

# Package ‘stratification’

February 15, 2012

**Type** Package

**Title** Univariate Stratification of Survey Populations

**Version** 2.0-3

**Date** 2011-02-02

**Author** Sophie Baillargeon <Sophie.Baillargeon@mat.ulaval.ca> and  
Louis-Paul Rivest <Louis-Paul.Rivest@mat.ulaval.ca>

**Maintainer** Sophie Baillargeon <Sophie.Baillargeon@mat.ulaval.ca>

**Description** This package performs univariate stratification of survey populations. The main function implements a generalization of the Lavallee-Hidiroglou method of stratum construction. The generalized method takes into account a discrepancy between the stratification variable and the survey variable. The determination of the optimal boundaries also incorporate, if desired, an anticipated non-response, a take-all stratum for large units, and a take-none stratum for small units. The well known cumulative root frequency rule of Dalenius and Hodges and the geometric rule of Gunning and Horgan are also implemented.

**License** GPL-2

**Repository** CRAN

**Date/Publication** 2011-02-03 08:19:39

## R topics documented:

stratification-package . . . . .	2
CochranHorganData . . . . .	6
MRTS . . . . .	8
SHS . . . . .	9
strata.bh . . . . .	10
strata.LH . . . . .	13

strata.rule . . . . .	22
strata.tool . . . . .	25
Sweden . . . . .	26
var.strata . . . . .	27

<b>Index</b>	<b>31</b>
--------------	-----------

---

stratification-package

*Collection of Functions for Univariate Stratification of Survey Populations*

---

## Description

This package contains various functions for univariate stratification of survey populations. The well known cumulative root frequency rule by Dalenius and Hodges (1959) and the geometric rule by Gunning and Horgan (2004) are implemented. However, the main function implements a generalized Lavallee-Hidiroglou (1988) method of strata construction. It can be used with Sethi's (1963) or Kozak's (2004) algorithm. The generalized method takes into account a discrepancy between the stratification variable  $X$  and the survey variable  $Y$ . The method can consider a loglinear model with mortality between the variables. When Kozak's algorithm is used, two additional models are available: a heteroscedastic linear model and a random replacement model as in Rivest (2002). The optimal boundaries determination also incorporates, if desired, an anticipated non-response, a take-all stratum for the large units and a take-none stratum for the small units.

## Details

Package: stratification  
 Type: Package  
 Version: 2.0-3  
 Date: 2011-02-02  
 License: GPL-2

## OVERVIEW OF THE FUNCTIONS

To determine the stratum sample sizes given a set of stratum boundaries: [strata.bh](#)

To determine the stratum boundaries first and, in a second step, the stratum sample sizes:

[strata.cumrootf](#): cumulative root frequency method by Dalenius and Hodges (1959)

[strata.geo](#): geometric method by Gunning and Horgan (2004)

To determine the optimal stratum boundaries and sample sizes in a single step:

[strata.LH](#): generalized Lavallee-Hidiroglou method with Sethi's (1963) or Kozak's (2004) algorithm

All these functions create objects of class "strata", which can be visualized with [print.strata](#) and [plot.strata](#). One can also apply, with the function [var.strata](#), a stratified design to a survey

variable  $Y$  different from the one used for the construction of the stratified design.

## INFORMATION RELATIVE TO MANY FUNCTIONS

The functions `strata.bh`, `strata.cumrootf`, `strata.geo` and `strata.LH` need to be given  $x$ , the values of the stratification variable  $X$ ,  $L_s$ , the desired number of sampled strata, `alloc`, an allocation rule, and a target sample size  $n$  or a target level of precision  $CV$  for the survey estimator. However, for Sethi's (1963) algorithm, only a target  $CV$  can be given. To reach a target  $n$  using the generalized Lavalée-Hidiroglou method, Kozak's (2004) algorithm has to be used with the `strata.LH` function.

Throughout the package, strata number 1 contains the smallest units and strata number  $L$  the biggest ones. So every vector of boundaries contains numbers in ascending order. The function `strata.bh` must be given boundaries `bh` fulfilling this condition. This remark also applies to the argument `initbh` of `strata.LH` used to give initial boundaries.

## DEFINITION OF STRATUM BOUNDARIES

Let's note  $b_0, b_1, \dots, b_L$  the stratum boundaries. Stratum  $h$  contains all the units with an  $X$ -value in the interval  $[b_{h-1}, b_h)$  for  $h = 1, \dots, L$  such that  $b_0 = \min(X)$  and  $b_L = \max(X) + 1$ , where  $\min(X)$  and  $\max(X)$  are respectively the minimum and the maximum values of the stratification variable. The argument `bh` of `strata.bh`, the argument `initbh` of `strata.LH` and the output value `bh` of any function of **stratification** with the prefix "strata" are length  $L - 1$  vectors of the boundaries  $b_1, b_2, \dots, b_{L-1}$ .

## MORE DETAILS ABOUT THE TAKE-NONE STRATUM

The  $L_s$  argument refers to the number of sampled strata. If the stratified design does not have a take-none stratum then  $L_s=L$  is the total number of strata, otherwise  $L_s=L - 1$ .

A non empty take-none stratum induces a bias in the estimator of the mean of  $Y$ , and the precision is measured by the relative root mean squared error (RRMSE), not by the coefficient of variation (CV). Regardless, in the functions the argument given to specify a target precision for the survey estimator is always named  $CV$ . However, in the output, the anticipated level of precision is named RRMSE for the functions accepting a `takenone` argument (`strata.bh` and `strata.LH`), and it is named  $CV$  for the other functions (`strata.cumrootf` and `strata.geo`).

When a `takenone` stratum is requested, one can specify a `bias.penalty` argument. We define the mean squared error for the estimator of the mean of  $Y$  by  $MSE = (bias.penalty \times bias)^2 + variance$ . It is sometimes possible to estimate the bias using the sum of the  $Y$  values in the take-none stratum from administrative data. In this situation, it might be appropriate to set `bias.penalty` to a value lower than 1. This will typically enlarge the take-none stratum. The value given to `bias.penalty` depends on the confidence level we have in the bias estimate. By default, it is assumed that no bias estimate is available and the whole bias contributes to the MSE (`bias.penalty=1`).

## CERTAIN STRATUM

It is possible to insure that some units are included in the sample with the argument `certain`. This argument is a vector containing the positions in the vector  $x$  of the units to be included with certainty in the sample. We say that these units form the certain stratum. They are excluded from the population prior to the determination of the stratum boundaries, but they are accounted for in the

calculation of the anticipated mean and the RRMSE. Essentially, these units form their own separate take-all stratum that is not subject to stratification. They do not have to be consecutive units according to the stratification variable, therefore their variance is meaningless. Non-response is not possible in the certain stratum. The functions return a value named `certain.info` containing the number of units in the certain stratum and their anticipated mean.

#### SPECIFICATION OF THE ALLOCATION RULE

The `alloc` argument must be a list containing the numerics `q1`, `q2` and `q3` which specify the allocation rule according to the general allocation scheme presented in Hidiroglou and Srinath (1993)

$$a_h = \frac{\gamma_h}{\sum \gamma_h} \quad \text{where} \quad \gamma_h = N_h^{2q_1} \bar{Y}_h^{2q_2} S_{yh}^{2q_3}.$$

For example, a proportional allocation is obtained when `q1=0.5` and `q2=q3=0`, a power allocation is obtained when `q1=q2=p/2` and `q3=0`, and a Neyman allocation (the default) is obtained when `q1=q3=0.5` and `q2=0`.

#### SPECIFICATION OF A MODEL BETWEEN $Y$ AND $X$

Every function can take into account a discrepancy between the stratification variable  $X$  and the survey variable  $Y$ . The functions `strata.bh`, `strata.cumrootf` and `strata.geo` perform allocation on the basis of anticipated moments whereas the `strata.LH` function goes further; it determines the optimal boundaries considering the anticipated moments. The following models for the relationship between  $Y$  and  $X$  can be specified through the `model` and `model.control` arguments:

- **loglinear model with mortality** (`model="loglinear"`):

$$Y = \begin{cases} \exp(\alpha + \text{beta} \log(X) + \text{epsilon}) & \text{with probability } p_h \\ 0 & \text{with probability } 1 - p_h \end{cases}$$

where  $\text{epsilon} \sim N(0, \text{sig}2)$  is independent of  $X$ . The parameter  $p_h$  is specified through `ph`, `ptakenone` and `pcertain` (elements of `model.control`). Note: The  $\alpha$  parameter does not have to be specified because  $\exp(\alpha)$  is a multiplicative factor that has no impact on the outcome.

- **heteroscedastic linear model** (`model="linear"`):

$$Y = \text{beta}X + \text{epsilon}$$

where  $\text{epsilon} \sim N(0, \text{sig}2 X^\gamma)$ .

- **random replacement model** (`model="random"`):

$$Y = \begin{cases} X & \text{with probability } 1 - \text{epsilon} \\ X_{new} & \text{with probability } \text{epsilon} \end{cases}$$

where  $X_{new}$  is a random variable independent of  $X$  having the same distribution than  $X$ .

The `model.control` argument is a list that can supply any of the following model parameter:

`beta` A numeric: the slope of the "loglinear" or "linear" model. The default is 1.

- `sig2` A numeric: the variance parameter of the "loglinear" or "linear" model. The default is 0.
- `ph` A vector giving the survival rate in each of the  $L_s$  sampled strata for the "loglinear" model. A single number can be given if the rate doesn't vary among strata. The default is 1 in each stratum.
- `ptakenone` A numeric: the survival rate in the take-none stratum, if a take-none stratum is added to the stratified design. The default is 1.
- `pcertain` A numeric: the survival rate in the certain stratum, if a certain stratum is added to the stratified design. The default is 1.
- `gamma` A numeric: the exponent of  $X$  in the residual variance of the "linear" model. The default is 0.
- `epsilon` A numeric: the probability that the  $Y$ -value for a unit is equal to the  $X$ -value for a randomly selected unit in the population. It concerns the "random" model only. The default is 0.

#### ADJUSTMENT FOR A TAKE-ALL STRATUM

If, at the end of the computations, the solution reached by any of these functions contains at least one take-some stratum with  $n_h > N_h$ , the computations are carried out again setting the take-some stratum with the largest units as a take-all stratum. This is done until  $n_h \leq N_h$  for all the take-some strata. This adjustment is optional for the function `strata.bh` (argument `takeall.adjust`).

Note: In special circumstances, the algorithm might result in more than one take-all stratum. If non non-response is taken into account in the take-all strata, we can see them as forming one big take-all stratum. However, if non-response occurs in those strata, their boundaries influence the value of the optimization criteria ( $n$  or  $CV$ ). In that case, we cannot see them as forming one big take-all stratum.

#### ROUNDING of the $n_h$ 's

These functions return the sample sizes `nh`, which are integers, but also the non-integer values of these sample sizes, named `nh.nonint`. The vector `nh.nonint` is the result of the application of the allocation rule. The `nh.nonint` values are rounded to get the integer sample sizes `nh`. Here is how the rounding is done. If a target  $CV$  is requested, the values are simply rounded to the largest integer. However, if a target  $n$  is requested, the rounding is a little more complicated because the `nh` should sum to the target  $n$  and we do not want positive `nh` inferior to 1 to be rounded to zero. Therefore, we first round to 1 the positive `nh` inferior to 1. Then we calculate how many values (say `nup`) must be rounded to the largest integer and how many must be rounded to the smallest integer in order to fulfill the condition  $\sum(nh)=n$ . We choose the `nup` values with the largest decimal part for the ceiling rounding, the other `nh` are rounded down.

#### Author(s)

Sophie Baillargeon <Sophie.Baillargeon@mat.ulaval.ca> and  
Louis-Paul Rivest <Louis-Paul.Rivest@mat.ulaval.ca>

#### References

Baillargeon, S. and Rivest, L.-P. (2010). Univariate stratification in R with the package stratification. Paper in progress.

- Baillargeon, S., Rivest, L.-P., Ferland, M. (2007). Stratification en enquetes entreprises : Une revue et quelques avancees. *Proceedings of the Survey Methods Section, 2007 SSC Annual Meeting*. [http://www.ssc.ca/survey/documents/SSC2007\\_S\\_Baillargeon.pdf](http://www.ssc.ca/survey/documents/SSC2007_S_Baillargeon.pdf)
- Dalenius, T. and Hodges, J.L., Jr. (1959). Minimum variance stratification. *Journal of the American Statistical Association*, **54**, 88-101.
- Gunning, P. and Horgan, J.M. (2004). A new algorithm for the construction of stratum boundaries in skewed populations. *Survey Methodology*, **30**(2), 159-166.
- Hidiroglou, M.A. and Srinath, K.P. (1993). Problems associated with designing subannual business surveys. *Journal of Business & Economic Statistics*, **11**, 397-405.
- Kozak, M. (2004). Optimal stratification using random search method in agricultural surveys. *Statistics in Transition*, **6**(5), 797-806.
- Lavallee, P. and Hidiroglou, M.A. (1988). On the stratification of skewed populations. *Survey Methodology*, **14**, 33-43.
- Rivest, L.-P. (2002). A generalization of the Lavallee and Hidiroglou algorithm for stratification in business surveys. *Survey Methodology*, **28**(2), 191-198.
- Rivest, L.-P. and Baillargeon, S. (2009). A general algorithm for univariate stratification. *International Stratification Review*, **77**(3), 331-344.
- Sethi, V. K. (1963). A note on optimum stratification of populations for estimating the population means. *The Australian Journal of Statistics*, **5**, 20-33.

---

CochranHorganData

*Populations Analyzed in Gunning and Horgan (2004) and Cochran (1961)*

---

## Description

The first population Debtors is an accounting population of debtors in an Irish firm, detailed in Horgan (2003). The other populations are three of the skewed populations in Cochran (1961).

These are:

UScities: the population in thousands of US cities in 1940;

UScolleges: the number of students in four-year US colleges in 1952-1953;

USbanks: the resources in millions of dollars of large commercial US banks.

## Usage

```
data(Debtors)
data(UScities)
data(UScolleges)
data(USbanks)
```

**Format**

The formats of these data sets are, respectively:

```
num [1:3369] 40 40 40 40 40 40 40 40 40 40 ...
num [1:1038] 10 10 10 10 10 10 10 10 10 10 ...
num [1:677] 200 201 202 202 207 210 211 213 215 217 ...
num [1:357] 70 71 72 72 72 73 73 73 73 ...
```

**Source**

Jane M. Horgan

**References**

Cochran, W.G. (1961). Comparison of methods for determining stratum boundaries. *Bulletin of the International Statistical Institute*, **32**(2), 345-358.

Gunning, P. and Horgan, J.M. (2004). A new algorithm for the construction of stratum boundaries in skewed populations. *Survey Methodology*, **30**(2), 159-166.

Horgan, J.M. (2003). A list sequential sampling scheme with applications in financial auditing. *IMA Journal of Management Mathematics*, **14**, 1-18.

**Examples**

```
### Reproduction of the results in Table 4 and Table 7 part 3 (case L=5) of
### Gunning and Horgan (2004). The differences in the nh come from different
### rounding. The more important differences observed for the cumulative
### root frequency method are due to the use of different numbers of classes.
data(Debtors)
strata.geo(x=Debtors,n=100,Ls=5,alloc=c(0.5,0,0.5))
strata.cumrootf(x=Debtors,nclass=40,n=100,Ls=5,alloc=c(0.5,0,0.5))
strata.LH(x=Debtors,CV=0.0360,Ls=5,alloc=c(0.5,0,0.5),takeall=1,algo="Sethi")

data(UScities)
strata.geo(x=UScities,n=100,Ls=5,alloc=c(0.5,0,0.5))
strata.cumrootf(x=UScities,nclass=40,n=100,Ls=5,alloc=c(0.5,0,0.5))
strata.LH(x=UScities,CV=0.0144,Ls=5,alloc=c(0.5,0,0.5),takeall=1,algo="Sethi")

data(UScolleges)
strata.geo(x=UScolleges,n=100,Ls=5,alloc=c(0.5,0,0.5))
strata.cumrootf(x=UScolleges,nclass=40,n=100,Ls=5,alloc=c(0.5,0,0.5))
strata.LH(x=UScolleges,CV=0.0184,Ls=5,alloc=c(0.5,0,0.5),takeall=1,algo="Sethi")

data(USbanks)
strata.geo(x=USbanks,n=100,Ls=5,alloc=c(0.5,0,0.5))
strata.cumrootf(x=USbanks,nclass=40,n=100,Ls=5,alloc=c(0.5,0,0.5))
strata.LH(x=USbanks,CV=0.0110,Ls=5,alloc=c(0.5,0,0.5),takeall=1,algo="Sethi")
```

MRTS

*Simulated Data from the Monthly Retail Trade Survey (MRTS) of  
Statistics Canada*

## Description

This data set is a vector containing the simulated values of a realistic stratification variable: the size measure used for Canadian retailers in the Monthly Retail Trade Survey (MRTS) carried out by Statistics Canada. This size measure is created using a combination of independent survey data and three administrative variables from the corporation tax return. The MRTS aims at estimating sales from retailers, which are a key monthly indicator of consumer purchasing patterns in Canada.

## Usage

```
data(MRTS)
```

## Format

The format is: num [1:2000] 141 209 238 257 261 ...

## Source

The data set has been simulated with the command:

```
exp(rst(n=2000, location=9.7093354, scale=0.7885551,
shape=-0.6384867, df=5.6243544)).
```

The function `rst` comes from the package `sn`. It generates random numbers for the skew-t distribution. The parameters of the distribution have been estimated with the assistance of Michel Ferland from Statistics Canada to be representative of the measure of size used in the MRTS.

## References

Baillargeon, S., Rivest, L.-P., Ferland, M. (2007). Stratification en enquetes entreprises : Une revue et quelques avancees. *Proceedings of the Survey Methods Section, 2007 SSC Annual Meeting*. [http://www.ssc.ca/survey/documents/SSC2007\\_S\\_Baillargeon.pdf](http://www.ssc.ca/survey/documents/SSC2007_S_Baillargeon.pdf)

Information on Statistics Canada Monthly Retail Trade Survey: <http://www.statcan.ca/cgi-bin/imdb/p2SV.pl?Function=getSurvey&SDDS=2406&lang=en&db=IMDB&dbg=f&adm=8&dis=2>

## Examples

```
# Production of results similar to those in Table 1 of Baillargeon, Rivest
# and Ferland (2007). The results are not the same because calculations in
# the paper were conducted on real data whereas, for confidentiality reason,
# the MRTS data included in the package is simulated.
data(MRTS)
geo<-strata.geo(x=MRTS,CV=0.01,Ls=4,alloc=c(0.5,0,0.5))
geo
aRRMSE.geo<-var.strata(geo,model="loglinear",
model.control=list(beta=0.9,sig2=0.015,ph=c(0.8,0.9,0.95,1)))
```

```

aRRMSE.geo$RRMSE
plot(geo,logscale=TRUE)
# The geometric method does not perform well because of some small units

cumrootf<-strata.cumrootf(x=MRTS,nclass=500,CV=0.01,Ls=4,alloc=c(0.5,0,0.5))
cumrootf
aRRMSE.cum<-var.strata(cumrootf,,rh=c(0.85,0.9,0.9,1),model="loglinear",
                      model.control=list(beta=0.9,sig2=0.015,ph=c(0.8,0.9,0.95,1)))
aRRMSE.cum$RRMSE

LH<-strata.LH(x=MRTS,CV=0.01,Ls=4,alloc=c(0.5,0,0.5),takeall=1,algo="Sethi")
LH
aRRMSE.LH<-var.strata(LH,rh=c(0.85,0.9,0.9,1),model="loglinear",
                      model.control=list(beta=0.9,sig2=0.015,ph=c(0.8,0.9,0.95,1)))
aRRMSE.LH$RRMSE

LH.full<-strata.LH(x=MRTS,CV=0.01,Ls=4,alloc=c(0.5,0,0.5),takeall=1,
                  algo="Sethi",rh=c(0.85,0.9,0.9,1),model="loglinear",
                  model.control=list(beta=0.9,sig2=0.015,ph=c(0.8,0.9,0.95,1)))
LH.full
aRRMSE.LH.full<-var.strata(LH.full,rh=c(0.85,0.9,0.9,1),model="loglinear",
                           model.control=list(beta=0.9,sig2=0.015,ph=c(0.8,0.9,0.95,1)))
aRRMSE.LH.full$RRMSE

```

---

SHS

*Data from the 2001 Survey of Household Spending (SHS), Statistics  
Canada*

---

### Description

This data set contains some variables from the 2001 Survey of Household Spending (SHS) carried out by Statistics Canada. The main purpose of this survey is to obtain detailed information about household spending during the reference year.

### Usage

```
data(SHS)
```

### Format

A data frame with 16057 observations on the following 7 variables.

CASEID Identification number  
WEIGHT Weight at household level  
PROVINCP Province or territory code  
URBRUR Urban rural code  
URBSIZEP Size of area of residence code  
HHINCTOT Household income before taxes  
M101 Household spending on recreation

## Details

In this package, HHINCTOT is used as a stratification variable and M101 as a survey variable.

## Source

Income Statistics Division, Statistics Canada.

## References

Information on Statistics Canada Survey of Household Spending: <http://www.statcan.ca/cgi-bin/imdb/p2SV.pl?Function=getSurvey&SDDS=3508&lang=en&db=IMDB&dbg=f&adm=8&dis=2>

## Examples

```
data(SHS)

# Estimation of the response rate
X<-SHS$HHINCTOT[SHS$HHINCTOT>0]
Y<-SHS$M101[SHS$HHINCTOT>0]
Y[Y<0]<-0
p<-sum(Y>0)/length(Y)

# Study of the relationship between X and Y for the active units
Xactive<-SHS$HHINCTOT[SHS$HHINCTOT>0&SHS$M101>0]
Yactive<-SHS$M101[SHS$HHINCTOT>0&SHS$M101>0]
plot(log(Xactive),log(Yactive))
# Extreme values are omitted for a more robust estimation
keep<-Xactive/Yactive>quantile(Xactive/Yactive,0.01)&
  Xactive/Yactive<quantile(Xactive/Yactive,0.99)
plot(log(Xactive)[keep],log(Yactive)[keep])
reg<-lm( log(Yactive)[keep]~log(Xactive)[keep] )
summary(reg)

# Stratification assuming X=Y
nomodel<-strata.LH(x=X,CV=0.05,Ls=3,alloc=c(0.5,0,0.5),takeall=0,model="none")
nomodel
var.strata(nomodel,y=Y) # The target CV is far from being reached

# Stratification taking into account a loglinear model with mortality
# between X and Y, using the estimated parameters values
model<-strata.LH(x=X,CV=0.05,Ls=3,alloc=c(0.5,0,0.5),takeall=0,model="loglinear",
  model.control=list(beta=reg$coef[2],sig2=summary(reg)$sigma^2,ph=0.97))
model
var.strata(model,y=Y) # The target CV is reached
```

## Description

The function `strata.bh` stratifies a population given a set of boundaries. It calculates the stratum sample sizes and the anticipated coefficient of variation or relative root mean squared error.

## Usage

```
strata.bh(x, bh, n = NULL, CV = NULL, Ls = 3, certain = NULL,
         alloc = list(q1 = 0.5, q2 = 0, q3 = 0.5), takenone = 0,
         bias.penalty = 1, takeall = 0, takeall.adjust = FALSE,
         rh = rep(1, Ls), model = c("none", "loglinear", "linear",
         "random"), model.control = list())
```

## Arguments

<code>x</code>	A vector containing the values of the stratification variable $X$ for every unit in the population.
<code>bh</code>	A vector of the $L - 1$ stratum boundaries where $L$ is the total number of strata. Therefore, if <code>takenone=0</code> then $L=Ls$ , and if <code>takenone=1</code> then $L=Ls+1$ .
<code>n</code>	A numeric: the target sample size. It has no default value. The argument <code>n</code> or the argument <code>CV</code> must be input.
<code>CV</code>	A numeric: the target coefficient of variation or relative root mean squared error if <code>takenone=1</code> . It has no default value. The argument <code>CV</code> or the argument <code>n</code> must be input.
<code>Ls</code>	A numeric: the number of sampled strata (excluding the take-none stratum). The default is 3.
<code>certain</code>	A vector giving the position, in the vector <code>x</code> , of the units that must be included in the sample (see <a href="#">stratification-package</a> ). By default <code>certain</code> is <code>NULL</code> , which means that no units are a priori chosen to be in the sample.
<code>alloc</code>	A list specifying the allocation scheme. The list must contain 3 numerics for the 3 exponents <code>q1</code> , <code>q2</code> and <code>q3</code> in the general allocation scheme (see <a href="#">stratification-package</a> ). The default is Neyman allocation ( <code>q1=q3=0.5</code> and <code>q2=0</code> )
<code>takenone</code>	A numeric: the number of take-none strata (0 or 1). The default is 0, i.e. no take-none stratum is included.
<code>bias.penalty</code>	A numeric between 0 and 1 giving the penalty for the bias in the anticipated mean squared error (MSE) of the survey estimator (see <a href="#">stratification-package</a> ). This argument is relevant only if <code>takenone=1</code> . The default is 1.
<code>takeall</code>	A numeric: the number of take-all strata. The default is 0, i.e. no take-all stratum is included.
<code>takeall.adjust</code>	A logical. If <code>TRUE</code> , when $n_h > N_h$ for a take-some stratum, the <code>takeall</code> argument is increased by one and the stratification is carried out again. This is done as long as $n_h \leq N_h$ for every take-some stratum. The default is <code>FALSE</code> , i.e. no adjustment is made. Note: in other functions of the package <b>stratification</b> , this adjustment is not optional; it is made automatically (see <a href="#">stratification-package</a> ).

rh	A vector giving the anticipated response rates in each of the $L$ s sampled strata. A single number can be given if the rates do not vary among strata. The default is 1 in each stratum.
model	A character string identifying the model used to describe the discrepancy between the stratification variable $X$ and the survey variable $Y$ . It can be "none" if one assumes $Y = X$ , "loglinear" for the loglinear model with mortality, "linear" for the heteroscedastic linear model or "random" for the random replacement model (see <a href="#">stratification-package</a> for a description of these models). The default is "none".
model.control	A list of model parameters (see <a href="#">stratification-package</a> ). The default values of the parameters correspond to the model $Y = X$ .

### Value

Nh	A vector of length $L$ containing the population sizes $N_h$ , i.e. the number of units in each stratum.
nh	A vector of length $L$ containing the sample sizes $n_h$ , i.e. the number of units to sample in each stratum. See <a href="#">stratification-package</a> for information about the rounding used to get these values.
n	The total sample size ( $\text{sum}(\text{nh})$ ).
nh.nonint	A vector of length $L$ containing the non-integer values of the sample sizes $n_h$ .
certain.info	A vector giving statistics for the certain stratum (see <a href="#">stratification-package</a> ). It contains $N_c$ , the number of units chosen a priori to be in the sample, and $\text{meanc}$ , the anticipated mean of $Y$ for these units.
opti.criteria	The final value of the criteria to optimize : either the non-integer value of the total sample size (if a target CV was given) or the RRMSE associated to non-integer stratum sample sizes (if a target $n$ was given).
meanh	A vector of length $L$ containing the anticipated means of $Y$ in each stratum.
varh	A vector of length $L$ containing the anticipated variances of $Y$ in each stratum.
mean	A numeric: the anticipated global mean value of $Y$ .
RMSE	A numeric: the root mean squared error (or standard error if $\text{takenone}=0$ ) of the anticipated global mean of $Y$ . This is defined as the squared root of: $(\text{bias.penalty} \times \text{bias of the mean})^2 + \text{variance of the mean}$ .
RRMSE	A numeric: the anticipated relative root mean squared error (or coefficient of variation if $\text{takenone}=0$ ) for the mean of $Y$ , i.e. $\text{RMSE}$ divided by $\text{mean}$ .
relativebias	A numeric: the anticipated relative bias of the estimator, i.e. $(\text{bias.penalty} \times \text{bias of the mean})$ divided by $\text{mean}$ . If $\text{takenone}=0$ , this numeric is zero.
propbiasMSE	A numeric: the proportion of the MSE attributable to the bias of the estimator, i.e. $(\text{bias.penalty} \times \text{bias of the mean})^2$ divided by the MSE of the mean. If $\text{takenone}=0$ , this numeric is zero.
stratumID	A factor identifying, for each observation in the input $X$ , the stratum to which it has been assigned. The value "certain" is given to units a priori chosen to be in the sample.

takeall	The number of take-all strata in the final solution. Note: It is possible that $n_h = N_h$ for non take-all strata because the condition for an automatic addition of a take-all stratum is $n_h > N_h$ .
call	The function call (object of class "call").
date	A character string that contains the system date and time when the function ended.
args	A list of all the arguments input to the function or used by default.

**Author(s)**

Sophie Baillargeon <Sophie.Baillargeon@mat.ulaval.ca> and  
Louis-Paul Rivest <Louis-Paul.Rivest@mat.ulaval.ca>

**See Also**

[print.strata](#), [plot.strata](#), [strata.cumrootf](#), [strata.geo](#), [strata.LH](#)

**Examples**

```
data(USbanks)
adjust<-strata.geo(x=USbanks,CV=0.01,Ls=4,alloc=c(0.35,0.35,0))
adjust
adjust$nh.nonint
noadjust<-strata.bh(x=USbanks,bh=adjust$bh,CV=0.01,Ls=4,
  alloc=c(0.35,0.35,0),takeall=0,takeall.adjust=FALSE)
noadjust
noadjust$nh.nonint
# without the adjustment for a take-all stratum, n is smaller than
# with the adjustment, but the target CV is not reached.
```

---

strata.LH

---

*Generalized Lavallee-Hidiroglou Method of Strata Construction*


---

**Description**

This function constructs optimal strata with a generalized Lavallee-Hidiroglou (1998) method. The function uses Kozak's (2004) algorithm by default, but it can also apply Sethi's (1963) algorithm (argument `algo="Sethi"`). The function takes into account a discrepancy between the stratification variable  $X$  and the survey variable  $Y$ . It can consider a loglinear model with mortality between the variables. With Kozak's algorithm, two additional models are implemented: an heteroscedastic linear model and a random replacement model as in Rivest (2002). The determination of the optimal boundaries also incorporates, if desired, an anticipated non-response, a take-all stratum for the large units and a take-none stratum for the small units.

Sethi's algorithm is not used by default because it can be numerically unstable, especially with a take-none stratum. Better results were obtained with Kozak's algorithm in our numerical experiments.

**Usage**

```
strata.LH(x, initbh, n = NULL, CV = NULL, Ls = 3, certain = NULL,
  alloc = list(q1 = 0.5, q2 = 0, q3 = 0.5), takenone = 0,
  bias.penalty = 1, takeall = 0, rh = rep(1, Ls),
  model = c("none", "loglinear", "linear", "random"),
  model.control = list(), algo = c("Kozak", "Sethi"),
  algo.control = list())
```

**Arguments**

x	A vector containing the values of the stratification variable $X$ for every unit in the population.
initbh	A vector of $Ls-1$ , or $Ls$ if <code>takenone=1</code> , initial stratum boundaries (see <b>Details</b> ).
n	A numeric: the target sample size. It has no default value. The argument <code>n</code> or the argument <code>CV</code> must be input.
CV	A numeric: the target coefficient of variation or relative root mean squared error if <code>takenone=1</code> . It has no default value. The argument <code>CV</code> or the argument <code>n</code> must be input.
Ls	A numeric: the number of sampled strata (excluding the take-none stratum). The default is 3.
certain	A vector giving the position, in the vector <code>x</code> , of the units that must be included in the sample (see <a href="#">stratification-package</a> ). By default <code>certain</code> is <code>NULL</code> , which means that no units are chosen a priori to be in the sample.
alloc	A list specifying the allocation scheme. The list must contain 3 numerics for the 3 exponents <code>q1</code> , <code>q2</code> and <code>q3</code> in the general allocation scheme (see <a href="#">stratification-package</a> ). The default is Neyman allocation ( <code>q1=q3=0.5</code> and <code>q2=0</code> )
takenone	A numeric: the number of take-none strata (0 or 1). The default is 0, i.e. no take-none stratum is included.
bias.penalty	A numeric between 0 and 1 giving the penalty for the bias in the anticipated mean squared error (MSE) of the survey estimator (see <a href="#">stratification-package</a> ). This argument is relevant only if <code>takenone=1</code> . The default is 1.
takeall	A numeric: the number of take-all strata. The default is 0, i.e. no take-all stratum is included.
rh	A vector giving the anticipated response rates in each of the $Ls$ sampled strata. A single number can be given if the rates do not vary among strata. The default is 1 in each stratum.
model	A character string identifying the model used to describe the discrepancy between the stratification variable $X$ and the survey variable $Y$ . It can be "none" if one assumes $Y = X$ , "loglinear" for the loglinear model with mortality, "linear" for the heteroscedastic linear model or "random" for the random replacement model (see <a href="#">stratification-package</a> for a description of these models). The default is "none", so the original Lavallee-Hidiroglou (1998) method of strata construction is used. The last two models "linear" and "random" are only available with Kozak's algorithm.

model.control	A list of model parameters (see <a href="#">stratification-package</a> ). The default values of the parameters correspond to the model $Y = X$ .
algo	A character string identifying which optimization algorithm is to be used. It can be "Kozak" for Kozak's (2004) algorithm or "Sethi" for Sethi's (1963) algorithm. The default is "Kozak".
algo.control	A list of parameters to control the optimization algorithm (see <b>Details</b> ).

## Details

### INITIAL STRATUM BOUNDARIES

The default value of `initbh` is the set of  $Ls-1$  equidistant points along the range of the  $X$ -values (arithmetic starting points of Gunning and Horgan (2007)). Let's define  $d = (max(X) - min(X)) / Ls$ , the default initial boundaries are  $b_h = min(X) + h \times d$  for  $h = 1, \dots, Ls - 1$ .

If `takenone=1` and `initbh` is of size  $Ls-1$ , the initial boundary of the take-none stratum is set to the first percentile of  $x$ . If this first percentile is equal to the minimum value of  $x$ , this initial boundary would lead to an empty take-none stratum. In that case, the initial boundary of the take-none stratum is rather set to the second smallest value of  $x$ .

Sometimes, the specified initial boundaries are not suitable for Kozak's algorithm. As written below, this algorithm verifies at each iteration that the sampled strata contains at least `minNh` units and that they have positive sample sizes  $n_h$ . If the initial boundaries do not meet these conditions, the algorithm might not be able to change the boundaries at all because any change would be rejected for failing to meet the conditions. Therefore, in such a situation, the specified initial boundaries are replaced by robust boundaries. The robust boundaries give an empty take-none stratum if such a stratum is requested, take-all strata as small as possible, and take-some strata with approximately the same number of unique  $X$ -values. The sampled strata given by the robust boundaries contain at least `minNh` units. These robust boundaries give, in most situation, a non-negative optimization criteria  $n$  if a target CV is requested (see Rivest and Baillargeon 2009 for a discussion of possible negative  $n$  values).

### ALGORITHMS

**Sethi:** The formulas implemented for Sethi's algorithm are presented in Baillargeon, Rivest and Ferland (2007).

**Original Kozak:** The original Kozak's algorithm is described in Kozak (2004). At each iteration, it chooses a stratum boundary at random and a random modification for this boundary among the  $2 * \text{maxstep}$  possible alternatives. If this modification reduces the optimization criteria, creates sampled strata containing at least `minNh` units and leads to positive  $n_h$ , it is accepted. Otherwise, the boundaries are not changed. If the boundaries remain unchanged for `maxstill` consecutive iterations, the algorithm stops. This algorithm is run `rep` times. Note: Kozak's (2004) used the condition  $n_h \geq 2$  instead of  $n_h > 0$ . We chose to modify this requirement because we noticed that more severe conditions sometimes prevented the algorithm from selecting a path leading to the optimal solution. According to our numerical experiments, if one sets a posteriori to 2 the  $n_h$ 's that are equal to 1 at the end of the algorithm, the new sample size is smaller than or equal to the one obtained with the condition  $n_h \geq 2$ .

**Modified Kozak:** The modified Kozak's algorithm is a non-random version of Kozak's algorithm. At each iteration of the algorithm, the  $(L - 1) * 2 * \text{maxstep}$  possible boundaries modifications among which the original Kozak's algorithm chooses at random are tried and the modification giving the largest decrease in the optimization criteria is kept. The modification must also create sampled

strata containing at least  $\text{minNh}$  units and it must lead to positive  $n_h$ . This algorithm is slower than the original Kozak's algorithm.

For data sets with a small number of unique  $X$ -values  $Nu$ , Kozak's algorithm is often caught in a local minimum. In such situations, a complete enumeration of all possible sets of boundaries is feasible. When  $\text{takenone}=0$ , there are  $\text{choose}(Nu-1, L-1)$  possible sets of boundaries. The function `strata.LH` carries out a complete enumeration of all possible cases if the number of possible sets of boundaries is lower than the `algo.control` argument `minsol`.

#### ALGORITHM PARAMETERS

The `algo.control` argument is a list to supply any of the following parameters which control the algorithm. Sethi's algorithm only uses the first argument `maxiter`.

`maxiter` A numeric: the maximal number of iterations. The default is 500 for Sethi's algorithm, 10 000 for the original Kozak's algorithm and 3000 for the modified Kozak's algorithm.

`method` A character string identifying which Kozak's algorithm is to be used. It can be "original" for the original Kozak's algorithm or "modified" for the modified Kozak's algorithm. The default is "original".

`minNh` A numeric for Kozak's algorithms only: the minimum number of units required in each sampled stratum (no restriction is put on the take-none stratum, if included). `minNh` must be greater or equal to 2, which is the default.

`maxstep` A numeric for Kozak's algorithms only: the maximal step. The default is 3.

`maxstill` A numeric for the original Kozak's algorithm only: the maximal number of iterations without a change in the boundaries. The default is 100.

`rep` A numeric for Kozak's original algorithm only: the number of repetitions of the algorithm. The default is 3. This argument can be given the special value "change". If `rep="change"` the algorithm is repeated 3 times for 9 combinations of initial boundaries and `maxstep` values. The initial boundaries used are the cumulative root frequency boundaries, the geometric boundaries or the default initial boundaries. The `maxstep` values used are 3,  $\min(10, \text{number of unique } X\text{-values})$  or  $\min(\text{ceiling}(\text{number of unique } X\text{-values}/10), 100)$ . The `maxstill` argument is given a value proportional to `maxstep`. The option `rep="change"` can be long to run, but it increases the chances for the algorithm to reach the global minimum.

`minsol` A numeric for Kozak's algorithms only: the minimum number of possible solutions below which the algorithm is not run. If the number of possible sets of boundaries is lower than `minsol`, every set of boundaries is tried. This ensures that the global minimum is reached. The default value of `minsol` is 1000. This parameter has to take a value between 100 and 1 000 000.

#### CONVERGENCE

It is possible for the algorithm not to converge. In this case, a warning is printed. The only possible cause of non-convergence for Kozak's algorithm, original or modified, is to reach the maximum number of iterations before the stopping rule has been met. For the algorithm to converge, the argument `maxiter` has to be increased. On the other hand, non-convergence of Sethi's algorithm has several possible causes:

- a division by zero caused by an empty stratum can occur ;

- a division by zero caused by a 0 stratum variance can occur ;
- the square root of a negative number (negative discriminant) can occur ;
- the maximum number of iterations can be reached (often because the algorithm is caught in a loop of non-optimal sets of boundaries).

If a non-convergence happens, the user can try to change the initial boundaries or the model parameters. The user can also choose to work with Kozak's algorithm which should converge given an appropriate maximum number of iterations.

Let's note however that even if the algorithm converges, it is not guaranteed that it has reached a global minimum. Local minimum can occur with both Sethi's and Kozak's algorithms (Rivest and Baillargeon, 2009).

### Value

Nh	A vector of length $L$ containing the population sizes $N_h$ , i.e. the number of units in each stratum.
nh	A vector of length $L$ containing the sample sizes $n_h$ , i.e. the number of units to sample in each stratum. See <a href="#">stratification-package</a> for information about the rounding used to get these values.
n	The total sample size (sum(nh)).
nh.nonint	A vector of length $L$ containing the non-integer values of the sample sizes $n_h$ .
certain.info	A vector giving statistics for the certain stratum (see <a href="#">stratification-package</a> ). It contains Nc, the number of units chosen a priori to be in the sample, and meanc, the anticipated mean of $Y$ for these units.
opti.criteria	The final value of the criteria to optimize : either the non-integer value of the total sample size (if a target CV was given) or the RRMSE associated to non-integer stratum sample sizes (if a target n was given).
bh	A vector of the $L - 1$ optimal stratum boundaries found by the algorithm.
meanh	A vector of length $L$ containing the anticipated means of $Y$ in each stratum.
varh	A vector of length $L$ containing the anticipated variances of $Y$ in each stratum.
mean	A numeric: the anticipated global mean value of $Y$ .
RMSE	A numeric: the root mean squared error (or standard error if takenone=0) of the anticipated global mean of $Y$ . This is defined as the squared root of: (bias.penalty x bias of the mean) <sup>2</sup> + variance of the mean.
RRMSE	A numeric: the anticipated relative root mean squared error (or coefficient of variation if takenone=0) for the mean of $Y$ , i.e. RMSE divided by mean.
relativebias	A numeric: the anticipated relative bias of the estimator, i.e. (bias.penalty x bias of the mean) divided by mean. If takenone=0, this numeric is zero.
propbiasMSE	A numeric: the proportion of the MSE attributable to the bias of the estimator, i.e. (bias.penalty x bias of the mean) <sup>2</sup> divided by the MSE of the mean. If takenone=0, this numeric is zero.
stratumID	A factor identifying, for each observation in the input $X$ , the stratum to which it has been assigned. The value "certain" is given to units a priori chosen to be in the sample.

takeall	The number of take-all strata in the final solution. Note: It is possible that $n_h = N_h$ for non take-all strata because the condition for an automatic addition of a take-all stratum is $n_h > N_h$ .
call	The function call (object of class "call").
date	A character string that contains the system date and time when the function ended.
args	A list of all the arguments input to the function or used by default.
initbh	Only if an iterative algorithm is run: a vector of the $L - 1$ initial stratum boundaries. Most of the time, this will be a copy of the input initial boundaries, but sometimes these input boundaries have to be replaced by robust boundaries (see <b>Details</b> ).
iter.detail	Only if an iterative algorithm is run. A matrix giving the detail of every iteration of the algorithm. Because of the automatic adjustment for a take-all stratum, the algorithm can be run more than once by adding 1 to the takeall argument each time. All the runs of the algorithm are included in iter.detail. The matrix iter.detail contains, for each iteration, the stratum boundaries, the step for Kozak's algorithm, the value of the criteria to optimize, the iteration identification number and the run identification number. For the original Kozak's algorithm, the repetition number is put in iter.detail last column. Also, for this algorithm, a row is added to iter.detail only for iterations that decrease the optimization criteria.
niter	Only if an iterative algorithm is run: the total number of iterations in the algorithm's run leading to the output solution. This number includes all the iterations, even the discarded ones.
converge	Only if an iterative algorithm is run: a logical indicating if the algorithm has converged (see <b>Details</b> ).
rep.detail	Only if rep is not equal to 1 with Kozak's original algorithm: a matrix giving the final solution for every repetition of the algorithm. It contains, for each repetition, the value of the criteria to optimize ("opti"), the number of take-all strata ("takeall"), the stratum boundaries ("bh"), the number of iterations of the algorithm for the final number of take-all strata ("niter"), the maxstep value ("maxstep"), the maxstill value ("maxstill"), the type of initial boundaries requested ("initbh.type"), the initial boundaries requested ("initbh"), a logical indicating whether the initial boundaries meet the condition $N_h \geq \min N_h$ ("Nhok.initbh"), a logical indicating whether the initial boundaries meet the condition $n_h > 0$ ("nhok.initbh"), the initial boundaries used to run the algorithm ("bhi"). The boundaries "bhi" are the same as the boundaries "initbh" if "Nhok.initbh" and "nhok.initbh" are TRUE. Otherwise, "bhi" are robust boundaries (see <b>Details</b> ). If robust boundaries have to be used, the last two columns of rep.detail indicate whether these boundaries meet the condition $N_h \geq \min N_h$ ("Nhok.robust") and the condition $n_h > 0$ ("nhok.robust").
rep.min	Only if rep is not equal to 1 with Kozak's original algorithm: the identification number of the algorithm repetition leading to the optimal plan, i.e. the row of rep.detail containing the proposed solution.
sol.detail	Only if the number of possible sets of boundaries is lower than minsol: a matrix giving information for the possible solutions fulfilling the conditions $N_h \geq$

	minNh and $n_h > 0$ . It contains the boundaries ("bh"), the number of units in each stratum ("Nh"), the stratum sample size ("nh"), the value of the criteria to optimize ("opti"), and the number of take-all strata ("takeall").
sol.min	Only if the number of possible sets of boundaries is lower than minsol: the row of sol.detail containing the optimal solution.
nsol	Only if the number of possible sets of boundaries is lower than minsol: the number of possible sets of boundaries (choose(Nu-1, L-1) if takenone=0, choose(Nu, L-1) otherwise).

### Author(s)

Sophie Baillargeon <Sophie.Baillargeon@mat.ulaval.ca> and  
Louis-Paul Rivest <Louis-Paul.Rivest@mat.ulaval.ca>

### References

- Baillargeon, S., Rivest, L.-P., Ferland, M. (2007). Stratification en enquetes entreprises : Une revue et quelques avancees. *Proceedings of the Survey Methods Section, 2007 SSC Annual Meeting*. [http://www.ssc.ca/survey/documents/SSC2007\\_S\\_Baillargeon.pdf](http://www.ssc.ca/survey/documents/SSC2007_S_Baillargeon.pdf)
- Gunning, P. and Horgan, J.M. (2007). Improving the Lavallee and Hidiroglou algorithm for stratification of skewed populations. *Journal of Statistical Computation and Simulation*, **77**(4), 277-291.
- Kozak, M. (2004). Optimal stratification using random search method in agricultural surveys. *Statistics in Transition*, **6**(5), 797-806.
- Lavallee, P. and Hidiroglou, M.A. (1988). On the stratification of skewed populations. *Survey Methodology*, **14**, 33-43.
- Rivest, L.-P. (1999). Stratum jumpers: Can we avoid them? *Proceedings of the Section on Survey Methods Research of the American Statistical Association*, 64-72.
- Rivest, L.-P. (2002). A generalization of the Lavallee and Hidiroglou algorithm for stratification in business surveys. *Survey Methodology*, **28**(2), 191-198.
- Rivest, L.-P. and Baillargeon, S. (2009). A general algorithm for univariate stratification. *International Stratification Review*, **77**(3), 331-344.
- Sethi, V. K. (1963). A note on optimum stratification of populations for estimating the population means. *The Australian Journal of Statistics*, **5**, 20-33.

### See Also

[print.strata](#), [plot.strata](#), [strata.cumrootf](#), [strata.geo](#)

### Examples

```
#####
### Sethi's algorithm versus Kozak's algorithm

# LACK OF CONVERGENCE
# Here is an example of numerical difficulties met with Sethi but not with Kozak
data(UScities)
Sethi<-strata.LH(x=UScities,CV=0.01,Ls=3,alloc=c(0.35,0.35,0),takenone=0,takeall=1,
```

```

      rh=1,model="loglinear",model.control=list(beta=1,sig2=0.5,ph=0.85),algo="Sethi",
      algo.control=list(maxiter=20))
Sethi
Sethi$iter.detail[1:5,]
Kozak<-strata.LH(x=UScities,CV=0.01,Ls=3,alloc=c(0.35,0.35,0),takenone=0,takeall=1,
      rh=1,model="loglinear",model.control=list(beta=1,sig2=0.5,ph=0.85),algo="Kozak")
Kozak
Kozak$iter.detail[Kozak$iter.detail[,"rep"]==Kozak$rep.min,]
# Looking at the iteration history for the optimization with Sethi and Kozak,
# we see that the initial boundaries are very close from the optimal ones.
# Kozak reaches very quickly a minimum. However, Sethi increases n instead of
# minimizing it and afterwards it oscillates between two sets of boundaries
# without converging.

# LOCAL MINIMUM
# In this example, Sethi's algorithm obviously reaches a local minimum since Kozak
# proposes a much smaller n.
data(UScities)
Sethi<-strata.LH(x=UScities,CV=0.01,Ls=4,alloc=c(0.5,0,0),takenone=0,takeall=1,rh=0.85,
      model="loglinear",model.control=list(beta=1.1,sig2=0,ph=1),algo="Sethi")
Kozak<-strata.LH(x=UScities,CV=0.01,Ls=4,alloc=c(0.5,0,0),takenone=0,takeall=1,rh=0.85,
      model="loglinear",model.control=list(beta=1.1,sig2=0,ph=1),algo="Kozak")

#####
### Initial boundaries

# Sometimes some initial boundaries are better than others.
# In the following example, Kozak's algorithm reaches a local minimum with the
# geometric initial boundaries. Initial boundaries forming strata of equal sizes
# (default) allow to reach a much smaller n.
data(Debtors)
default<-strata.LH(x=Debtors,CV=0.01,Ls=4,alloc=c(0.5,0,0.5),takenone=0,takeall=1,
      rh=1,model="loglinear",model.control=list(beta=1,sig2=0.05,ph=0.85),algo="Kozak")
initbh<-strata.geo(x=Debtors,CV=0.01,Ls=4,alloc=c(0.5,0,0.5),rh=1,model="loglinear",
      model.control=list(beta=1,sig2=0.05,ph=0.85))$bh
geo<-strata.LH(x=Debtors,initbh=initbh,CV=0.01,Ls=4,alloc=c(0.5,0,0.5),
      takenone=0,takeall=1,rh=1,model="loglinear",model.control=list(beta=1,
      sig2=0.05,ph=0.85),algo="Kozak")

#####
### Take-none stratum

# As illustrated in the following example, it is sometimes beneficial to include a
# take-none stratum in the stratified design (possibly with a bias penalty lower than 1).
data(Debtors)
sans<-strata.LH(x=Debtors,CV=0.05,Ls=3,alloc=c(0.35,0.35,0),takenone=0,
      takeall=1,rh=c(0.8,0.9,1),model="loglinear",model.control=list(beta=1.1,
      sig2=0,ph=1,ptakenone=1),algo="Kozak")
avec<-strata.LH(x=Debtors,CV=0.05,Ls=3,alloc=c(0.35,0.35,0),takenone=1,
      takeall=1,rh=c(0.8,0.9,1),model="loglinear",model.control=list(beta=1.1,
      sig2=0,ph=1,ptakenone=1),algo="Kozak")
avec0.5<-strata.LH(x=Debtors,CV=0.05,Ls=3,alloc=c(0.35,0.35,0),takenone=1,

```

```

bias.penalty=0.5,takeall=1,rh=c(0.8,0.9,1),model="loglinear",
model.control=list(beta=1.1,sig2=0,ph=1,ptakenone=1),algo="Kozak")

# Note: Sethi does not converge here. This occurs often with a take-none stratum.
avec.Sethi<-strata.LH(x=Debtors,CV=0.05,Ls=3,alloc=c(0.35,0.35,0),takenone=1,
takeall=1,rh=c(0.8,0.9,1),model="loglinear",model.control=list(beta=1.1,
sig2=0,ph=1,ptakenone=1),algo="Sethi")

#####
### Automatic detection of a take-all stratum

# As in the following example, a beneficial take-all stratum is not always
# detected by the algorithm. Therefore, it is often a good idea to obtain a
# stratified design with and without a take-all stratum and to compare
# the results.
data(UScities)
sans<-strata.LH(x=UScities,CV=0.01,Ls=3,alloc=c(0.35,0.35,0),takenone=0,takeall=0,
rh=1,model="loglinear",model.control=list(beta=0.9,sig2=0,ph=1),algo="Kozak")
sans
avec<-strata.LH(x=UScities,CV=0.01,Ls=3,alloc=c(0.35,0.35,0),takenone=0,takeall=1,
rh=1,model="loglinear",model.control=list(beta=0.9,sig2=0,ph=1),algo="Kozak")
avec

#####
### Models for the discrepancy between Y and X

data(Sweden)

# LOGLINEAR MODEL WITH MORTALITY: see help(Sweden)

# HETEROSCEDASTIC LINEAR MODEL: We fix gamma=2.
beta.lin<-mean(Sweden$RMT85/Sweden$REV84)
sig2.lin<-var(Sweden$RMT85/Sweden$REV84)
strata.LH(x=Sweden$REV84,CV=0.05,Ls=5,alloc=c(0.5,0,0.5),takeall=1,
model="linear",model.control=list(beta=beta.lin,sig2=sig2.lin,gamma=2),
algo="Kozak",algo.control=list(method="modified"))
# Verification of equation 3.6 of Rivest (2002)
beta.log<-1
sig2.log<-log(1+sig2.lin/beta.lin^2)
strata.LH(x=Sweden$REV84,CV=0.05,Ls=5,alloc=c(0.5,0,0.5),takeall=1,
model="loglinear",model.control=list(beta=beta.log,sig2=sig2.log,ph=1),
algo="Kozak",algo.control=list(method="modified"))
# The two models give the same stratified design.

# RANDOM REPLACEMENT MODEL: example in Rivest (1999)
strata.LH(x=Sweden$REV84,CV=0.05,Ls=5,alloc=c(0.5,0,0.5),takeall=1,
model="none",algo="Sethi") # Table 1 with a different rounding of the nh's
e0<-strata.LH(x=Sweden$REV84,CV=0.05,Ls=5,alloc=c(0.5,0,0.5),takeall=1,
model="none",algo="Kozak")
e0 # Better than Sethi
var.strata(e0,y=Sweden$RMT85)

```

```
e0.001<-strata.LH(x=Sweden$REV84,CV=0.05,Ls=5,alloc=c(0.5,0,0.5),takeall=1,
  model="random",model.control=list(epsilon=0.011),algo="Kozak")
e0.001 # Table 2 part 3 with a different rounding of the nh's
var.strata(e0.001,y=Sweden$RMT85)
```

---

strata.rule

---

*Non-Iterative Methods of Strata Construction*


---

## Description

These functions first determine boundaries to stratify a population. Then, in a second independent step, the sample sizes are calculated given a CV or the CV is computed given the total sample size. The function `strata.cumrootf` uses the cumulative root frequency method by Dalenius and Hodges (1959) and `strata.geo` uses the geometric method by Gunning and Horgan (2004). A model can be specified for the relationship between the stratification variable  $X$  and the survey variable  $Y$ , but this model has no impact on the first step of boundary determination. It only influences the calculation of the  $n$  or of the CV by the use of anticipated means and variances of  $Y$  instead of the empirical means and variances of  $X$ .

## Usage

```
strata.cumrootf(x, nclass, n = NULL, CV = NULL, Ls = 3, certain = NULL,
  alloc = list(q1 = 0.5, q2 = 0, q3 = 0.5), rh = rep(1, Ls),
  model = c("none", "loglinear", "linear", "random"),
  model.control = list())
```

```
strata.geo(x, n = NULL, CV = NULL, Ls = 3, certain=NULL,
  alloc = list(q1 = 0.5, q2 = 0, q3 = 0.5), rh = rep(1, Ls),
  model = c("none", "loglinear", "linear", "random"),
  model.control = list())
```

## Arguments

<code>x</code>	A vector containing the values of the stratification variable $X$ for every unit in the population.
<code>nclass</code>	A numeric for the cumulative root frequency method only: the number of classes (see <b>Details</b> ). The default is $\min(L*10, \text{length}(x))$ .
<code>n</code>	A numeric: the target sample size. It has no default value. The argument $n$ or the argument CV must be input.
<code>CV</code>	A numeric: the target coefficient of variation. It has no default value. The argument CV or the argument $n$ must be input.
<code>Ls</code>	A numeric: the number of sampled strata (here $Ls$ is always equal to $L$ because no take-none stratum can be added to the stratified design). The default is 3.
<code>certain</code>	A vector giving the position, in the vector $x$ , of the units that must be included in the sample (see <a href="#">stratification-package</a> ). By default <code>certain</code> is NULL, which means that no units are chosen a priori to be in the sample.

alloc	A list specifying the allocation scheme. The list must contain 3 numerics for the 3 exponents q1, q2 and q3 in the general allocation scheme (see <a href="#">stratification-package</a> ). The default is Neyman allocation (q1=q3=0.5 and q2=0)
rh	A vector giving the anticipated response rates in each of the Ls sampled strata. A single number can be given if the rates do not vary among strata. The default is 1 in each stratum.
model	A character string identifying the model used to describe the discrepancy between the stratification variable $X$ and the survey variable $Y$ . It can be "none" if one assumes $Y = X$ , "loglinear" for the loglinear model with mortality, "linear" for the heteroscedastic linear model or "random" for the random replacement model (see <a href="#">stratification-package</a> for a description of these models). The default is "none".
model.control	A list of model parameters (see <a href="#">stratification-package</a> ). The default values of the parameters correspond to the model $Y = X$ .

### Details

The efficiency of the cumulative root frequency method depends on the number of classes nclass. However, there is no theory about how to choose the best value for nclass (Hedlin 2000). This is a limit of the method.

### Value

Nh	A vector of length $L$ containing the population sizes $N_h$ , i.e. the number of units in each stratum.
nh	A vector of length $L$ containing the sample sizes $n_h$ , i.e. the number of units to sample in each stratum. See <a href="#">stratification-package</a> for information about the rounding used to get these values.
n	The total sample size (sum(nh)).
nh.nonint	A vector of length $L$ containing the non integer values of sample sizes $n_h$ .
certain.info	A vector giving statistics for the certain stratum (see <a href="#">stratification-package</a> ). It contains Nc, the number of units chosen a priori to be in the sample, and meanc, the anticipated mean of $Y$ for these units.
opti.criteria	The final value of the criteria to optimize : either the non-integer value of the total sample size (if a target CV was given) or the RRMSE associated to non-integer stratum sample sizes (if a target n was given).
bh	A vector of the $L - 1$ stratum boundaries proposed by the method.
meanh	A vector of length $L$ containing the anticipated means of $Y$ in each stratum.
varh	A vector of length $L$ containing the anticipated variances of $Y$ in each stratum.
mean	A numeric: the anticipated global mean value of $Y$ .
stderr	A numeric: the standard error of the anticipated global mean of $Y$ .
CV	The anticipated coefficient of variation for the mean of $Y$ , i.e. stderr divided mean.

stratumID	A factor identifying, for each observation in the input $X$ , the stratum to which it has been assigned. The value "certain" is given to units a priori chosen to be in the sample.
nclassh	A vector for the cumulative root frequency method only: the number of classes in each stratum.
takeall	The number of take-all stratum in the final solution.
call	The function call (object of class "call").
date	A character string that contains the system date and time when the function ended.
args	A list of all the arguments input to the function or used by default.

### Author(s)

Sophie Baillargeon <Sophie.Baillargeon@mat.ulaval.ca> and  
Louis-Paul Rivest <Louis-Paul.Rivest@mat.ulaval.ca>

### References

- Dalenius, T. and Hodges, J.L., Jr. (1959). Minimum variance stratification. *Journal of the American Statistical Association*, **54**, 88-101.
- Gunning, P. and Horgan, J.M. (2004). A new algorithm for the construction of stratum boundaries in skewed populations. *Survey Methodology*, **30**(2), 159-166.
- Hedlin, D. (2000). A procedure for stratification by an extended Ekman rule. *Journal of Official Statistics*, **61**, 15-29.

### See Also

[print.strata](#), [plot.strata](#), [strata.LH](#)

### Examples

```
### Example for strata.cumrootf
data(MRTS)
res<-NULL
for ( n in seq(100,1500,10)){
  cum<-strata.cumrootf(x=MRTS,nclass=n,CV=0.01,Ls=4,alloc=c(0.5,0,0.5))
  res<-rbind(res,c(n,cum$n))
}
plot(res,ylab="suggested sample size n",xlab="number of classes",main=expression(
  paste("Example of the effect of nclass on n for the cum",sqrt(f)," method")))

### Example for strata.geo
data(Sweden)
strata.geo(x=Sweden$REV84,CV=0.05,Ls=3,alloc=c(0.5,0,0.5),model="none")
strata.geo(x=Sweden$REV84,CV=0.05,Ls=3,alloc=c(0.5,0,0.5),model="loglinear",
  model.control=list(beta=1.058355,sig2=0.06593083,ph=1))
strata.geo(x=Sweden$REV84,CV=0.05,Ls=3,alloc=c(0.5,0,0.5),rh=0.85,
  model="loglinear",model.control=list(beta=1.058355,sig2=0.06593083,ph=1))
# When non-response or a model is added, the stratum boundaries do not change,
# only the nh's do.
```

**Description**

`print.strata` prints a "strata" object, presenting the stratification information into a table.

`plot.strata` produces a histogram of the stratification variable  $X$  in which the stratification boundaries are drawn. A table with the  $N_h$  and  $n_h$  values is also added at the top of the plot.

**Usage**

```
## S3 method for class 'strata'
print(x, ...)

## S3 method for class 'strata'
plot(x, logscale = FALSE, drop = 0, main =
      paste("Graphical Representation of the Stratified Design", xname),
      xlab, ...)
```

**Arguments**

<code>x</code>	An object of class "strata" to print or to plot.
<code>logscale</code>	A logical indicating whether the $X$ axis should be represented on the log scale or not. The default is FALSE.
<code>drop</code>	A integer indicating how many of the largest values of $X$ should be omitted in the plot. This argument is useful when some large values of $X$ stretch the $X$ range too much.
<code>main</code>	A character string giving the title of the plot.
<code>xlab</code>	A character string naming the $X$ axis.
<code>...</code>	Additional arguments affecting the print or the plot produced.

**Note**

When the object of class "strata" contains a certain stratum, `plot.strata` removes from the data the units in this stratum before generating the histogram.

**Author(s)**

Sophie Baillargeon <Sophie.Baillargeon@mat.ulaval.ca> and  
Louis-Paul Rivest <Louis-Paul.Rivest@mat.ulaval.ca>

**See Also**

[strata.bh](#), [strata.cumrootf](#), [strata.geo](#), [strata.LH](#)

**Examples**

```

data(MRTS)
cumrootf<-strata.cumrootf(x=MRTS,nclass=500,CV=0.01,Ls=4,alloc=c(0.5,0,0.5))
print(cumrootf)
plot(cumrootf)
plot(cumrootf,drop=5)
plot(cumrootf,logscale=TRUE)
geo<-strata.geo(x=MRTS,CV=0.01,Ls=4,alloc=c(0.5,0,0.5))
print(geo)
plot(geo,logscale=TRUE)
# The geometric method does not perform well because of some small units
LH<-strata.LH(x=MRTS,CV=0.01,Ls=4,alloc=c(0.5,0,0.5),takeall=1)
print(LH)
plot(LH,logscale=TRUE)

```

---

Sweden

*The MU284 Population of Sweden Municipalities from Sarndal et al. (1992)*


---

**Description**

This data set comes from Sarndal et al.'s book (1992), Appendix B. It contains different variables that describe 284 municipalities in Sweden.

**Usage**

```
data(Sweden)
```

**Format**

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

id Identifier running from 1 to 284

P85 1985 population (in thousands)

P75 1975 population (in thousands)

RMT85 Revenues from the 1985 municipal taxation (in millions of kronor)

CS82 Number of Conservative seats in municipal council

SS82 Number of Social-Democratic seats in municipal council

S82 Total number of seats in municipal council

ME84 Number of municipal employees in 1984

REV84 Real estate values according to 1984 assessment (in millions of kronor)

REG Geographic region indicator

CL Cluster indicator (a cluster consists of a set of neighboring municipalities)

## Details

In this package, REV84 is used as a stratification variable and RMT85 as a survey variable.

## Source

Sarndal, C. E., Swensson, B. and Wretman, J. (1992). *Model Assisted Survey Sampling*. Springer Verlag, New York.

## References

Rivest, L.-P. (2002). A generalization of the Lavallee and Hidioglou algorithm for stratification in business surveys. *Survey Methodology*, **28**(2), 191-198.

## Examples

```
data(Sweden)
X<-Sweden$REV84
Y<-Sweden$RMT85

# Study of the relationship between X and Y
plot(log(X),log(Y))
# Extreme values are omitted for a more robust estimation
keep<-X/Y>quantile(X/Y,0.03)&X/Y<quantile(X/Y,0.97)
plot(log(X)[keep],log(Y)[keep])
reg<-lm( log(Y)[keep]~log(X)[keep] )
summary(reg)

# Stratification assuming X=Y
nomodel<-strata.LH(x=X,CV=0.05,Ls=3,alloc=c(0.5,0,0.5),takeall=1,model="none")
nomodel
var.strata(nomodel,y=Y) # The target CV is not reached

# Stratification taking into account a loglinear model between X and Y,
# using the estimated parameters values
model<-strata.LH(x=X,CV=0.05,Ls=3,alloc=c(0.5,0,0.5),takeall=1,model="loglinear",
  model.control=list(beta=reg$coef[2],sig2=summary(reg)$sigma^2,ph=1))
model
var.strata(model,y=Y) # The target CV is reached
```

---

var.strata

*Anticipated Variances and RRMSE from a Stratified Design for a Survey Variable Y*

---

## Description

var.strata calculates the anticipated means, variances and relative root mean squared error (RRMSE) obtained when applying a stratified design to a survey variable  $Y$ . The variable  $Y$  can be input or it can be defined from  $X$  by a specified loglinear with mortality, heteroscedastic linear or random

replacement model.

`print.var.strata` prints a "var.strata" object, presenting the stratification information into a table.

### Usage

```
var.strata(strata, y = NULL, rh = strata$args$rh, rh.postcorr =
          FALSE, model = c("none", "loglinear", "linear", "random"),
          model.control = list())
```

```
## S3 method for class 'var.strata'
print(x, ...)
```

### Arguments

<code>strata</code>	An object of class "strata".
<code>y</code>	A vector containing the values of the survey variable $Y$ for every unit of the population. The default is that $Y$ is not given.
<code>rh</code>	A vector giving the anticipated response rates in each of the $L_s$ sampled strata. A single number can be given if the rates do not vary among strata. The default is to use the rates given in the <code>strata.bh</code> object.
<code>rh.postcorr</code>	A logical. If TRUE, a posterior correction for non-response is applied. This correction takes into account the non-response in the <code>strata.bh</code> object. The default is FALSE, i.e. no posterior correction is made (see <b>Details</b> ).
<code>model</code>	A character string identifying the model used to describe the discrepancy between the stratification variable $X$ and the survey variable $Y$ . It can be "none" if one assumes $Y = X$ , "loglinear" for the loglinear model with mortality, "linear" for the heteroscedastic linear model or "random" for the random replacement model (see <a href="#">stratification-package</a> for a description of these models). The default is "none".
<code>model.control</code>	A list of model parameters (see <a href="#">stratification-package</a> ). The default values of the parameters correspond to the model $Y = X$ .
<code>x</code>	An object of class "var.strata" to print.
<code>...</code>	Additional arguments affecting the print produced.

### Details

#### POSTERIOR CORRECTION FOR NON-RESPONSE

The optional posterior correction for non-response is done as follows. For each take-some stratum,  $n_h$  is increased if the input `rh` is lower than the anticipated response rate in the `strata.bh` object, and  $n_h$  is decreased if the input `rh` is higher than the anticipated response rate given when creating the `strata.bh` object. The modification of  $n_h$  is done by multiplying it by `strata$args$rh/rh`.

The weakness of this posterior correction is that it cannot take into account non-response in a take-all stratum. In that stratum,  $n_h$  cannot be increased since it is equal to  $N_h$ . To correctly account for non-response in a take-all stratum, the boundary of the stratum has to be lowered. This is what the generalized Lavallee-Hidirolou method does ([strata.LH](#)).

**Value**

nh	A vector of length $L$ containing the sample sizes $n_h$ , i.e. the number of units to sample in each stratum. This number can be different than <code>strata\$nh</code> if <code>rh.postcorr=TRUE</code> .
n	The total sample size ( <code>sum(nh)</code> ). This number can be different than <code>strata\$n</code> if <code>rh.postcorr=TRUE</code> .
certain.info	A vector giving statistics for the certain stratum (see <a href="#">stratification-package</a> ). It contains <code>Nc</code> , the number of units chosen a priori to be in the sample, and <code>meanc</code> , the anticipated mean of $Y$ for these units.
meanh	A vector of length $L$ containing the anticipated means of $Y$ in each stratum.
varh	A vector of length $L$ containing the anticipated variances of $Y$ in each stratum.
mean	A numeric: the anticipated global mean value of $Y$ .
RMSE	A numeric: the root mean squared error (or standard error if <code>strata\$args\$takenone=0</code> ) of the anticipated global mean of $Y$ . This is defined as the squared root of: $(\text{bias.penalty} \times \text{bias of the mean})^2 + \text{variance of the mean}$ .
RRMSE	A numeric: the anticipated relative root mean squared error (or coefficient of variation if <code>strata\$args\$takenone=0</code> ) for the mean of $Y$ , i.e. RMSE divided by mean.
relativebias	A numeric: the anticipated relative bias of the estimator, i.e. $(\text{bias.penalty} \times \text{bias of the mean})$ divided by mean. If <code>strata\$args\$takenone=0</code> , this numeric is zero.
propbiasMSE	A numeric: the proportion of the MSE attributable to the bias of the estimator, i.e. $(\text{bias.penalty} \times \text{bias of the mean})^2$ divided by the MSE of the mean. If <code>strata\$args\$takenone=0</code> , this numeric is zero.
call	The function call (object of class "call").
date	A character string that contains the system date and time when the function ended.
args	A list of all the arguments input to the function or used by default.

**Author(s)**

Sophie Baillargeon <Sophie.Baillargeon@mat.ulaval.ca> and  
 Louis-Paul Rivest <Louis-Paul.Rivest@mat.ulaval.ca>

**See Also**

[strata.bh](#), [strata.cumrootf](#), [strata.geo](#), [strata.LH](#)

**Examples**

```
data(Sweden)
nomodel<-strata.LH(x=Sweden$REV84,CV=0.05,Ls=3,alloc=c(0.5,0,0.5),
  takeall=1,model="none")
# We can give a vector of the Y values for every unit in the population
var.strata(nomodel,y=Sweden$RMT85)
```

```
# Or specify a model between X and Y
var.strata(nomodel,model="loglinear",model.control=list(beta=1.058355,
  sig2=0.06593083,ph=1))
# Compared to taking into account the model in the optimization
model<-strata.LH(x=Sweden$REV84,CV=0.05,Ls=3,alloc=c(0.5,0,0.5),
  takeall=1,model="loglinear",model.control=list(beta=1.058355,
  sig2=0.06593083,ph=1))
var.strata(model,y=Sweden$RMT85)

### Examples of posterior correction for non-response
data(MRTS)
LH<-strata.LH(x=MRTS,CV=0.01,Ls=4,alloc=c(0.5,0,0.5),takeall=1)
LH
# Without non-response in the take-all strata
var.strata(LH,rh.postcorr=TRUE,rh=c(0.85,0.9,0.9,1))
strata.LH(x=MRTS,CV=0.01,Ls=4,alloc=c(0.5,0,0.5),takeall=1,rh=c(0.85,0.9,0.9,1))
# With non-response in the take-all strata
var.strata(LH,rh.postcorr=TRUE,rh=0.9)
strata.LH(x=MRTS,CV=0.01,Ls=4,alloc=c(0.5,0,0.5),takeall=1,rh=0.9)
```

# Index

## \*Topic **datasets**

- CochranHorganData, [6](#)
- MRTS, [8](#)
- SHS, [9](#)
- Sweden, [26](#)

## \*Topic **methods**

- strata.tool, [25](#)

## \*Topic **package**

- stratification-package, [2](#)

## \*Topic **survey**

- strata.bh, [10](#)
- strata.LH, [13](#)
- strata.rule, [22](#)
- var.strata, [27](#)

CochranHorganData, [6](#)

Debtors (CochranHorganData), [6](#)

MRTS, [8](#)

plot.strata, [2](#), [13](#), [19](#), [24](#)

plot.strata (strata.tool), [25](#)

print.strata, [2](#), [13](#), [19](#), [24](#)

print.strata (strata.tool), [25](#)

print.var.strata (var.strata), [27](#)

SHS, [9](#)

strata.bh, [2–4](#), [10](#), [25](#), [29](#)

strata.cumrootf, [2–4](#), [13](#), [19](#), [25](#), [29](#)

strata.cumrootf (strata.rule), [22](#)

strata.geo, [2–4](#), [13](#), [19](#), [25](#), [29](#)

strata.geo (strata.rule), [22](#)

strata.LH, [2–4](#), [13](#), [13](#), [24](#), [25](#), [28](#), [29](#)

strata.rule, [22](#)

strata.tool, [25](#)

stratification

- (stratification-package), [2](#)

stratification-package, [11](#), [12](#), [14](#), [15](#), [17](#),  
[22](#), [23](#), [28](#), [29](#)

stratification-package, [2](#)

Sweden, [26](#)

USbanks (CochranHorganData), [6](#)

UScities (CochranHorganData), [6](#)

UScolleges (CochranHorganData), [6](#)

var.strata, [2](#), [27](#)