorted by population index:

- Column 1: population indices of the selected units.
- Column 2: first-order inclusion probabilities of the selected units.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[S.piPS](#), [S.STSI](#), [E.STpiPS](#), [PikSTPPS](#)

Examples

```
#####
## Example 1
#####
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
x <- c(52, 60, 75, 100, 50)
Strata <- c("A", "A", "A", "B", "B")
nh <- c(2, 2)
res <- S.STpiPS(Strata, x, nh)
sam <- res[, 1]
U[sam]
pik <- res[, 2]
pik
#####
## Example 2
#####
data(Lucy)
attach(Lucy)
N1 <- summary(Level)[[1]]
N2 <- summary(Level)[[2]]
N3 <- summary(Level)[[3]]
nh <- c(70, 100, 200)
res <- S.STpiPS(Level, Employees, nh)
sam <- res[, 1]
data <- Lucy[sam, ]
dim(data)
pik <- res[, 2]
```

S.STPPS

Stratified Probability Proportional to Size With-Replacement Sampling

Description

Draws a stratified with-replacement sample where within each stratum units are selected using probability proportional to size (PPS-WR).

Usage

```
S.STPPS(S, x, mh)
```

Arguments

S	Vector of length N identifying the stratum membership of each unit in the population.
x	Vector of length N containing positive auxiliary size values for each unit in the population.
mh	Integer vector of length H specifying the number of draws within each stratum.

Details

Within each stratum h , $\text{mh}[h]$ draws are made with probabilities $p_k = x_k / \sum_{k \in h} x_k$. The same unit may appear more than once within a stratum. Use [E.STPPS](#) to estimate population totals from this sample.

Value

A data frame with $\text{sum}(\text{mh})$ rows and two columns:

- `sam`: population indices of the selected units.
- `pk`: within-stratum selection probabilities of each draw.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[S.PPS](#), [S.STpiPS](#), [E.STPPS](#)

Examples

```
#####
## Example 1
#####
# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# The auxiliary information
x <- c(52, 60, 75, 100, 50)
# Vector Strata contains an indicator variable of stratum membership
Strata <- c("A", "A", "A", "B", "B")
# Then sample size in each stratum
mh <- c(2,2)
# Draws a stratified PPS sample with replacement of size n=4
res <- S.STPPS(Strata, x, mh)
# The selected sample
sam <- res[,1]
U[sam]
# The selection probability of each unit selected to be in the sample
pk <- res[,2]
pk

#####
## Example 2
#####
```

```

# Uses the Lucy data to draw a stratified random sample
# according to a PPS design in each stratum

data(Lucy)
attach(Lucy)
# Level is the stratifying variable
summary(Level)
# Defines the sample size at each stratum
m1<-70
m2<-100
m3<-200
mh<-c(m1,m2,m3)
# Draws a stratified sample
res<-S.STPPS(Level, Income, mh)
# The selected sample
sam<-res[,1]
# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
data
dim(data)
# The selection probability of each unit selected in the sample
pk <- res[,2]
pk

```

S.STSI

Stratified Simple Random Sampling Without Replacement

Description

Draws a stratified simple random sample without replacement from a finite population. Within each stratum, units are selected by simple random sampling without replacement.

Usage

```
S.STSI(S, Nh, nh)
```

Arguments

S	Vector of length N identifying the stratum membership of each unit in the population.
Nh	Integer vector of length H with the population size of each stratum.
nh	Integer vector of length H with the sample size of each stratum. Must satisfy $nh[h] \leq Nh[h]$ for all h.

Details

The function selects $nh[h]$ units from stratum h using `base::sample`, and returns all selected indices sorted in ascending order. Use [E.STSI](#) to estimate population totals from this sample.

Value

A sorted vector of population indices of the selected units, of length `sum(nh)`.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*.
 Editorial Universidad Santo Tomas.

See Also

[E.STSI](#), [S.SI](#), [S.STpiPS](#)

Examples

```
#####
## Example 1
#####
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
Strata <- c("A", "A", "A", "B", "B")
Nh <- c(3, 2)
nh <- c(2, 1)
sam <- S.STSI(Strata, Nh, nh)
sam
U[sam]
#####
## Example 2
#####
data(Lucy)
attach(Lucy)
N1 <- summary(Level)[[1]]
N2 <- summary(Level)[[2]]
N3 <- summary(Level)[[3]]
Nh <- c(N1, N2, N3)
nh <- c(70, 100, 200)
sam <- S.STSI(Level, Nh, nh)
data <- Lucy[sam, ]
dim(data)
```

Description

Draws a systematic sample from a finite population of size N using a fixed sampling interval a . A random start r is chosen uniformly from 1 to a , and every a -th unit thereafter is selected.

Usage

S.SY(N, a)

Arguments

N Population size.
a Sampling interval (skip). The expected sample size is approximately N/a .

Details

The random start r is drawn from `sample(a, 1)`, and then units $r, r + a, r + 2a, \dots$ are selected. If N is not a multiple of a , the sample size varies by one unit depending on the random start. Use [E.SY](#) to estimate population totals.

Value

A vector containing the population indices of the selected units.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[E.SY](#), [S.SI](#)

Examples

```
#####
## Example 1
#####
# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# The population of size N=5 is divided in a=2 groups
# Draws a Systematic sample.
sam <- S.SY(5,2)
sam
# The selected sample is
U[sam]
# There are only two possible samples

#####
## Example 2
#####
# Uses the Lucy data to draw a Systematic sample
```

```

data(Lucy)
attach(Lucy)

N <- dim(Lucy)[1]
# The population is divided in 6 groups
# The selected sample
sam <- S.SY(N,6)
# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
data
dim(data)

```

S.WR

*Simple Random Sampling With Replacement***Description**

Draws a simple random sample of size m with replacement from a finite population of size N . Returns the frequency of selection for each unit drawn at least once.

Usage

```
S.WR(N, m)
```

Arguments

N	Population size.
m	Number of draws (sample size with replacement).

Details

The number of times each unit is selected follows a multinomial distribution with equal probabilities $1/N$. The function uses a sequential binomial draw approach. Use [E.WR](#) to estimate population totals.

Value

A vector of population indices of length m , where each element is the index of a selected unit. Units may appear more than once.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[E.WR](#), [S.SI](#), [S.PPS](#)

Examples

```
#####
## Example 1
#####
# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# Draws a simple random sample with replacement of size m=3
sam <- S.WR(5,3)
sam
# The selected sample
U[sam]

#####
## Example 2
#####
# Uses the Lucy data to draw a random sample of units according to a
# simple random sampling with replacement design
data(Lucy)
attach(Lucy)

N <- dim(Lucy)[1]
m <- 400
sam<-S.WR(N,m)
# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
data
dim(data)
```

Support

Sampling Support for Fixed-Size Without-Replacement Designs

Description

Enumerates all possible samples of size n from a population of size N , returning the complete sampling support as a matrix.

Usage

```
Support(N, n, ID = FALSE)
```

Arguments

N	Population size. Recommended $N \leq 15$.
n	Sample size.
ID	Optional vector of population labels of length N . If provided, labels replace integer indices in the output. If FALSE (default), integer indices are returned.

Details

This function uses a combinatorial algorithm to enumerate all $\text{choose}(N, n)$ subsets of size n from $\{1, \dots, N\}$. It is intended for small populations only. For $N > 15$ it becomes very slow.

Value

A matrix with $\text{choose}(N, n)$ rows and n columns. Each row contains the indices (or labels if ID is provided) of the units in one possible sample. Samples are listed in lexicographic order.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[Ik](#), [SupportWR](#), [SupportRS](#)

Examples

```
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
N <- length(U)
n <- 2
# Ten possible samples of size n=2
Support(N, n)
# Labeled support
Support(N, n, ID = U)
# Support showing values of y
y <- c(32, 34, 46, 89, 35)
Support(N, n, ID = y)
```

SupportRS

Complete Sampling Support for All Sample Sizes

Description

Enumerates all possible non-empty subsets of a population of size N , covering all sample sizes from 1 to N . The result includes the empty set as the first row.

Usage

```
SupportRS(N, ID = FALSE)
```

Arguments

N	Population size. Recommended $N \leq 10$.
ID	Optional vector of population labels of length N. If provided, labels replace integer indices in the output.

Details

This function stacks the outputs of [Support](#) for all sample sizes $n = 1, \dots, N$. It is only feasible for small populations ($N \leq 10$) due to exponential growth.

Value

A matrix with 2^N rows and N columns. Each row is one subset, with NA used as padding for subsets smaller than N. The first row represents the empty set (all zeros).

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[Support](#), [IkRS](#)

Examples

```
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
N <- length(U)
# Complete support for all sample sizes
SupportRS(N)
# Labeled support
SupportRS(N, ID = U)
```

Description

Enumerates all distinct unordered outcomes (multisets) of size m drawn with replacement from a population of size N.

Usage

```
SupportWR(N, m, ID = FALSE)
```

Arguments

N	Population size.
m	Number of draws (sample size with replacement).
ID	Optional vector of population labels of length N. If FALSE (default), integer indices are returned.

Details

The number of distinct unordered with-replacement outcomes of size m from N units is $\binom{N+m-1}{m}$. This is much smaller than the N^m ordered outcomes. The algorithm uses a nested loop to generate all non-decreasing sequences of length m from $\{1, \dots, N\}$.

Value

A matrix with $\text{choose}(N+m-1, m)$ rows and m columns. Each row contains the (sorted) indices of one possible unordered outcome. If ID is provided, population labels replace indices.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[IkWR](#), [nk](#), [p.WR](#)

Examples

```
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
N <- length(U)
m <- 2
# With-replacement support
SupportWR(N, m)
SupportWR(N, m, ID = U)
y <- c(32, 34, 46, 89, 35)
SupportWR(N, m, ID = y)
```

T.SIC

Cluster Totals for Single-Stage Cluster Sampling

Description

Computes the total of each variable of interest within each cluster (Primary Sampling Unit) in a single-stage cluster sample.

Usage

T.SIC(y, Cluster)

Arguments

y	Vector, matrix or data frame containing the values of the variables of interest for every unit in the sample.
Cluster	Vector identifying the cluster (PSU) membership of each unit in the sample.

Details

This function aggregates the sample data by cluster, producing the cluster- level totals needed for estimation under single-stage cluster sampling. The output can be passed directly to [E.1SI](#) or [E.SI](#) treating each cluster total as an observation.

Value

A matrix with one row per cluster and one column per variable of interest (plus a first column Ni with the cluster size). Row names are the cluster labels.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[E.1SI](#), [E.2SI](#)

Examples

```
#####
## Example 1
#####
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
y1 <- c(32, 34, 46, 89, 35)
y2 <- c(1, 1, 1, 0, 0)
y3 <- cbind(y1, y2)
Cluster <- c("C1", "C2", "C1", "C2", "C1")
T.SIC(y1, Cluster)
T.SIC(y3, Cluster)
#####
## Example 2 - Cluster sampling with Lucy data
#####
data(Lucy)
attach(Lucy)
UI <- c("A", "B", "C", "D", "E")
NI <- length(UI)
nI <- 2
samI <- S.SI(NI, nI)
dataI <- UI[samI]
Lucy1 <- Lucy[which(Zone == dataI[1]), ]
Lucy2 <- Lucy[which(Zone == dataI[2]), ]
LucyI <- rbind(Lucy1, Lucy2)
attach(LucyI)
Cluster <- as.factor(as.integer(Zone))
estima <- data.frame(Income, Employees, Taxes)
Ty <- T.SIC(estima, Cluster)
E.SI(NI, nI, Ty)
```

VarHT

Exact Variance of the Horvitz-Thompson Estimator

Description

Computes the exact variance of the Horvitz-Thompson estimator of the population total for a given fixed-size without-replacement sampling design, using the full sampling support.

Usage

```
VarHT(y, N, n, p)
```

Arguments

y	Vector of length N with the population values of the variable of interest.
N	Population size. Recommended $N \leq 15$.
n	Sample size.
p	Vector of probabilities for each possible sample in the support. Must sum to 1.

Details

The exact Horvitz-Thompson variance is:

$$V(\hat{t}_{y,\pi}) = \sum_{k=1}^N \sum_{l=1}^N \Delta_{kl} \frac{y_k}{\pi_k} \frac{y_l}{\pi_l}$$

where $\Delta_{kl} = \pi_{kl} - \pi_k \pi_l$. This requires enumerating the full support and is only feasible for small populations ($N \leq 15$).

Value

A scalar: the exact variance of the Horvitz-Thompson estimator $V(\hat{t}_{y,\pi})$.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Horvitz, D.G. and Thompson, D.J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47, 663-685.
 Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.

See Also

[Deltak1](#), [VarSYGHT](#), [HT](#)

Examples

```
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
y1 <- c(32, 34, 46, 89, 35)
y2 <- c(1, 1, 1, 0, 0)
N <- length(U)
n <- 2
p <- c(0.13, 0.2, 0.15, 0.1, 0.15, 0.04, 0.02, 0.06, 0.07, 0.08)
# Theoretical variance of the HT estimator
VarHT(y1, N, n, p)
VarHT(y2, N, n, p)
```

VarSYGHT

Two different variance estimators for the Horvitz-Thompson estimator

Description

This function estimates the variance of the Horvitz-Thompson estimator. Two different variance estimators are computed: the original one, due to Horvitz-Thompson and the one due to Sen (1953) and Yates, Grundy (1953). The two approaches yield unbiased estimator under fixed-size sampling schemes.

Usage

```
VarSYGHT(y, N, n, p)
```

Arguments

y	Vector containing the information of the characteristic of interest for every unit in the population.
N	Population size.
n	Sample size.
p	A vector containing the selection probabilities of a fixed size without replacement sampling design. The sum of the values of this vector must be one.

Details

The function returns two variance estimator for every possible sample within a fixed-size sampling support. The first estimator is due to Horvitz-Thompson and is given by the following expression:

$$\widehat{Var}_1(\hat{t}_{y,\pi}) = \sum_{k \in U} \sum_{l \in U} \frac{\Delta_{kl}}{\pi_{kl}} \frac{y_k}{\pi_k} \frac{y_l}{\pi_l}$$

The second estimator is due to Sen (1953) and Yates-Grundy (1953). It is given by the following expression:

$$\widehat{Var}_2(\hat{t}_{y,\pi}) = -\frac{1}{2} \sum_{k \in U} \sum_{l \in U} \frac{\Delta_{kl}}{\pi_{kl}} \left(\frac{y_k}{\pi_k} - \frac{y_l}{\pi_l} \right)^2$$

Value

This function returns a data frame of every possible sample in within a sampling support, with its corresponding variance estimates.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

Examples

```
# Example 1
# Without replacement sampling
# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# Vector y1 and y2 are the values of the variables of interest
y1<-c(32, 34, 46, 89, 35)
y2<-c(1,1,1,0,0)
```

```

# The population size is N=5
N <- length(U)
# The sample size is n=2
n <- 2
# p is the probability of selection of every possible sample
p <- c(0.13, 0.2, 0.15, 0.1, 0.15, 0.04, 0.02, 0.06, 0.07, 0.08)

# Calculates the estimated variance for the HT estimator
VarSYGHT(y1, N, n, p)
VarSYGHT(y2, N, n, p)

# Unbiasedness holds in the estimator of the total
sum(y1)
sum(VarSYGHT(y1, N, n, p)$p * VarSYGHT(y1, N, n, p)$Est.HT)
sum(y2)
sum(VarSYGHT(y2, N, n, p)$p * VarSYGHT(y2, N, n, p)$Est.HT)

# Unbiasedness also holds in the two variances
VarHT(y1, N, n, p)
sum(VarSYGHT(y1, N, n, p)$p * VarSYGHT(y1, N, n, p)$Est.Var1)
sum(VarSYGHT(y1, N, n, p)$p * VarSYGHT(y1, N, n, p)$Est.Var2)

VarHT(y2, N, n, p)
sum(VarSYGHT(y2, N, n, p)$p * VarSYGHT(y2, N, n, p)$Est.Var1)
sum(VarSYGHT(y2, N, n, p)$p * VarSYGHT(y2, N, n, p)$Est.Var2)

# Example 2: negative variance estimates

x = c(2.5, 2.0, 1.1, 0.5)
N = 4
n = 2
p = c(0.31, 0.20, 0.14, 0.03, 0.01, 0.31)

VarSYGHT(x, N, n, p)

# Unbiasedness holds in the estimator of the total
sum(x)
sum(VarSYGHT(x, N, n, p)$p * VarSYGHT(x, N, n, p)$Est.HT)

# Unbiasedness also holds in the two variances
VarHT(x, N, n, p)
sum(VarSYGHT(x, N, n, p)$p * VarSYGHT(x, N, n, p)$Est.Var1)
sum(VarSYGHT(x, N, n, p)$p * VarSYGHT(x, N, n, p)$Est.Var2)

```

Description

Computes the generalised regression (GREG) weights for each unit in the sample. These weights incorporate both the sampling design weights and a calibration adjustment based on known popu-

lation totals of auxiliary variables.

Usage

`Wk(x, tx, Pik, ck, b0 = FALSE)`

Arguments

<code>x</code>	Vector or matrix of auxiliary variables observed in the sample.
<code>tx</code>	Vector of known population totals of the auxiliary variables.
<code>Pik</code>	Vector of first-order inclusion probabilities for each unit in the sample.
<code>ck</code>	Vector of variance-stabilising constants. Typically $c_k = 1$ (homoscedastic) or $c_k = x$ (heteroscedastic).
<code>b0</code>	Logical. If TRUE, an intercept column is prepended to <code>x</code> . Default is FALSE.

Details

The GREG weight for unit k is:

$$w_k = \frac{1}{\pi_k} + \mathbf{x}_k^T \left(\sum_s \frac{v_k \mathbf{x}_k \mathbf{x}_k^T}{\pi_k} \right)^{-1} (\mathbf{t}_x - \hat{\mathbf{t}}_{x,\pi})$$

where $v_k = 1/(\pi_k c_k)$ and c_k is a variance-stabilising constant. The GREG estimator is then $\hat{t}_{GREG} = \sum_s w_k y_k$.

Value

A numeric vector of length `n` with the GREG weight for each unit in the sample.

Author(s)

Hugo Andres Gutierrez Rojas <hagutierrezro at gmail.com>

References

Sarndal, C-E. and Swensson, B. and Wretman, J. (1992), *Model Assisted Survey Sampling*. Springer.
 Gutierrez, H. A. (2009), *Estrategias de muestreo: Diseno de encuestas y estimacion de parametros*. Editorial Universidad Santo Tomas.

See Also

[GREG.SI](#), [E.Beta](#)

Examples

```
#####
## Example 1
#####
# Without replacement sampling
# Vector U contains the label of a population of size N=5
U <- c("Yves", "Ken", "Erik", "Sharon", "Leslie")
# Vector x is the auxiliary information and y is the variables of interest
x<-c(32, 34, 46, 89, 35)
y<-c(52, 60, 75, 100, 50)
# pik is some vector of inclusion probabilities in the sample
# In this case the sample size is equal to the population size
pik<-rep(1,5)
w1<-Wk(x,tx=236,pik,ck=1,b0=FALSE)
sum(x*w1)
# Draws a sample size without replacement
sam <- sample(5,2)
pik <- c (0.8,0.2,0.2,0.5,0.3)
# The auxiliary information an variable of interest in the selected smaple
x.s<-x[sam]
y.s<-y[sam]
# The vector of inclusion probabilities in the selected smaple
pik.s<-pik[sam]
# Calibration weights under some specifics model
w2<-Wk(x.s,tx=236,pik.s,ck=1,b0=FALSE)
sum(x.s*w2)

w3<-Wk(x.s,tx=c(5,236),pik.s,ck=1,b0=TRUE)
sum(w3)
sum(x.s*w3)

w4<-Wk(x.s,tx=c(5,236),pik.s,ck=x.s,b0=TRUE)
sum(w4)
sum(x.s*w4)

w5<-Wk(x.s,tx=236,pik.s,ck=x.s,b0=FALSE)
sum(x.s*w5)

#####
## Example 2: Linear models involving continuous auxiliary information
#####

# Draws a simple random sample without replacement
data(Lucy)
attach(Lucy)

N <- dim(Lucy)[1]
n <- 400
Pik <- rep(n/N, n)
sam <- S.SI(N,n)
# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
```

```

attach(data)
names(data)

##### common ratio model #####

estima<-data.frame(Income)
x <- Employees
tx <- sum(Lucy$Employees)
w <- Wk(x, tx, Pik, ck=1, b0=FALSE)
sum(x*w)
tx
# The calibration estimation
colSums(estima*w)

##### Simple regression model without intercept #####

estima<-data.frame(Income, Employees)
x <- Taxes
tx <- sum(Lucy$Taxes)
w<-Wk(x,tx,Pik,ck=x,b0=FALSE)
sum(x*w)
tx
# The calibration estimation
colSums(estima*w)

##### Multiple regression model without intercept #####

estima<-data.frame(Income)
x <- cbind(Employees, Taxes)
tx <- c(sum(Lucy$Employees), sum(Lucy$Taxes))
w <- Wk(x,tx,Pik,ck=1,b0=FALSE)
sum(x[,1]*w)
sum(x[,2]*w)
tx
# The calibration estimation
colSums(estima*w)

##### Simple regression model with intercept #####

estima<-data.frame(Income, Employees)
x <- Taxes
tx <- c(N,sum(Lucy$Taxes))
w <- Wk(x,tx,Pik,ck=1,b0=TRUE)
sum(1*w)
sum(x*w)
tx
# The calibration estimation
colSums(estima*w)

##### Multiple regression model with intercept #####

estima<-data.frame(Income)
x <- cbind(Employees, Taxes)

```

```

tx <- c(N, sum(Lucy$Employees), sum(Lucy$Taxes))
w <- Wk(x,tx,Pik,ck=1,b0=TRUE)
sum(1*w)
sum(x[,1]*w)
sum(x[,2]*w)
tx
# The calibration estimation
colSums(estima*w)

#####
## Example 3: Linear models involving discrete auxiliary information
#####

# Draws a simple random sample without replacement
data(Lucy)
attach(Lucy)

N <- dim(Lucy)[1]
n <- 400
sam <- S.SI(N,n)
# The information about the units in the sample is stored in an object called data
data <- Lucy[sam,]
attach(data)
names(data)
# Vector of inclusion probabilities for units in the selected sample
Pik<-rep(n/N,n)
# The auxiliary information is discrete type
Doma<-Domains(Level)

##### Poststratified common mean model #####

estima<-data.frame(Income, Employees, Taxes)
tx <- colSums(Domains(Lucy$Level))
w <- Wk(Doma,tx,Pik,ck=1,b0=FALSE)
sum(Doma[,1]*w)
sum(Doma[,2]*w)
sum(Doma[,3]*w)
tx
# The calibration estimation
colSums(estima*w)

##### Poststratified common ratio model #####

estima<-data.frame(Income, Employees)
x<-Doma*Taxes
tx <- colSums(Domains(Lucy$Level))
w <- Wk(x,tx,Pik,ck=1,b0=FALSE)
sum(x[,1]*w)
sum(x[,2]*w)
sum(x[,3]*w)
tx
# The calibration estimation
colSums(estima*w)

```

Index

* datasets

- BigCity, [3](#)
- BigLucy, [4](#)
- Lucy, [56](#)

- BigCity, [3](#), [5](#), [57](#)
- BigLucy, [4](#), [4](#), [57](#)

- Deltakl, [6](#), [65](#), [90](#)
- Domains, [7](#), [54](#)

- E.1SI, [8](#), [11](#), [88](#)
- E.2SI, [9](#), [10](#), [33](#), [88](#)
- E.BE, [13](#), [70](#)
- E.Beta, [15](#), [37](#), [38](#), [93](#)
- E.piPS, [17](#), [19](#), [21](#), [22](#), [26](#), [45](#), [72](#)
- E.PO, [18](#), [19](#), [73](#)
- E.PPS, [20](#), [27](#), [42](#), [74](#)
- E.Quantile, [21](#)
- E.SI, [8](#), [14](#), [16](#), [22](#), [23](#), [29](#), [30](#), [37](#), [38](#), [45](#), [76](#), [88](#)
- E.STpiPS, [25](#), [27](#), [29](#), [77](#)
- E.STPPS, [27](#), [79](#)
- E.STSI, [8](#), [23](#), [26](#), [28](#), [55](#), [80](#), [81](#)
- E.SY, [29](#), [82](#)
- E.Trim, [31](#)
- E.UC, [11](#), [32](#)
- E.WR, [36](#), [83](#), [84](#)

- GREG.SI, [16](#), [23](#), [37](#), [93](#)

- HH, [21](#), [41](#), [74](#)
- HT, [42](#), [44](#), [90](#)

- Ik, [50](#), [51–53](#), [62](#), [65](#), [85](#)
- IkRS, [51](#), [86](#)
- IkWR, [52](#), [58](#), [59](#), [87](#)
- IPFP, [53](#)

- kish_allocation, [54](#)

- Lucy, [4](#), [5](#), [56](#)

- nk, [53](#), [57](#), [60](#), [87](#)

- OrderWR, [58](#)

- p.WR, [58](#), [60](#), [87](#)
- Pik, [6](#), [51](#), [61](#), [65](#)
- PikHol, [63](#), [66](#), [68](#)
- Pikl, [6](#), [51](#), [62](#), [64](#)
- PikPPS, [18](#), [62–64](#), [66](#), [68](#), [72](#), [73](#)
- PikSTPPS, [64](#), [66](#), [67](#), [77](#)

- S.BE, [14](#), [69](#)
- S.piPS, [18](#), [66](#), [71](#), [73](#), [74](#), [77](#)
- S.PO, [19](#), [70](#), [72](#)
- S.PPS, [21](#), [42](#), [60](#), [74](#), [79](#), [84](#)
- S.SI, [23](#), [70](#), [75](#), [81](#), [82](#), [84](#)
- S.STpiPS, [26](#), [68](#), [77](#), [79](#), [81](#)
- S.STPPS, [27](#), [72](#), [78](#)
- S.STSI, [29](#), [55](#), [76](#), [77](#), [80](#)
- S.SY, [30](#), [76](#), [81](#)
- S.WR, [37](#), [83](#)
- Support, [51](#), [84](#), [86](#)
- SupportRS, [52](#), [85](#), [85](#)
- SupportWR, [52](#), [53](#), [58–60](#), [85](#), [86](#)

- T.SIC, [88](#)

- VarHT, [6](#), [45](#), [65](#), [89](#)
- VarSYGHT, [90](#), [90](#)

- Wk, [38](#), [54](#), [92](#)