Package 'StratifiedSampling'

September 24, 2021

Type Package

Title Different Methods for Stratified Sampling

Version 0.3.0

Description Integrating a stratified structure in the population in a sampling design can considerably reduce the variance of the Horvitz-Thompson estimator. We propose in this package different methods to handle the selection of a balanced sample in stratified population. For more details see Raphaël Jauslin, Esther Eustache and Yves Tillé (2021) <arXiv:2101.05568>. The package propose also a method based on optimal transport and balanced sampling, see Raphaël Jauslin and Yves Tillé <arXiv:2105.08379>.

URL https://github.com/RJauslin/StratifiedSampling

BugReports https://github.com/RJauslin/StratifiedSampling/issues

License GPL (>= 2)

Encoding UTF-8

LinkingTo RcppArmadillo, Rcpp

Imports Rcpp, transport, proxy, MASS, sampling

Depends Matrix, R (>= 3.5.0)

Suggests knitr, rmarkdown, ggplot2, BalancedSampling, stats, testthat, StatMatch, laeken, prettydoc

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation yes

Author Raphael Jauslin [aut, cre] (<https://orcid.org/0000-0003-1088-3356>), Esther Eustache [aut], Bardia Panahbehagh [aut], Yves Tillé [aut] (<https://orcid.org/0000-0003-0904-5523>)

Maintainer Raphael Jauslin <raphael.jauslin@unine.ch>

Repository CRAN

Date/Publication 2021-09-24 12:20:02 UTC

R topics documented:

`balstrat` *Balanced Stratification*

Description

Select a stratified balanced sample. The function is similar to `balancedstratification` of the package sampling.

Usage

```
balstrat(X, strata, pik)
```

Arguments

`X`	A matrix of size (N x p) of auxiliary variables on which the sample must be balanced.
`strata`	A vector of integers that specifies the stratification.
`pik`	A vector of inclusion probabilities.

Details

The function implements the method proposed by Chauvet (2009). Firstly, a flight phase is performed on each strata. Secondly, a flight phase is applied on the whole population by aggregating the strata. Finally, a landing phase is applied by suppression of variables.

Value

A vector with elements equal to 0 or 1. The value 1 indicates that the unit is selected while the value 0 is for rejected units.

Author(s)

Raphaël Jauslin <raphael.jauslin@unine.ch>

References

Chauvet, G. (2009). Stratified balanced sampling. *Survey Methodology*, 35:115-119.

See Also

ffphase,landingRM

Examples

```
N <- 100
n <- 10
p <- 4
X <- matrix(rgamma(N*p,4,25),ncol = p)
strata <- as.matrix(rep(1:n,each = N/n))
pik <- rep(n/N,N)

s <- balstrat(X,strata,pik)

t(X/pik)%*%s
t(X/pik)%*%pik

Xcat <- disj(strata)

t(Xcat)%*%s
t(Xcat)%*%pik
```

---

bsmatch *Statistical matching using optimal transport and balanced sampling*

Description

We propose a method based on the output of the function otmatch. The method consists of choosing a unit from sample 2 to assign to a particular unit from sample 1.

Usage

```
bsmatch(object, Z2)
```

Arguments

object	A data.frame, output from the function otmatch.
Z2	A optional matrix, if we want to add some variables for the stratified balanced sampling step.

Details

All details of the method can be seen in the manuscript: Raphaël Jauslin and Yves Tillé (2021) <arXiv:2105.08379>.

Value

A list of two objects, A data.frame that contains the matching and the normalized weights. The first two columns of the data.frame contain the unit identities of the two samples. The third column are the final weights. All remaining columns are the matching variables.

See Also

otmatch, stratifiedcube

Examples

```
#--- SET UP
N=1000
p=5
X=array(rnorm(N*p),c(N,p))
EPS= 1e-9

n1=100
n2=200

s1=sampling::srswor(n1,N)
s2=sampling::srswor(n2,N)


id1=(1:N)[s1==1]
id2=(1:N)[s2==1]

d1=rep(N/n1,n1)
d2=rep(N/n2,n2)

X1=X[s1==1,]
X2=X[s2==1,]

#--- HARMONIZATION

re=harmonize(X1,d1,id1,X2,d2,id2)
w1=re$w1
w2=re$w2
```

```
#--- STATISTICAL MATCHING WITH OT

object = otmatch(X1,id1,X2,id2,w1,w2)

#--- BALANCED SAMPLING

out <- bsmatch(object)
```

---

calibRaking *Calibration using raking ratio*

Description

This function is inspired by the function `calib` of the package sampling. It computes the g-weights of the calibration estimator.

Usage

```
calibRaking(Xs, d, total, q, max_iter = 500L, tol = 1e-09)
```

Arguments

`Xs`	A matrix of calibration variables.
`d`	A vector, the initial weights.
`total`	A vector that represents the initial weights.
`q`	A vector of positive value that account for heteroscedasticity.
`max_iter`	An integer, the maximum number of iterations. Default = 500.
`tol`	A scalar that represents the tolerance value for the algorithm. Default = 1e-9.

Details

More details on the different calibration methods can be read in Tillé Y. (2020).

Value

A vector, the value of the g-weights.

Author(s)

Raphaël Jauslin <raphael.jauslin@unine.ch>

References

Tillé, Y. (2020). *Sampling and estimation from finite populations*. Wiley, New York

c_bound *C bound*

Description

This function is returning the number of unit that we need such that some conditions are fulfilled. See Details

Usage

```
c_bound(pik)
```

Arguments

pik	vector of the inclusion probabilities.

Details

The function is computing the number of unit K that we need to add such that the following conditions are fulfilled :

- $\sum_{k=1}^{K} \pi_k \geq 1$
- $\sum_{k=1}^{K} 1 - \pi_k \geq 1$
- Let c be the constant such that $\sum_{k=2}^{K} min(c\pi_k, 1) = n$, we must have that $\pi_1 \geq 1 - 1/c$

Value

An integer value, the number of units that we need to respect the constraints.

Author(s)

Raphael Jauslin <raphael.jauslin@unine.ch>

See Also

osod

disj *Disjunctive*

Description

This function transforms a categorical vector into a matrix of indicators.

Usage

```
disj(strata)
```

Arguments

strata	A vector of integers that represents the categories.

Value

A matrix of indicators.

Author(s)

Raphaël Jauslin <raphael.jauslin@unine.ch>

Examples

```
strata <- rep(c(1,2,3),each = 4)
disj(strata)
```

disjMatrix *Disjunctive for matrix*

Description

This function transforms a categorical matrix into a matrix of indicators variables.

Usage

```
disjMatrix(strata)
```

Arguments

strata	A matrix of integers that contains categorical vector in each column.

Value

A matrix of indicators.

Author(s)

Raphaël Jauslin <raphael.jauslin@unine.ch>

Examples

```
Xcat <-  matrix(c(sample(x = 1:6, size = 100, replace = TRUE),
            sample(x = 1:6, size = 100, replace = TRUE),
            sample(x = 1:6, size = 100, replace = TRUE)),ncol = 3)
disjMatrix(Xcat)
```

---

fbs *Fast Balanced Sampling*

Description

This function implements the method proposed by Hasler and Tillé (2014). It should be used for selecting a sample from highly stratified population.

Usage

```
fbs(X, strata, pik)
```

Arguments

X	A matrix of size (N x p) of auxiliary variables on which the sample must be balanced.
strata	A vector of integers that specifies the stratification.
pik	A vector of inclusion probabilities.

Details

Firstly a flight phase is performed on each strata. Secondly, several flight phases are applied by adding one by one the stratum. By doing this, some strata are managed on-the-fly. Finally, a landing phase is applied by suppression of the variables. If the number of element selected in each stratum is not equal to an integer, the function can be very time-consuming.

Value

A vector with elements equal to 0 or 1. The value 1 indicates that the unit is selected while the value 0 is for rejected units.

Author(s)

Raphaël Jauslin <raphael.jauslin@unine.ch>

References

Hasler, C. and Tillé Y. (2014). Fast balanced sampling for highly stratified population. *Computational Statistics and Data Analysis*, 74, 81-94

Examples

```
N <- 100
n <- 10
x1 <- rgamma(N,4,25)
x2 <- rgamma(N,4,25)

strata <- rep(1:n,each = N/n)

pik <- rep(n/N,N)
X <- as.matrix(cbind(matrix(c(x1,x2),ncol = 2)))

s <- fbs(X,strata,pik)

t(X/pik)%*%s
t(X/pik)%*%pik

Xcat <- disj(strata)

t(Xcat)%*%s
t(Xcat)%*%pik
```

---

ffphase *Fast flight phase of the cube method*

Description

This function computes the flight phase of the cube method proposed by Chauvet and Tillé (2006).

Usage

```
ffphase(X, pik)
```

Arguments

X	A matrix of size (N x p) of auxiliary variables on which the sample must be balanced.
pik	A vector of inclusion probabilities.

Details

This function implements the method proposed by (Chauvet and Tillé 2006). It recursively transforms the vector of inclusion probabilities `pik` into a sample that respects the balancing equations. The algorithm stops when the null space of the sub-matrix B is empty. For more information see (Chauvet and Tillé 2006).

The function uses the function `Null` to find the null space of the sub-matrix B.

Value

Updated vector of `pik` that contains 0 and 1 for unit that are rejected or selected.

Author(s)

Raphaël Jauslin <raphael.jauslin@unine.ch>

References

Chauvet, G. and Tillé, Y. (2006). A fast algorithm of balanced sampling. *Computational Statistics*, 21/1:53-62

See Also

`fastflightphase`, `flightphase`.

Examples

```
N <- 100
n <- 10
p <- 4

pik <- rep(n/N,N)
X <- cbind(pik,matrix(rgamma(N*p,4,25),ncol= p))

pikstar <- ffphase(X,pik)
t(X/pik)%*%pikstar
t(X/pik)%*%pik
pikstar
```

---

findB *Find best sub-matrix B in stratifiedcube*

Description

This function is computing a sub-matrix used in `stratifiedcube`.

Usage

```
findB(X, strata)
```

Arguments

X	A matrix of size (N x p) of auxiliary variables on which the sample must be balanced.
strata	A vector of integers that specifies the stratification.

Details

The function finds the smallest matrix B such that it contains only one more row than the number of columns. It consecutively adds the right number of rows depending on the number of categories that is added.

Value

A list of two components. The sub-matrix of X and the corresponding disjunctive matrix. If we use the function cbind to combine the two matrices, the resulting matrix has only one more row than the number of columns.

Author(s)

Raphaël Jauslin <raphael.jauslin@unine.ch>

Examples

```
N <- 1000
strata <-  sample(x = 1:6, size = N, replace = TRUE)

p <- 3
X <- matrix(rnorm(N*p),ncol = 3)
findB(X,strata)
```

---

harmonize *Harmonization by calibration*

Description

This function harmonize the two weight schemes such that the totals are equal.

Usage

```
harmonize(X1, d1, id1, X2, d2, id2, totals)
```

Arguments

X1	A matrix, the matching variables of sample 1.
d1	A numeric vector that contains the initial weights of the sample 1.
id1	A character or numeric vector that contains the labels of the units in sample 1.
X2	A matrix, the matching variables of sample 2.
d2	A numeric vector that contains the initial weights of the sample 1.
id2	A character or numeric vector that contains the labels of the units in sample 2.
totals	An optional numeric vector that contains the totals of the matching variables.

Details

All details of the method can be seen in the manuscript: Raphaël Jauslin and Yves Tillé (2021) <arXiv:>.

Value

A list of two vectors, the new weights of sample 1 (respectively new weights of sample 2).

Examples

```
#--- SET UP

N = 1000
p = 5
X = array(rnorm(N*p),c(N,p))

n1=100
n2=200

s1 = sampling::srswor(n1,N)
s2 = sampling::srswor(n2,N)

id1=(1:N)[s1==1]
id2=(1:N)[s2==1]

d1=rep(N/n1,n1)
d2=rep(N/n2,n2)

X1 = X[s1==1,]
X2 = X[s2==1,]

re <- harmonize(X1,d1,id1,X2,d2,id2)

colSums(re$w1*X1)
colSums(re$w2*X2)

#---  if the true totals is known
```

```
totals <- c(N,colSums(X))
re <- harmonize(X1,d1,id1,X2,d2,id2,totals)

colSums(re$w1*X1)
colSums(re$w2*X2)
colSums(X)
```

`inclprob` *Inclusion Probabilities*

Description

Computes first-order inclusion probabilities from a vector of positive numbers.

Usage

```
inclprob(x, n)
```

Arguments

`x`	vector of positive numbers.
`n`	sample size (could be a positive real value).

Details

The function is implemented in C++ so that it can be used in the code of other C++ functions. The implementation is based on the function `inclusionprobabilities` of the package sampling.

Value

A vector of inclusion probabilities proportional to `x` and such that the sum is equal to the value `n`.

Author(s)

Raphael Jauslin <raphael.jauslin@unine.ch>

See Also

`inclusionprobabilities`

Examples

```
x <- runif(100)
pik <- inclprob(x,70)
sum(pik)
```

`landingRM` *Landing by suppression of variables*

Description

This function performs the landing phase of the cube method using suppression of variables proposed by Chauvet and Tillé (2006).

Usage

```
landingRM(X, pikstar)
```

Arguments

`X`	matrix of auxiliary variables on which the sample must be balanced. (The matrix should be divided by the original inclusion probabilities.)
`pikstar`	vector of updated inclusion probabilities by the flight phase. See `ffphase`

Value

A vector with elements equal to 0 or 1. The value 1 indicates that the unit is selected while the value 0 is for rejected units.

Author(s)

Raphaël Jauslin <raphael.jauslin@unine.ch>

References

Chauvet, G. and Tillé, Y. (2006). A fast algorithm of balanced sampling. *Computational Statistics*, 21/1:53-62

See Also

`fbs`, `balstrat`.

Examples

```
N <- 1000
n <- 10
p <- 4
pik <- rep(n/N,N)
X <- cbind(pik,matrix(rgamma(N*p,4,25),ncol= p))
pikstar <- ffphase(X,pik)
s <- landingRM(X/pik,pikstar)
sum(s)
t(X/pik)%*%pik
t(X/pik)%*%pikstar
t(X/pik)%*%s
```

ncat *Number of categories*

Description

This function returns the number of factor in each column of a categorical matrix.

Usage

```
ncat(Xcat)
```

Arguments

Xcat	A matrix of integers that contains categorical vector in each column.

Value

A row vector that contains the number of categories in each column.

Author(s)

Raphaël Jauslin <raphael.jauslin@unine.ch>

Examples

```
Xcat <-  matrix(c(sample(x = 1:6, size = 100, replace = TRUE),
            sample(x = 1:6, size = 100, replace = TRUE),
            sample(x = 1:6, size = 100, replace = TRUE)),ncol = 3)
ncat(Xcat)
```

osod *One-step One Decision sampling method*

Description

This function implements the One-step One Decision method. It can be used using equal or unequal inclusion probabilities. The method is particularly useful for selecting a sample from a stream.

Usage

```
osod(pikr)
```

Arguments

pikr	A vector of inclusion probabilities.

Details

The method sequentially transforms the vector of inclusion probabilities into a sample whose values are equal to 0 or 1. The method respects the inclusion probabilities and can handle equal or unequal inclusion probabilities.

The method does not take into account the whole vector of inclusion probabilities by having a sequential implementation. This means that the method is fast and can be implemented in a flow.

Value

A vector with elements equal to 0 or 1. The value 1 indicates that the unit is selected while the value 0 is for rejected units.

Author(s)

Raphael Jauslin <raphael.jauslin@unine.ch>

See Also

c_bound

Examples

```
N <- 1000
n <- 100
pik <- inclprob(runif(N),n)
s <- osod(pik)
```

---

otmatch *Statistical Matching using Optimal transport*

Description

This function computes the statistical matching between two complex survey samples with weighting schemes. The function uses the function transport of the package transport.

Usage

```
otmatch(
  X1,
  id1,
  X2,
  id2,
  w1,
  w2,
  dist_method = "Euclidean",
```

```
  transport_method = "shortsimplex",
  EPS = 1e-09
)
```

Arguments

X1	A matrix, the matching variables of sample 1.
id1	A character or numeric vector that contains the labels of the units in sample 1.
X2	A matrix, the matching variables of sample 2.
id2	A character or numeric vector that contains the labels of the units in sample 1.
w1	A numeric vector that contains the weights of the sample 1, harmonized by the function harmonize.
w2	A numeric vector that contains the weights of the sample 2, harmonized by the function harmonize.
dist_method	A string that specified the distance used by the function dist of the package proxy. Default "Euclidean".
transport_method	A string that specified the distance used by the function transport of the package transport. Default "shortsimplex".
EPS	an numeric scalar to determine if the value is rounded to 0.

Details

All details of the method can be seen in : Raphaël Jauslin and Yves Tillé (2021) <arXiv:2105.08379>.

Value

A data.frame that contains the matching. The first two columns contain the unit identities of the two samples. The third column is the final weights. All remaining columns are the matching variables.

Examples

```
#--- SET UP
N=1000
p=5
X=array(rnorm(N*p),c(N,p))
EPS= 1e-9

n1=100
n2=200

s1 = sampling::srswor(n1,N)
s2 = sampling::srswor(n2,N)


id1=(1:N)[s1==1]
id2=(1:N)[s2==1]
```

```
d1=rep(N/n1,n1)
d2=rep(N/n2,n2)

X1=X[s1==1,]
X2=X[s2==1,]

#--- HARMONIZATION

re=harmonize(X1,d1,id1,X2,d2,id2)
w1=re$w1
w2=re$w2

#--- STATISTICAL MATCHING WITH OT

object = otmatch(X1,id1,X2,id2,w1,w2)


round(colSums(object$weight*object[,4:ncol(object)]),3)
round(colSums(w1*X1),3)
round(colSums(w2*X2),3)
```

`stratifiedcube` *Stratified Sampling*

Description

This function implements a method for selecting a stratified sample. It really improves the performance of the function `fbs` and `balstrat`.

Usage

```
stratifiedcube(X, strata, pik)
```

Arguments

`X`	A matrix of size (N x p) of auxiliary variables on which the sample must be balanced.
`strata`	A vector of integers that specifies the stratification..
`pik`	A vector of inclusion probabilities.

Details

The function is selecting a balanced sample very quickly even if the sum of inclusion probabilities within strata are non-integer. The function should be used in preference. Firstly, a flight phase is performed on each strata. Secondly, the function `findB` is used to find a particular matrix to apply a flight phase by using the cube method proposed by Chauvet, G. and Tillé, Y. (2006). Finally, a landing phase is applied by suppression of variables.

Value

A vector with elements equal to 0 or 1. The value 1 indicates that the unit is selected while the value 0 is for rejected units.

References

Chauvet, G. and Tillé, Y. (2006). A fast algorithm of balanced sampling. *Computational Statistics*, 21/1:53-62

See Also

fbs, balstrat, landingRM, ffphase

Examples

```
N <- 100
n <- 10
p <- 4
X <- matrix(rgamma(N*p,4,25),ncol = p)
strata <- as.matrix(rep(1:n,each = N/n))
pik <- rep(n/N,N)

s <- stratifiedcube(X,strata,pik)

t(X/pik)%*%s
t(X/pik)%*%pik

Xcat <- disj(strata)

t(Xcat)%*%s
t(Xcat)%*%pik
```

---

varApp *Approximated variance for balanced sampling*

Description

Approximated variance for balanced sampling

Usage

```
varApp(X, strata, pik, y)
```

Arguments

X	A matrix of size (N x p) of auxiliary variables on which the sample must be balanced.
strata	A vector of integers that represents the categories.
pik	A vector of inclusion probabilities.
y	A variable of interest.

Details

This function gives an approximation of the variance of the Horvitz-Thompson total estimator presented by Hasler and Tillé (2014).

Value

a scalar, the value of the approximated variance.

Author(s)

Raphaël Jauslin <raphael.jauslin@unine.ch>

References

Hasler, C. and Tillé, Y. (2014). Fast balanced sampling for highly stratified population. *Computational Statistics and Data Analysis*, 74:81-94.

See Also

varEst

Examples

```
N <- 1000
n <- 400
x1 <- rgamma(N,4,25)
x2 <- rgamma(N,4,25)

strata <- as.matrix(rep(1:40,each = 25)) # 25 strata
Xcat <- disjMatrix(strata)
pik <- rep(n/N,N)
X <- as.matrix(matrix(c(x1,x2),ncol = 2))

s <- stratifiedcube(X,strata,pik)

y <- 20*strata + rnorm(1000,120) # variable of interest
# y_ht <- sum(y[which(s==1)]/pik[which(s == 1)]) # Horvitz-Thompson estimator
# (sum(y_ht) - sum(y))^2 # true variance
varEst(X,strata,pik,s,y)
varApp(X,strata,pik,y)
```

varEst *Estimator of the approximated variance for balanced sampling*

Description

Estimator of the approximated variance for balanced sampling

Usage

```
varEst(X, strata, pik, s, y)
```

Arguments

X	A matrix of size (N x p) of auxiliary variables on which the sample must be balanced.
strata	A vector of integers that represents the categories.
pik	A vector of inclusion probabilities.
s	A sample (vector of 0 and 1, if rejected or selected).
y	A variable of interest.

Details

This function gives an estimator of the approximated variance of the Horvitz-Thompson total estimator presented by Hasler C. and Tillé Y. (2014).

Value

a scalar, the value of the estimated variance.

Author(s)

Raphaël Jauslin <raphael.jauslin@unine.ch>

References

Hasler, C. and Tillé, Y. (2014). Fast balanced sampling for highly stratified population. *Computational Statistics and Data Analysis*, 74:81-94.

See Also

varApp

Examples

```
N <- 1000
n <- 400
x1 <- rgamma(N,4,25)
x2 <- rgamma(N,4,25)

strata <- as.matrix(rep(1:40,each = 25)) # 25 strata
Xcat <- disjMatrix(strata)
pik <- rep(n/N,N)
X <- as.matrix(matrix(c(x1,x2),ncol = 2))

s <- stratifiedcube(X,strata,pik)

y <- 20*strata + rnorm(1000,120) # variable of interest
# y_ht <- sum(y[which(s==1)]/pik[which(s == 1)]) # Horvitz-Thompson estimator
# (sum(y_ht) - sum(y))^2 # true variance
varEst(X,strata,pik,s,y)
varApp(X,strata,pik,y)
```

Index