

Package ‘BuSuCo’

September 8, 2026

Type Package

Title Coordinate Survey Samples for Businesses

Version 0.2.0

Description Contains a number of functions for different business sample coordination algorithms together with a function to generate customizable business data sets. For more details see Straubinger (2025) <<https://ubt.opus.hbz-nrw.de/frontdoor/index/index/docId/2785>>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Imports data.table, EnvStats

Config/testthat/edition 3

NeedsCompilation no

Author Johannes Straubinger [aut, cre],
Ralf Münnich [aut]

Maintainer Johannes Straubinger <johannes.straubinger@statistik.gv.at>

Repository CRAN

Date/Publication 2026-09-08 13:40:29 UTC

Contents

Dutch.BSC	2
French.BSC	4
Gen.Data	7
is.in.interval	10
nearest.neighbour	10
Norwegian.BSC	11
random.order	13
Stratify	14
Survey.relevance	15

Swiss.BSC	16
target.corr.sort	19
uneq.sample	20

Index	22
--------------	-----------

Dutch.BSC	<i>Applying the Modified Dutch Business Sample Coordination Algorithm</i>
-----------	---

Description

The function applies the Dutch business sample coordination algorithm as described by Smeets & Boonstra (2018) with some modifications. These modifications can be found in Straubinger(2025).

Usage

```
Dutch.BSC(  
  DATA,  
  period,  
  sf,  
  rf,  
  years,  
  chi_S,  
  survey.prio,  
  burden.periods,  
  n.min = 3,  
  alpha = 0,  
  vita = 0,  
  ggamma = 0,  
  seed = 1,  
  Survey.relevant,  
  stratification  
)
```

Arguments

DATA	A data table. Can be generated with Gen.Data. It must include the variables ID, value_P1, value_corr_P1, and domain_P1. The latter three must further exist for every period which the surveys shall cover. The function returns the initial data set with some modifications as output which can be used as input for the next draw.
period	An integer. Defines the period in which the sample shall be drawn. Must be smaller or equal to the number of periods available in DATA, Survey.relevant, and in stratification.
sf	A numeric vector. Defines the sampling fraction for each survey.
rf	A numeric vector. Defines the rotation fraction for each survey in the current period.

years	An integer vector. Defines the periodicity of all surveys. E.g. annual surveys would receive 1, biannual surveys 2, etc.
chi_S	A numeric vector. Defines the time needed to answer to each survey in hours.
survey.prio	An integer vector. Defines the order of surveys, i.e. surveys which shall be drawn first should be placed in front.
burden.periods	An integer. Defines the number of periods which shall be taken into account for the calculation of the accumulated response burden.
n.min	An integer. Defines the minimum sample size over all surveys. Default value is 3.
alpha	A numeric value. Defines to which degree the stratum size of an element shall be taken into account for the calculation of the accumulated response burden. Default value is 0, i.e. the stratum size will not be taken into account at all.
vita	A numeric value. Defines to which degree the enterprise size of an element shall be taken into account for the calculation of the accumulated response burden. Default value is 0, i.e. the enterprise size will not be taken into account at all.
ggamma	A numeric value. Defines how much enterprises which are spared from any survey over more than one period shall receive a lower accumulated response burden. Default value is 0, i.e. pause periods will not influence the calculation of the accumulated response burden at all. If $ggamma < 0$, the accumulated response burden will be reduced after a longer pause duration. If $ggamma > 0$, accumulated response burden will increase after a longer pause duration.
seed	An integer. Sets the seed for any random draws happening inside the function. Default value is 1.
Survey.relevant	An array of logical values. Can be generated with Survey.relevance The dimensions cover the number of periods, the number of surveys, and the number of elements in DATA.
stratification	An array of character strings. Can be generated with Stratify. The dimensions cover the number of periods, the number of surveys, and the number of elements in DATA.

Value

A list of two (DATA, in.sample) is returned. DATA is the original data set given under DATA with some updates on response burden indicators. It can be used as input for further runs of the function. in.sample. is a matrix containing logical values. Its dimensions are $nrow(DATA)$ and the number of surveys drawn. It shows for each elements and each survey whether whether it was drawn or not.

References

- Smeets, M., & Boonstra, J. H. (2018): *Sampling Coordination of Business Surveys at Statistics Netherlands* In: B. Lorenc et al. (Eds.), *The Unit Problem and Other Current Topics in Business Survey Methodology* (pp. 1-6). Newcastle upon Tyne, UK: Cambridge Scholars Publishing.
- Straubinger, J. (2025): *Methods to Coordinate Survey Samples in the Context of German Business Statistics* (PhD Thesis). Universität Trier, Germany.

See Also

[`'data.table'`][`data.table::data.table`] for more information on `data.table` objects.

Examples

```
DATA <- Gen.Data(N = 100, periods = 5, D = 2, D.probs = c(.8,.2), H = 2,
H.size = c(.1,.3), borders = c(100))

Sur_rel <- Survey.relevance(DATA, periods = 5, years = c(1,2),
domains.survey = list(1:2,1), ge.or.less = c("ge", "1"),
TO.or.EM = c("TO","EM"), size.class = c(0,5))

strat <- Stratify(DATA, Survey.relevant = Sur_rel, periods = 5,
years = c(1,2), TO.or.EM = c("TO","EM"), strata.TO = c(500), strata.EM = c(10))

S1 <- Dutch.BSC(DATA, period=1, sf=c(.2,.5), rf=c(.5,1), years=c(1,2),
chi_S=c(.5,2), survey.prio=1:2, burden.periods=5, n.min = 3, alpha=0, vita=0,
ggamma=0, Survey.relevant=Sur_rel, stratification=strat)

S2 <- Dutch.BSC(DATA = S1$DATA, period=2, sf=c(.2,.5), rf=c(.5,1),
years=c(1,2), chi_S=c(.5,2),survey.prio=1:2, burden.periods=5, n.min = 3,
alpha=0, vita=0, ggamma=0, Survey.relevant=Sur_rel, stratification=strat)
```

French.BSC

Applying the Modified French Business Sample Coordination Algorithm

Description

The function applies the French business sample coordination algorithm as described by Guggemos & Sautory (2012) and Gros (2015) with some modifications. These modifications can be found in Straubinger (2025).

Usage

```
French.BSC(
  DATA,
  period,
  sf,
  rf,
  years,
  chi_S,
  survey.prio,
  burden.periods,
  n.min = 3,
  alpha = 0,
```

```

    vita = 0,
    ggamma = 0,
    seed = 1,
    Survey.relevant,
    stratification
)

```

Arguments

DATA	A data table. Can be generated with Gen.Data. It must include the variables ID, value_P1, value_corr_P1, and domain_P1. The latter three must further exist for every period which the surveys shall cover. The function returns the initial data set with some modifications as output which can be used as input for the next draw.
period	An integer. Defines the period in which the sample shall be drawn. Must be smaller or equal to the number of periods available in DATA, Survey.relevant, and in stratification.
sf	A numeric vector. Defines the sampling fraction for each survey.
rf	A numeric vector. Defines the rotation fraction for each survey in the current period.
years	An integer vector. Defines the periodicity of all surveys. E.g. annual surveys would receive 1, biannual surveys 2, etc.
chi_S	A numeric vector. Defines the time needed to answer to each survey in hours.
survey.prio	An integer vector. Defines the order of surveys, i.e. surveys which shall be drawn first should be placed in front.
burden.periods	An integer. Defines the number of periods which shall be taken into account for the calculation of the accumulated response burden.
n.min	An integer. Defines the minimum sample size over all surveys. Default value is 3.
alpha	A numeric value. Defines to which degree the stratum size of an element shall be taken into account for the calculation of the accumulated response burden. Default value is 0, i.e. the stratum size will not be taken into account at all.
vita	A numeric value. Defines to which degree the enterprise size of an element shall be taken into account for the calculation of the accumulated response burden. Default value is 0, i.e. the enterprise size will not be taken into account at all.
ggamma	A numeric value. Defines how much enterprises which are spared from any survey over more than one period shall receive a lower accumulated response burden. Default value is 0, i.e. pause periods will not influence the calculation of the accumulated response burden at all. If $ggamma < 0$, the accumulated response burden will be reduced after a longer pause duration. If $ggamma > 0$, accumulated response burden will increase after a longer pause duration.
seed	An integer. Sets the seed for any random draws happening inside the function. Default value is 1.

Survey.relevant

An array of logical values. Can be generated with Survey.relevance The dimensions cover the number of periods, the number of surveys, and the number of elements in DATA.

stratification

An array of character strings. Can be generated with Stratify. The dimensions cover the number of periods, the number of surveys, and the number of elements in DATA.

Value

A list of two (DATA, in.sample) is returned. DATA is the original data set given under DATA with some updates on response burden indicators. It can be used as input for further runs of the function. in.sample is a matrix containing logical values. Its dimensions are nrow(DATA) and the number of surveys drawn. It shows for each element and each survey whether it was drawn or not.

References

Gros, E. (2015): *Études et Simulations Relatives à la Nouvelle Méthode de Coordination des Échantillons d'Enquêtes Entreprises et Établissements Développé à l'Insee*. In: Journées de Méthodologie Statistique de l'Insee, 2015, pp. 1-23.

Guggemos, F. & Sautory, O. (2012): *Sampling Coordination of Business Surveys Conducted by Insee*. In: Proceedings of the Fourth International Conference of Establishment Surveys. Montréal, Canada.

Straubinger, J. (2025): *Methods to Coordinate Survey Samples in the Context of German Business Statistics* (PhD Thesis). Universität Trier, Germany.

See Also

[‘data.table’][data.table::data.table] for more information on data.table objects.

Examples

```
DATA <- Gen.Data(N = 100, periods = 5, D = 2, D.probs = c(.8,.2), H = 2,
H.size = c(.1,.3), borders = c(100))
```

```
Sur_rel <- Survey.relevance(DATA, periods = 5, years = c(1,2),
domains.survey = list(1:2,1), ge.or.less = c("ge", "l"),
T0.or.EM = c("T0","EM"), size.class = c(0,5))
```

```
strat <- Stratify(DATA, Survey.relevant = Sur_rel, periods = 5,
years = c(1,2), T0.or.EM = c("T0","EM"), strata.T0 = c(500), strata.EM = c(10))
```

```
S1 <- French.BSC(DATA, period=1, sf=c(.2,.5), rf=c(.5,1), years=c(1,2),
chi_S=c(.5,2), survey.prio=1:2, burden.periods=5, n.min = 3, alpha=0, vita=0,
ggamma=0, Survey.relevant=Sur_rel, stratification=strat)
```

```
S2 <- French.BSC(DATA = S1$DATA, period=2, sf=c(.2,.5), rf=c(.5,1),
years=c(1,2), chi_S=c(.5,2),survey.prio=1:2, burden.periods=5, n.min = 3,
alpha=0, vita=0, ggamma=0, Survey.relevant=Sur_rel, stratification=strat)
```

Gen.Data

*Generate synthetic business data***Description**

Function to generate a synthetic business data set with two Pareto-distributed numeric variables over a desired number of time periods. The function allows the user to further customise the number of domains, the number of strata, and population dynamics among others.

Usage

```
Gen.Data(
  N,
  periods,
  D,
  D.probs,
  H,
  H.mode = "border",
  H.size = NA,
  borders = NA,
  births = NA,
  deaths = NA,
  corr.B.U = 0.75,
  Ratio.B.U = 200,
  n.jump = NA,
  jump.mode = "both",
  jump.period = NA,
  h.dynamics = "both",
  d.mode = "automatic",
  d.dynamics = NA,
  d.leavers = NA,
  d.period = NA,
  seed = 1
)
```

Arguments

N	An integer. Defines the size of the data set.
periods	An integer. Defines the number of periods of the data set.
D	An integer. Defines the number of domains.
D.probs	A numeric vector. Defines the probabilities of domain membership. <code>length(D.probs)</code> must be equal to D. If D.probs do not sum up to 1, domain membership will not match the given probabilities.
H	An integer. Defines the number so size class strata in the data set.

H.mode	A character string. Indicates how size class strata shall be constructed. If "size" is chosen, size class strata will be built by defining how many elements shall be in each stratum. For this, H.size must be specified. If "border" is chosen, strata will be built by defining where the stratum borders shall be. For this, borders must be specified.
H.size	A numeric vector. Required, if H.mode == "size". Defines the proportional size of size class strata of the whole population. $\text{length}(\text{H.size}) == H$ must hold. If $\text{sum}(\text{H.size}) \neq 1$, the function will adjust the sizes automatically.
borders	A numeric vector. Required, if H.mode == "border". Defines the borders of size class strata. $\text{length}(\text{borders}) == H-1$ must hold. If borders are chosen too far off, the number of strata can be smaller than desired.
births	A vector of integers > 0. Defines the number of births per period. $\text{length}(\text{births}) == \text{periods}-1$ must hold. If not specified, the function will automatically assign 0 births per period.
deaths	A vector of integers > 0. Defines the number of deaths per period. $\text{length}(\text{deaths}) == \text{periods}-1$ must hold. If not specified, the function will automatically assign 0 deaths per period.
corr.B.U	Numeric. Defines the desired correlation between the two numeric variables which will be created. $-1 \leq \text{corr.B.U} \leq 1$ must hold.
Ratio.B.U	Numeric. Defines the (approximate) ratio between the two numeric variables which will be created. $\text{Ratio.B.U} > 0$ is required.
n.jump	A vector of integers > 0. Defines the number of stratum jumpers per period. $\text{length}(\text{n.jump})$ must be equal to H. If not specified, the function will automatically assign 0 stratum jumpers per period.
jump.mode	A character string. Defines the direction of stratum jumpers. If "up" is chosen, only stratum jumpers changing from smaller to larger size strata are created. If "down" is chosen, only stratum jumpers changing from larger to smaller size strata are created. In either of these two cases, strata with few elements might be emptied after time. If "both" is chosen (default), stratum jumpers into both directions will be created.
jump.period	A numeric vector. Allows for customisation of the number of stratum jumpers per period. Each entry is a factor with which n.jump will be multiplied in each period. $\text{length}(\text{jump.period}) == \text{periods}-1$ must hold. If not specified, the function will automatically assign 1 as a factor for each period, i.e. there will be the same number of stratum jumpers each period.
h.dynamics	A character string. Defines the mode of population dynamics. If "jump.only" is chosen, all elements will keep their values of both numeric variables, except for stratum jumpers. If "e.term" is chosen, an error term will be added to the values of both numeric variables. In this case, the number of stratum jumpers will be random. If "both" is chosen (default), the desired number of stratum jumpers will occur while all other elements will be added an error term on the values of both numeric variables.
d.mode	A character string. Defines the mode of domain jumpers. If "automatic" is chosen (default), elements will remain with a defined probability in their domain. If "manual" is chosen, the user can define the number of domain jumpers per domain per period.

d.dynamics	A numeric vector. Defines the probabilities for elements to remain in their domain if d.mode == "automatic", length(d.dynamics) must be equal to D. If not specified, the function will automatically assign 1 as a factor for each period, i.e. there will be no domain jumpers.
d.leavers	A numeric vector. Defines the number of domain jumpers per domain. length(d.leavers) must be equal to D. Only necessary, if d.mode == "manual". If not specified, the function will automatically assign 0 domain jumpers per domain.
d.period	A numeric vector. Allows for customisation of the number of domain jumpers per period. Each entry is a factor with which d.leavers will be multiplied in each period. length(d.period) == periods-1 must hold. If not specified, the function will automatically assign 1 as a factor for each period, i.e. there will be the same number of domain jumpers each period.
seed	An integer. An integer. Sets the first seed for randomly reordering of y within target.corr.sort which is part of the function. Default is 1.

Value

A data table of the generated data set is returned. The rows of the data set are the single elements. The columns are structured as follows. ID is the ID of each element. value_P1 are the values of the first numeric variable of the data set. It is intended to resemble a turnover distribution. value_corr_P1 are the values of the second numeric variable of the data set. It is intended to resemble the number of employees and its values are correlated with a correlation of (approximately) corr.B.U. If periods > 1, this correlation might decrease, depending on the values given to the various parameters determining population dynamics. domain_P1 and stratum_P1 are the domain and stratum membership indicators. If periods > 1, all variables except for ID will be continued in each period.

See Also

[`'data.table'`][`data.table::data.table`] for more information on data.table objects.

Examples

```
DATA <- Gen.Data(N = 100, periods = 5, D = 2, D.probs = c(.8,.2), H = 2,
borders = c(500))
```

```
DATA <- Gen.Data(N = 100, periods = 3, D = 2, D.probs = c(.8,.2), H = 2,
borders = c(500), births = c(5,5), deaths = c(5,5))
```

```
DATA <- Gen.Data(N = 1000, periods = 3, D = 2, D.probs = c(.5,.5), H = 2,
H.mode = "size", H.size = c(.8,.2), n.jump = c(10,10), jump.period = c(1,.5))
```

```
DATA <- Gen.Data(N = 1000, periods = 3, D = 2, D.probs = c(.8,.2), H = 2,
borders = c(500), d.mode = "manual", d.leavers = c(10,2), d.period = c(.5,1))
```

is.in.interval	<i>Check whether a number is within a given interval</i>
----------------	--

Description

The function checks whether a number is within a given interval by comparing the number to the borders of the interval. The borders can be included, if wished.

Usage

```
is.in.interval(x, borders, include.borders = FALSE)
```

Arguments

x	A number, must be numeric.
borders	A numeric vector of length 2 indicating the borders of the interval.
include.borders	Logical. If FALSE (default), the function will return FALSE, if x sits exactly at one of the two borders. If TRUE, the function will return TRUE, if x sits exactly at one of the two borders.

Value

Logical. Either TRUE or FALSE is returned

Examples

```
x <- 1
borders <- c(0,2)
is.in.interval(x,borders)

x <- 2
borders <- c(0,2)
is.in.interval(x,borders)
is.in.interval(x,borders, include.borders = TRUE)
```

nearest.neighbour	<i>Find the nearest neighbour of one or more elements</i>
-------------------	---

Description

The function finds the nearest neighbour of all elements marked as recipients in status.

Usage

```
nearest.neighbour(status, value, id)
```

Arguments

status	A logical vector. All elements with TRUE will be the recipients and all elements with FALSE will be the donors.
value	A numeric vector. Contains the values of which the nearest neighbour shall be found.
id	An integer vector. Contains the IDs of all elements.

Value

A vector of the IDs of the nearest neighbours to the recipients among the donor values. If all(status == TRUE), the recipients' IDs will be returned instead.

Examples

```
status <- sample(c(TRUE,FALSE), 10, replace = TRUE)
value <- stats::rnorm(10)
id <- 1:10
nearest.neighbour(status, value, id)
```

Norwegian.BSC

Applying the Modified Norwegian Business Sample Coordination Algorithm

Description

The function applies the Norwegian business sample coordination algorithm as described by Zhang (2009) with some modifications. These modifications can be found in Straubinger (2025).

Usage

```
Norwegian.BSC(
  DATA,
  period,
  sf,
  rf,
  years,
  chi_S,
  survey.prio,
  n.min = 3,
  alpha = 0,
  vita = 0,
  seed = 1,
```

```

    Survey.relevant,
    stratification
)

```

Arguments

DATA	A data table. Can be generated with Gen.Data. It must include the variables ID, value_P1, value_corr_P1, and domain_P1. The latter three must further exist for every period which the surveys shall cover. The function returns the initial data set with some modifications as output which can be used as input for the next draw.
period	An integer. Defines the period in which the sample shall be drawn. Must be smaller or equal to the number of periods available in DATA, Survey.relevant, and in stratification.
sf	A numeric vector. Defines the sampling fraction for each survey.
rf	A numeric vector. Defines the rotation fraction for each survey in the current period.
years	An integer vector. Defines the periodicity of all surveys. E.g. annual surveys would receive 1, biannual surveys 2, etc.
chi_S	A numeric vector. Defines the time needed to answer to each survey in hours.
survey.prio	An integer vector. Defines the order of surveys, i.e. surveys which shall be drawn first should be placed in front.
n.min	An integer. Defines the minimum sample size over all surveys. Default value is 3.
alpha	A numeric value. Defines to which degree the stratum size of an element shall be taken into account for the calculation of the accumulated response burden. Default value is 0, i.e. the stratum size will not be taken into account at all.
vita	A numeric value. Defines to which degree the enterprise size of an element shall be taken into account for the calculation of the accumulated response burden. Default value is 0, i.e. the enterprise size will not be taken into account at all.
seed	An integer. Sets the seed for any random draws happening inside the function. Default value is 1.
Survey.relevant	An array of logical values. Can be generated with Survey.relevance The dimensions cover the number of periods, the number of surveys, and the number of elements in DATA.
stratification	An array of character strings. Can be generated with Stratify. The dimensions cover the number of periods, the number of surveys, and the number of elements in DATA.

Value

A list of two (DATA, in.sample) is returned. DATA is the original data set given under DATA with some updates on response burden indicators. It can be used as input for further runs of the function. in.sample is a matrix containing logical values. Its dimensions are nrow(DATA) and the number of surveys drawn. It shows for each element and each survey whether it was drawn or not.

References

Zhang, L.-C. (2009): *Coordination of business surveys* In: European Establishment Statistics Workshop (EESW09). Stockholm, Sweden.

Straubinger, J. (2025): *Methods to Coordinate Survey Samples in the Context of German Business Statistics* (PhD Thesis). Universität Trier, Germany.

See Also

[`'data.table'`][`data.table::data.table`] for more information on `data.table` objects.

Examples

```
DATA <- Gen.Data(N = 100, periods = 5, D = 2, D.probs = c(.8,.2), H = 2,
H.size = c(.1,.3), borders = c(100))

Sur_rel <- Survey.relevance(DATA, periods = 5, years = c(1,2),
domains.survey = list(1:2,1), ge.or.less = c("ge", "1"),
T0.or.EM = c("T0","EM"), size.class = c(0,5))

strat <- Stratify(DATA, Survey.relevant = Sur_rel, periods = 5,
years = c(1,2), T0.or.EM = c("T0","EM"), strata.T0 = c(500), strata.EM = c(10))

S1 <- Norwegian.BSC(DATA, period=1, sf=c(.2,.5), rf=c(.5,1), years=c(1,2),
chi_S=c(.5,2), survey.prio=1:2, n.min = 3, alpha=0, vita=0,
Survey.relevant=Sur_rel, stratification=strat)

S2 <- Norwegian.BSC(DATA = S1$DATA, period=2, sf=c(.2,.5), rf=c(.5,1),
years=c(1,2), chi_S=c(.5,2), survey.prio=1:2, n.min = 3, alpha=0, vita=0,
Survey.relevant=Sur_rel, stratification=strat)
```

random.order

Order elements of a data set randomly

Description

The function randomly reorders the elements of a data set.

Usage

```
random.order(dat, seed)
```

Arguments

<code>dat</code>	A data frame or a data table.
<code>seed</code>	An integer. Sets the seed for the random re-ordering of the data set.

Value

A data frame. The data set given in `dat` is now randomly reordered.

See Also

`['data.table']``[data.table::data.table]` for more information on `data.table` objects.

Examples

```
DATA <- Gen.Data(N = 100, periods = 5, D = 2, D.probs = c(.8,.2), H = 2,
H.size = c(.1,.3), borders = c(100))

DATA <- random.order(DATA,1)
```

Stratify

Create an array indicating the stratum membership

Description

The function creates an array of character strings indicating the stratum membership of each element in each time period.

Usage

```
Stratify(DATA, Survey.relevant, periods, years, T0.or.EM, strata.T0, strata.EM)
```

Arguments

DATA	A data table. Can be generated with <code>Gen.Data</code> . It must include the variables <code>ID</code> , <code>value_P1</code> , <code>value_corr_P1</code> , and <code>domain_P1</code> . The latter three must further exist for every period which the surveys shall cover.
Survey.relevant	An array of logical values. Can be generated with <code>Survey.relevance</code> . The dimensions cover the number of periods, the number of surveys, and the number of elements in <code>DATA</code> .
periods	An integer. Defines the number of periods which the surveys shall cover.
years	An integer vector. Defines the periodicity of all surveys. E.g. annual surveys would receive 1, biannual surveys 2, etc.
T0.or.EM	A vector of character strings. Defines whether elements shall be marked as relevant for each survey based on turnover values or number of employee values. Only "TO" for turnover and "EM" for the number of employees are allowed as values.
strata.T0	A numeric vector. Defines the stratum borders when turnover shall be used for stratification. Minimum and maximum values do not need to be defined and will be added automatically by the function.

`strata.EM` A numeric vector. Defines the stratum borders when the number of employees shall be used for stratification. Minimum and maximum values do not need to be defined and will be added automatically by the function.

Value

An array of character strings indicating the stratum membership of each element. The dimensions cover the number of periods, the number of surveys, and the number of elements in DATA. It can be used directly as input for the survey coordination functions Dutch.BSC, French.BSC, and Norwegian.BSC.

Examples

```
DATA <- Gen.Data(N = 100, periods = 5, D = 2, D.probs = c(.8,.2), H = 2,
H.size = c(.1,.3), borders = c(100))

Sur_rel <- Survey.relevance(DATA, periods = 5, years = c(1,2),
domains.survey = list(1:2,1), ge.or.less = c("ge", "1"),
T0.or.EM = c("T0","EM"), size.class = c(0,5))

strat <- Stratify(DATA, Survey.relevant = Sur_rel, periods = 5,
years = c(1,2), T0.or.EM = c("T0","EM"), strata.T0 = c(500), strata.EM = c(10))
```

Survey.relevance	<i>Create an array indicating the survey relevance</i>
------------------	--

Description

The function creates an array of logical values indicating the survey relevance of each element in each time period.

Usage

```
Survey.relevance(
  DATA,
  periods,
  years,
  domains.survey,
  ge.or.less,
  T0.or.EM,
  size.class
)
```

Arguments

DATA	A data table. Can be generated with Gen.Data. It must include the variables ID, value_P1, value_corr_P1, and domain_P1. The latter three must further exist for every period which the surveys shall cover.
periods	An integer. Defines the number of periods which the surveys shall cover.
years	An integer vector. Defines the periodicity of all surveys. E.g. annual surveys would receive 1, biannual surveys 2, etc.
domains.survey	A list. Each list entry defines the domains which shall be covered by each survey.
ge.or.less	A vector of character strings. Defines whether elements greater or equal, or less than the thresholds given in size.class shall be marked as relevant for each survey. Only "ge" for greater or equal and "l" for less are allowed as values.
TO.or.EM	A vector of character strings. Defines whether elements shall be marked as relevant for each survey based on turnover values or number of employee values. Only "TO" for turnover and "EM" for the number of employees are allowed as values.
size.class	A numeric vector. Defines the numeric thresholds based on which elements shall be marked as relevant for each survey.

Value

An array of logical values indicating the survey relevance of each element. The dimensions cover the number of periods, the number of surveys, and the number of elements in DATA. It can be used directly as input for the survey coordination functions Dutch.BSC, French.BSC, Norwegian.BSC, and Swiss.BSC as well as for the function Stratify.

Examples

```
DATA <- Gen.Data(N = 100, periods = 5, D = 2, D.probs = c(.8,.2), H = 2,
H.size = c(.1,.3), borders = c(100))

Sur_rel <- Survey.relevance(DATA, periods = 5, years = c(1,2),
domains.survey = list(1:2,1), ge.or.less = c("ge", "l"),
TO.or.EM = c("TO","EM"), size.class = c(0,5))
```

Swiss.BSC

Applying the Modified Swiss Business Sample Coordination Algorithm

Description

The function applies the French business sample coordination algorithm as described by Qualité (2009) with some modifications. These modifications can be found in Straubinger(2025).

Usage

```

Swiss.BSC(
  DATA,
  period,
  sf,
  rf,
  years,
  pi.k.compute,
  survey.prio,
  n.min = 3,
  seed = 1,
  Survey.relevant,
  coord.info = NULL
)

```

Arguments

DATA	A data table. Can be generated with Gen.Data. It must include the variables ID, value_P1, value_corr_P1, and domain_P1. The latter three must further exist for every period which the surveys shall cover. The function returns the initial data set with some modifications as output which can be used as input for the next draw.
period	An integer. Defines the period in which the sample shall be drawn. Must be smaller or equal to the number of periods available in DATA, Survey.relevant, and in stratification.
sf	A numeric vector. Defines the sampling fraction for each survey.
rf	A numeric vector. Defines the rotation fraction for each survey in the current period.
years	An integer vector. Defines the periodicity of all surveys. E.g. annual surveys would receive 1, biannual surveys 2, etc.
pi.k.compute	Either a vector of character strings or a single character string. Defines which variable shall be used to calculate the inclusion probabilities of each survey. Only "value" and "value_corr" are allowed as entries. If only one entry is given, the function will automatically recycle it for all surveys.
survey.prio	An integer vector. Defines the order of surveys, i.e. surveys which shall be drawn first should be placed in front.
n.min	An integer. Defines the minimum sample size over all surveys. Default value is 3.
seed	An integer. Sets the seed for any random draws happening inside the function.
Survey.relevant	An array of logical values. Can be generated with Survey.relevance The dimensions cover the number of periods, the number of surveys, and the number of elements in DATA.
coord.info	A list of values needed for coordination. <code>length(coord.info) == nrow(DATA)</code> must hold. Each entry is a list of two, containing <code>[[1]]</code> the vector of interval

borders and `[[2]]` the matrix of inclusion indicators per sample for each interval. Can be used from previous survey periods coordinated with Swiss.BSC. Default value is NULL which results in no coordination with any of the previous periods.

Value

A list of three (DATA, in.sample, coord.info) is returned. DATA is the original data set given under DATA with some updates on response burden indicators. It can be used as input for further runs of the function. in.sample. is a matrix containing logical values. Its dimensions are `nrow(DATA)` and the number of surveys drawn. It shows for each elements and each survey whether it was drawn or not. coord.info contains values needed for further coordination. It can directly be used as input for another survey period, if desired.

References

Qualité, L. (2009): *Unequal probability sampling and repeated surveys* (PhD Thesis). Université de Neuchâtel, Switzerland.

Straubinger, J. (2025): *Methods to Coordinate Survey Samples in the Context of German Business Statistics* (PhD Thesis). Universität Trier, Germany.

See Also

`[‘data.table’][data.table::data.table]` for more information on data.table objects.

Examples

```
DATA <- Gen.Data(N = 100, periods = 5, D = 2, D.probs = c(.8,.2), H = 2,
H.size = c(.1,.3), borders = c(100))

Sur_rel <- Survey.relevance(DATA, periods = 5, years = c(1,2),
domains.survey = list(1:2,1), ge.or.less = c("ge", "1"),
T0.or.EM = c("T0","EM"), size.class = c(0,5))

strat <- Stratify(DATA, Survey.relevant = Sur_rel, periods = 5,
years = c(1,2), T0.or.EM = c("T0","EM"), strata.T0 = c(500), strata.EM = c(10))

S1 <- Swiss.BSC(DATA, period=1, sf=c(.2,.5), rf=c(.5,1), years=c(1,2),
pi.k.compute = "value_corr", survey.prio=1:2, n.min = 3,
Survey.relevant=Sur_rel, coord.info = NULL)

S2 <- Swiss.BSC(DATA = S1$DATA, period=2, sf=c(.2,.5), rf=c(.5,1),
years=c(1,2), pi.k.compute = "value_corr", survey.prio=1:2, n.min = 3,
Survey.relevant=Sur_rel, coord.info = S1$coord.info)
```

target.corr.sort

*Ordering Permutation to meet a desired Correlation***Description**

Rearrange the order of elements of *x* and *y* until a desired correlation between the two is (approximately) reached. Depending on the values of each element, a deviation from the targetted correlation is sometimes unavoidable.

Usage

```
target.corr.sort(
  x,
  y,
  target,
  tolerance = 0.01,
  seed = 1,
  direction = "lower",
  tries = 1000,
  ...
)
```

Arguments

<i>x</i>	A vector, must be numeric. <code>length(x) > 1</code> must hold.
<i>y</i>	A vector, must be numeric. <code>length(x) == length(y)</code> must hold.
<i>target</i>	A single, numeric value. Gives the desired correlation between <i>x</i> and <i>y</i> . <code>-1 <= target <= 1</code> must hold.
<i>tolerance</i>	A single, numeric value. <code>tolerance > 0</code> must hold. Specifies how large the deviation from the targetted correlation between <i>x</i> and <i>y</i> is allowed to be. Default value is 0.01, i.e. the algorithm stops when <code>target - tolerance <= cor(x,y) <= target + tolerance</code> .
<i>seed</i>	An integer. Sets the first seed for randomly reordering of <i>y</i> with <code>set.seed()</code> . Default is 1.
<i>direction</i>	A character string indicating from which direction the targetted correlation shall be approached. If "lower" (default) is given, the correlation between <i>x</i> and <i>y</i> is more likely to be a little smaller than the targetted correlation. If "upper" is given, it is more likely to be a little larger than the targetted correlation.
<i>tries</i>	An integer defining how often each step of the algorithm shall be repeated. A larger number of tries might lead to the correlation between <i>x</i> and <i>y</i> being closer to the targetted one, but with the cost of longer computation time. Default is 1000.
<i>...</i>	Further arguments passed to from other methods.

Details

The underlying algorithm shuffles the order of y values until a desired correlation is (approximately reached). To achieve this, the original order of y is shuffled by adding `rnorm(length(y),0,v)` to the original order and checking whether the correlation of the reordered y values and the x values is close enough (within the given tolerance) to the target correlation. This is done in two steps. First, the parameter v is increased until the targetted correlation is reached within 10 times the given tolerance, or until the maximum number of iterations (given by tries) for each v is reached. Then, the algorithm is repeated while trying to reach the targetted correlation within the given tolerance. In some cases, it is impossible to reach a certain correlation, so the algorithm will stop after a number of iterations (also determined by tries). Note, that 'tries' does not refer to the total number of iterations, but rather to the number of iterations for each tried value of v.

Value

A list of three (x,y,corr) is returned. x returns the original values of x, y the rearranged values of y, and corr the new correlation between x and y.

Examples

```
x <- 1:10
y <- sample(x,10)

target.corr.sort(x,y,.5)
target.corr.sort(x,y,.5, direction = "upper")
target.corr.sort(x,y,-.5)
target.corr.sort(x,y,.2, tolerance = .0001, tries = 100)
```

uneq.sample	<i>Draw a sample with unequal probabilities</i>
-------------	---

Description

Draw a sample with unequal probabilities, either with replacement, or without replacement. It is widely known that the `sample()` function in base R does not select a sample with the correct inclusion probabilities, if those are not equal (see Tillé, 2023)

Usage

```
uneq.sample(n, p.k, replacement = FALSE)
```

Arguments

n	An integer giving the desired sample size. $0 < n \leq \text{length}(p.k)$ must hold.
p.k	A numeric vector with the proportional values of the size variable (inclusion probabilities/n) of each element. $p.k \geq 0$ must hold for all p.k. $\text{sum}(p.k) > 0$ must hold. If $\text{sum}(p.k) \neq 1$, <code>uneq.sample</code> will automatically adjust the weights.
replacement	A logical value. In FALSE (default) <code>uneq.sample</code> will return a sample without replacement. If TRUE <code>uneq.sample</code> will return a sample with replacement.

Value

An integer vector showing the elements which were sampled.

References

Tillé Y. (2023): *Remarks on some misconceptions about unequal probability sampling without replacement*. In: Computer Science Review 47 (2023)

Examples

```
incl_probs <- c(.1,.1,.8)
uneq.sample(2,incl_probs)

# Automatic adjustment of inclusion probabilities if sum(p.k) != 1:
incl_probs <- c(.1,.1,.8,.5)
uneq.sample(2,incl_probs)

# A small Monte Carlo simulation:
vec <- vector("numeric")
incl_probs <- c(.2,.3,.5)
for(i in 1:100000){
  set.seed(i)
  vec[i] <- uneq.sample(1,incl_probs)
}
prop.table(table(vec))
```

Index

Dutch.BSC, [2](#)

French.BSC, [4](#)

Gen.Data, [7](#)

is.in.interval, [10](#)

nearest.neighbour, [10](#)

Norwegian.BSC, [11](#)

random.order, [13](#)

Stratify, [14](#)

Survey.relevance, [15](#)

Swiss.BSC, [16](#)

target.corr.sort, [19](#)

uneq.sample, [20](#)