

Package ‘Spbsampling’

March 12, 2019

Title Spatially Balanced Sampling

Version 1.2.0

Description Selection of spatially balanced samples. In particular, the implemented sampling designs allow to select probability samples spread over the population of interest, in any dimension and using any distance function (e.g. Euclidean distance, Manhattan distance). For more details, Benedetti R and Piersimoni F (2017) <doi:10.1002/bimj.201600194> and Benedetti R and Piersimoni F (2017) <arXiv:1710.09116>. The implementation has been done in C++ through the use of 'Rcpp' and 'RcppArmadillo'.

Depends R (>= 3.1)

License GPL-3

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp

RoxygenNote 6.1.1

NeedsCompilation yes

Author Francesco Pantalone [aut, cre],
Roberto Benedetti [aut],
Federica Piersimoni [aut]

Maintainer Francesco Pantalone <pantalone.fra@gmail.com>

Repository CRAN

Date/Publication 2019-03-12 08:32:45 UTC

R topics documented:

hpwd	2
income_emilia	3
lucas_abruzzo	4
pwd	5
sbi	6

simul1	7
simul2	8
simul3	9
Spbsampling	10
stprod	11
stsum	12
swd	13

Index	15
--------------	-----------

hpwd	<i>Heuristic Product Within Distance (Spatially Balanced Sampling Design)</i>
------	---

Description

Selects spatially balanced samples through the use of Heuristic Product Within Distance design (HPWD). The level of the spread can be chosen through the parameter β , which is regulated by the exponent of the distance matrix ($D^1 \rightarrow \beta = 1$, $D^{10} \rightarrow \beta = 10$). The higher β is, the more the sample is going to be spread. To have constant inclusion probabilities $\pi_i = nsamp/N$, where $nsamp$ is sample size and N is population size, the distance matrix has to be standardized with function [stprod](#).

Usage

```
hpwd(dis, nsamp, nrepl = 1L)
```

Arguments

<code>dis</code>	A distance matrix NxN that specifies how far are all the pairs of units in the population.
<code>nsamp</code>	Sample size.
<code>nrepl</code>	Number of samples to draw (default = 1).

Details

The HPWD design generates samples approximately with the same probabilities of the [pwd](#) but with a significantly smaller number of steps. In fact, this algorithm randomly selects a sample of size n exactly with n steps, updating at each step the selection probability of not-selected units, depending on their distance from the units that were already selected in the previous steps.

Value

Return a matrix `nrepl x nsamp`, which contains the `nrepl` selected samples, each of them stored in a row. In particular, the i -th row contains all labels of units selected in the i -th sample.

References

Benedetti R, Piersimoni F (2017). “A spatially balanced design with probability function proportional to the within sample distance.” *Biometrical Journal*, **59**(5), 1067–1084.

Benedetti R, Piersimoni F (2017). “Fast Selection of Spatially Balanced Samples.” *arXiv*.

Examples

```
# Example 1
# Draw 50 samples of dimension 10 without constant probabilities and beta = 1
dis <- as.matrix(dist(cbind(lucas_abruzzo$x, lucas_abruzzo$y))) # distance matrix
samples <- hpwd(dis, 10, 50) # drawn samples

# Example 2
# Draw 100 samples of dimension 15 with constant probabilities equal to nsamp/N and beta = 1
# with N = population size
dis <- as.matrix(dist(cbind(lucas_abruzzo$x, lucas_abruzzo$y))) # distance matrix
vec <- rep(1, nrow(dis)) # vector of constraints
stand_dist <- stprod(dis, vec, 1e-15, 1000) # standardized matrix
samples <- hpwd(stand_dist, 15, 100) # drawn samples

# Example 3
# Draw 100 samples of dimension 15 with constant probabilities equal to nsamp/N and beta = 10
# with N = population size
dis <- as.matrix(dist(cbind(lucas_abruzzo$x, lucas_abruzzo$y))) # distance matrix
dis <- dis^10 # setting beta = 10
vec <- rep(1, nrow(dis)) # vector of constraints
stand_dist <- stprod(dis, vec, 1e-15, 1000) # standardized matrix
samples <- hpwd(stand_dist, 15, 100) # drawn samples
```

income_emilia

The income of municipalities of "Emilia Romagna".

Description

The dataset contains the total income of the municipalities in the region "Emilia Romagna", in Italy. Each municipality is defined by their own ISTAT (Istituto nazionale di statistica, Italy) code, a name and its geographical positions (through a pair of coordinates). For each of them we have the following auxiliary variables: province, number of taxpayers and total income of the municipality.

Usage

income_emilia

Format

A data frame with 334 rows and 7 variables:

municipality_code identification municipality code
municipality name of the municipality
province province of the municipality
numtaxpay number of taxpayers in the municipality
tot_inc average income of the municipality
x_coord coordinate x of the municipality
y_coord coordinate y of the municipality

Source

The dataset is a rearrangement from the data released by MEF - Dipartimento delle Finanze (Italy).

lucas_abruzzo	<i>LUCAS data for the region "Abruzzo", Italy.</i>
---------------	--

Description

The land use/cover area frame statistical survey, abbreviated as LUCAS, is a European field survey program funded and executed by Eurostat. Its objective is to set up area frame surveys for the provision of coherent and harmonised statistics on land use and land cover in the European Union (EU). Note that in LUCAS survey the concept of land is extended to inland water areas (lakes, river, coastal areas, etc.) and does not embrace uses below the earth's surface (mine deposits, subways, etc.). The LUCAS survey is a point survey, in particular the basic unit of observation is a circle with a radius of 1.5m (corresponding to an identifiable point on an orthophoto). In the classification there is a clear distinction between land cover and land use: land cover means physical cover ("material") observed at the earth's surface; land use means socio-economic function of the observed earth's surface. For each of both we assign a code to identified which type the point is. Land cover has 8 main categories, which are indicated by letter:

- A** artificial land
- B** cropland
- C** woodland
- D** shrubland
- E** grassland
- F** bareland
- G** water
- H** wetlands

Every main category has subclasses, which are indicated by the combination of the letter of the category and digits. Altogether there are 84 classes. Land use has 14 main categories. It has altogether 33 classes, which are indicated by the combination of the letter "U" and three digits.

Usage

```
lucas_abruzzo
```

Format

A data frame with 2699 rows and 7 variables:

id identified code for the unit spatial point

prov province

elev elevation of the unit spatial point, meant as the height above or below sea level

x coordinate x

y coordinate y

lc land cover code

lu land use code

Source

The dataset contains the data from LUCAS 2012 for the region "Abruzzo", Italy.

pwd

Product Within Distance (Spatially Balanced Sampling Design)

Description

Selects spatially balanced samples through the use of the Product Within Distance design (PWD). The level of the spread can be chosen through the parameter β , which is regulated by the exponent of the distance matrix ($D^1 \rightarrow \beta = 1$, $D^{10} \rightarrow \beta = 10$). The higher β is, the more the sample is going to be spread. To have constant inclusion probabilities $\pi_i = nsamp/N$, where $nsamp$ is sample size and N is population size, the distance matrix has to be standardized with function [stprod](#).

Usage

```
pwd(dis, nsamp, nrepl = 1L, niter = 10L)
```

Arguments

dis	A distance matrix NxN that specifies how far are all the pairs of units in the population.
nsamp	Sample size.
nrepl	Number of samples to draw (default = 1).
niter	Number of iterations for the algorithm. More iterations are better but require more time. Usually 10 is very efficient (default = 10).

Value

Return a matrix `nrepl x nsamp`, which contains the `nrepl` selected samples, each of them stored in a row. In particular, the *i*-th row contains all labels of units selected in the *i*-th sample.

References

Benedetti R, Piersimoni F (2017). “A spatially balanced design with probability function proportional to the within sample distance.” *Biometrical Journal*, **59**(5), 1067–1084.

Examples

```
# Example 1
# Draw 20 samples of dimension 15 without constant probabilities and with beta = 1
dis <- as.matrix(dist(cbind(lucas_abruzzo$x, lucas_abruzzo$y))) # distance matrix
nsamp <- 15 # sample size
nrepl <- 20 # number of samples to draw
niter <- 10 # number of iterations in the algorithm
samples <- pwd(dis, niter, nsamp, nrepl) # drawn samples

# Example 2
# Draw 20 samples of dimension 15 with constant probabilities equal to nsamp/N
# with N = population size
dis <- as.matrix(dist(cbind(lucas_abruzzo$x, lucas_abruzzo$y))) # distance matrix
nsamp <- 15 # sample size
nrepl <- 20 # number of samples to draw
niter <- 10 # number of iterations in the algorithm
vec <- rep(0, nrow(dis)) # vector of constraints
stand_dist <- stprod(dis, vec, 1e-15, 1000) # standardized matrix
samples <- pwd(stand_dist, niter, nsamp, nrepl) # drawn samples

# Example 3
# Draw 20 samples of dimension 15 with constant probabilities equal to nsamp/N and beta = 10
# with N = population size
dis <- as.matrix(dist(cbind(lucas_abruzzo$x, lucas_abruzzo$y))) # distance matrix
dis <- dis^10 # setting beta = 10
nsamp <- 15 # sample size
nrepl <- 20 # number of samples to draw
niter <- 10 # number of iterations in the algorithm
vec <- rep(0, nrow(dis)) # vector of constraints
stand_dist <- stprod(dis, vec, 1e-15, 1000) # standardized matrix
samples <- pwd(stand_dist, niter, nsamp, nrepl) # drawn samples
```

Description

Computes the Spatial Balance Index (SBI), which is a measure of balance of a sample. The lower it is, the better the balance.

Usage

```
sbi(dis, pi, s)
```

Arguments

<code>dis</code>	A distance matrix $N \times N$ that specifies how far are all the pairs of units in the population.
<code>pi</code>	A vector of first order inclusion probabilities of the units of the population.
<code>s</code>	A vector of labels of the sample.

Details

The SBI is based on Voronoi polygons. Given a sample s , each unit i in the sample has its own Voronoi polygon, which is composed by all population units closer to i than to any other sample unit j . Then, per each Voronoi polygon, define v_i as the sum of the inclusion probabilities of all units in the i -th Voronoi polygon. Finally, the variance of v_i is the SBI.

Value

Return the Spatial Balance Index.

References

Stevens DL, Olsen AR (2004). "Spatially Balanced Sampling of Natural Resources." *Journal of the American Statistical Association*, **99**(465), 262–278.

Examples

```
dis <- as.matrix(dist(cbind(simull$x, simull$y))) # distance matrix
vec <- rep(0, nrow(dis)) # vector of constraints
stand_dist <- stprod(dis, vec) # standardized matrix
nsamp <- 100 # sample size
pi <- rep(100 / nrow(dis), nrow(dis)) # vector of probabilities inclusion
sample <- pwd(stand_dist, 100) # sample
sbi(dis, pi, sample[1,])
```

simull

Simulated Population 1.

Description

The dataset contains a simulated georeferenced population of dimension $N = 1000$. The coordinates are generated in the range $[0, 1]$ as a simulated realization of a particular random point pattern: the Neyman-Scott process with Cauchy cluster kernel. The nine values for each unit are generated according to the outcome of a Gaussian stochastic process, with an intensity dependence parameter $\rho = 0.001$ (that means low dependence) and with no spatial trend.

Usage

simul1

Format

A data frame with 1000 rows and 11 variables:

x coordinate x

y coordinate y

z11 first value of the unit

z12 second value of the unit

z13 third value of the unit

z14 fourth value of the unit

z15 fifth value of the unit

z16 sixth value of the unit

z17 seventh value of the unit

z18 eighth value of the unit

z19 ninth value of the unit

Source

Benedetti R, Piersimoni F (2017). “A spatially balanced design with probability function proportional to the within sample distance.” *Biometrical Journal*, **59**(5), 1067–1084.

simul2

Simulated Population 2.

Description

The dataset contains a simulated georeferenced population of dimension $N = 1000$. The coordinates are generated in the range $[0, 1]$ as a simulated realization of a particular random point pattern: the Neyman-Scott process with Cauchy cluster kernel. The nine values for each unit are generated according to the outcome of a Gaussian stochastic process, with an intensity dependence parameter $\rho = 0.01$ (that means medium dependence) and with a spatial trend $x_1 + x_2 + \epsilon$.

Usage

simul2

Format

A data frame with 1000 rows and 11 variables:

x coordinate x

y coordinate y

z21 first value of the unit

z22 second value of the unit

z23 third value of the unit

z24 fourth value of the unit

z25 fifth value of the unit

z26 sixth value of the unit

z27 seventh value of the unit

z28 eighth value of the unit

z29 ninth value of the unit

Source

Benedetti R, Piersimoni F (2017). “A spatially balanced design with probability function proportional to the within sample distance.” *Biometrical Journal*, **59**(5), 1067–1084.

simul3

Simulated Population 3.

Description

The dataset contains a simulated georeferenced population of dimension $N = 1000$. The coordinates are generated in the range $[0, 1]$ as a simulated realization of a particular random point pattern: the Neyman-Scott process with Cauchy cluster kernel. The nine values for each unit are generated according to the outcome of a Gaussian stochastic process, with an intensity dependence parameter $\rho = 0.1$ (that means high dependence) and with a spatial trend $x_1 + x_2 + \epsilon$.

Usage

simul3

Format

A data frame with 1000 rows and 11 variables:

x coordinate x

y coordinate y

z31 first value of the unit

z32 second value of the unit

- z33** third value of the unit
- z34** fourth value of the unit
- z35** fifth value of the unit
- z36** sixth value of the unit
- z37** seventh value of the unit
- z38** eighth value of the unit
- z39** ninth value of the unit

Source

Benedetti R, Piersimoni F (2017). “A spatially balanced design with probability function proportional to the within sample distance.” *Biometrical Journal*, **59**(5), 1067–1084.

Spbsampling

Spatially balanced sampling designs

Description

Selection of spatially balanced samples. In particular, the implemented sampling designs allow to select probability samples spread over the population of interest, in any dimension and using any distance function (e.g. Euclidean distance, Manhattan distance). The implementation has been done in C++ through the use of Rcpp and RcppArmadillo.

Author(s)

Francesco Pantalone, Roberto Benedetti, Federica Piersimoni

Maintainer: Francesco Pantalone <pantalone.fra@gmail.com>

References

Benedetti R, Piersimoni F (2017). “A spatially balanced design with probability function proportional to the within sample distance.” *Biometrical Journal*, **59**(5), 1067–1084.

Benedetti R, Piersimoni F (2017). “Fast Selection of Spatially Balanced Samples.” *arXiv*.

stprod	<i>Standardize a symmetric matrix (distances) to fixed row (column) products</i>
--------	--

Description

Standardizes a distance matrix to fixed rows and columns products. The function iteratively constrains a logarithmic transformed matrix to known products, and in order to keep the symmetry of the matrix, at each iteration performs an average with its transpose. When the known products are all equal to a constant (e.g. 1), this method provides a simple and accurate way to scale a distance matrix to a doubly stochastic matrix.

Usage

```
stprod(mat, vec, differ = 1e-15, niter = 1000L)
```

Arguments

mat	A distance matrix size NxN.
vec	A vector of row (column) constraints.
differ	A scalar with the maximum accepted difference with the constraint (default = 1e-15).
niter	An integer with the maximum number of iterations (default = 1000).

Details

The standardized matrix will not be affected by problems arising from units with different inclusion probabilities caused by undesired features of the spatial distribution of the population, as edge effects and/or isolated points.

Value

Return a standardized distance matrix of size NxN.

References

Benedetti R, Piersimoni F (2017). "A spatially balanced design with probability function proportional to the within sample distance." *Biometrical Journal*, **59**(5), 1067–1084.

Examples

```
dis <- as.matrix(dist(cbind(simul1$x, simul1$y))) # distance matrix
con <- rep(0, nrow(dis)) # vector of constraints
stand_dist <- stprod(dis, con) # standardized matrix
```

stsum	<i>Standardize a symmetric matrix (distances) to fixed row (column) totals</i>
-------	--

Description

Standardizes a distance matrix to fixed rows and columns products. The function iteratively constrains the rows sums of the matrix to known totals, and in order to keep the symmetry of the matrix, at each iteration performs an average with its transpose. When the known totals are all equal to a constant (e.g. 1), this method provides a simple and accurate way to scale a distance matrix to a doubly stochastic matrix.

Usage

```
stsum(mat, vec, differ = 1e-15, niter = 1000L)
```

Arguments

mat	A distance matrix size NxN.
vec	A vector of row (column) constraints.
differ	A scalar with the maximum accepted difference with the constraint (default = 1e-15).
niter	An integer with the maximum number of iterations (default = 1000).

Details

The standardized matrix will not be affected by problems arising from units with different inclusion probabilities caused by undesired features of the spatial distribution of the population, as edge effects and/or isolated points.

Value

Return a standardized distance matrix of size NxN.

References

Benedetti R, Piersimoni F (2017). “A spatially balanced design with probability function proportional to the within sample distance.” *Biometrical Journal*, **59**(5), 1067–1084.

Examples

```
dis <- as.matrix(dist(cbind(simul2$x, simul2$y))) # distance matrix
con <- rep(1, nrow(dis)) # vector of constraints
stand_dist <- stsum(dis, con) # standardized matrix
```

swd

*Sum Within Distance (Spatially Balanced Sampling Design)***Description**

Selects spatially balanced samples through the use of the Sum Within Distance design (SWD). To have a constant inclusion probabilities $\pi_i = nsamp/N$, where *nsamp* is sample size and *N* is population size, the distance matrix has to be standardized with function [stsum](#).

Usage

```
swd(dis, nsamp, bexp, nrepl = 1L, niter = 10L)
```

Arguments

<code>dis</code>	A distance matrix NxN that specifies how far are all the pairs of units in the population.
<code>nsamp</code>	Sample size.
<code>bexp</code>	Parameter β for the algorithm. The higher β is, the more the sample is going to be spread.
<code>nrepl</code>	Number of samples to draw (default = 1).
<code>niter</code>	Number of iterations for the algorithm. More iterations are better but require more time. Usually 10 is very efficient (default = 10).

Value

Return a matrix `nrepl x nsamp`, which contains the `nrepl` selected samples, each of them stored in a row. In particular, the *i*-th row contains all labels of units selected in the *i*-th sample.

References

Benedetti R, Piersimoni F (2017). "A spatially balanced design with probability function proportional to the within sample distance." *Biometrical Journal*, **59**(5), 1067–1084.

Examples

```
# Example 1
# Draw 20 samples of dimension 15 without constant probabilities and beta = 1
dis <- as.matrix(dist(cbind(income_emilia$x_coord, income_emilia$y_coord))) # distance matrix
nsamp <- 15 # sample size
nrepl <- 20 # number of samples to draw
niter <- 10 # number of iterations in the algorithm
bexp <- 10 # parameter beta
samples <- swd(dis, niter, nsamp, nrepl, bexp) # drawn samples

# Example 2
# Draw 20 samples of dimension 15 with constant probabilities equal to nsamp/N and beta = 10
```

```
# with N = population size
dis <- as.matrix(dist(cbind(income_emilia$x_coord, income_emilia$y_coord))) # distance matrix
nsamp <- 15 # sample size
nrepl <- 20 # numbers of samples to draw
niter <- 10 # numbers of iterations in the algorithm
bexp <- 10 # parameter beta
vec <- rep(1, nrow(dis)) # vector of constraints
stand_dist <- stsum(dis, vec, 1e-15, 1000) # standardized matrix
samples <- swd(stand_dist, niter, nsamp, nrepl, bexp) # drawn samples
```

Index

*Topic **datasets**

income_emilia, [3](#)

lucas_abruzzo, [4](#)

simul1, [7](#)

simul2, [8](#)

simul3, [9](#)

hpwd, [2](#)

income_emilia, [3](#)

lucas_abruzzo, [4](#)

pwd, [2](#), [5](#)

sbi, [6](#)

simul1, [7](#)

simul2, [8](#)

simul3, [9](#)

Spbsampling, [10](#)

Spbsampling-package (Spbsampling), [10](#)

stprod, [2](#), [5](#), [11](#)

stsum, [12](#), [13](#)

swd, [13](#)