

# Package ‘stratallo’

October 14, 2022

**Title** Optimum Sample Allocation in Stratified Sampling Schemes

**Version** 2.1.0

**Description** Functions in this package provide solution to classical problem in survey methodology - an optimum sample allocation in stratified sampling schemes. In this context, the optimal allocation is in the classical Tschuprov-Neyman's sense and it satisfies additional lower or upper bounds restrictions imposed on sample sizes in strata. There are few different algorithms available to use, and one them is based on popular sample allocation method that applies Neyman allocation to recursively reduced set of strata.

This package also provides the function that computes a solution to the minimum sample size allocation problem, which is a minor modification of the classical optimum sample allocation. This problem lies in the determination of a vector of strata sample sizes that minimizes total sample size, under assumed fixed level of the pi-estimator's variance. As in the case of the classical optimal allocation, the problem of minimum sample size allocation can be complemented by imposing upper bounds constraints on sample sizes in strata.

**License** GPL-2

**URL** <https://github.com/wojciech/stratallo>

**BugReports** <https://github.com/wojciech/stratallo/issues>

**Copyright** Wojciech Wójciak

**Language** en-US

**Encoding** UTF-8

**LazyData** true

**Imports** checkmate, lifecycle

**RoxygenNote** 7.2.0

**Suggests** rmarkdown, knitr, spelling, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Collate** 'helpers.R' 'algorithms\_onesided.R' 'algorithms\_twosided.R'  
'opt.R' 'pop.R' 'stratallo-package.R' 'variance.R'

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Wojciech Wójciak [aut, cre],  
 Jacek Wesołowski [sad],  
 Robert Wieczorkowski [ctb]

**Maintainer** Wojciech Wójciak <wojciech.wojciak@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-09-13 16:10:02 UTC

## R topics documented:

stratallo-package . . . . .	2
allocation_summary . . . . .	3
dopt . . . . .	4
dopt_upper . . . . .	7
h_get_which_violates . . . . .	9
nopt . . . . .	10
pop10_mM . . . . .	12
pop507 . . . . .	12
pop969 . . . . .	13
rnabox . . . . .	13
rna_onesided . . . . .	15
var_tst . . . . .	17

**Index** **19**

---

stratallo-package	<i>Functions for Optimal Sample Allocation in Stratified Sampling Schemes</i>
-------------------	---

---

## Description

Optimal Sample Allocation in Stratified Sampling Schemes

## Author(s)

Wojciech Wójciak <wojciech.wojciak@gmail.com>

## References

Wesołowski, J., Wieczorkowski, R., Wójciak, W. (2021), Optimality of the recursive Neyman allocation, *Journal of Survey Statistics and Methodology*, doi: [10.1093/jssam/smab018](https://doi.org/10.1093/jssam/smab018), doi: [10.48550/arXiv.2105.14486](https://doi.org/10.48550/arXiv.2105.14486)

Wójciak, W. (2022), Minimum sample size allocation in stratified sampling under constraints on variance and strata sample sizes, doi: [10.48550/arXiv.2204.04035](https://doi.org/10.48550/arXiv.2204.04035)

Wójciak, W. (2019), Optimal allocation in stratified sampling schemes, *MSc Thesis*, Warsaw University of Technology, Warsaw, Poland. [http://home.elka.pw.edu.pl/~wojciak/msc\\_optimal\\_allocation.pdf](http://home.elka.pw.edu.pl/~wojciak/msc_optimal_allocation.pdf)

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

---

allocation\_summary      *Summarizing the Allocation*

---

## Description

### [Stable]

A helper function that creates and returns a simple `data.frame` with summary of the allocation as returned by the `dopt()` or `nopt()`. See the illustrate example below.

## Usage

```
allocation_summary(x, a, m = NULL, M = NULL)
```

## Arguments

x	(numeric) the allocation vector.
a	(numeric) parameters $a_1, \dots, a_H$ of variance function $D$ . See <code>dopt()</code> or <code>nopt()</code> for more details.
m	(numeric or NA) lower bounds constraints imposed on sample sizes in strata.
M	(numeric or NA) upper bounds constraints imposed on sample sizes in strata.

## Details

The summary `data.frame` contains the following columns:

- (1) the allocation vector,
  - (2) lower bounds constraints (optional),
  - (3) upper bounds constraints (optional),
  - (4) indication whether the allocation for a given stratum is of take-min type, i.e.  $x == m$  (optional),
  - (5) indication whether the allocation for a given stratum is of take-max type, i.e.  $x == M$  (optional),
  - (6) indication whether the allocation for a given stratum is of take-neyman type, i.e.  $m < x < M$ .
- The last row contains the sum of all of the values from the other rows (wherever feasible).

## Value

A `data.frame` with the allocation summary.

**Examples**

```

a <- c(3000, 4000, 5000, 2000)
m <- c(100, 90, 70, 80)
M <- c(200, 150, 300, 210)

x1 <- dopt(n = 400, a, m)
s1 <- allocation_summary(x1, a, m)

x2 <- dopt(n = 540, a, m, M)
s2 <- allocation_summary(x2, a, m, M)

```

dopt

*Optimum Sample Allocation in Stratified Sampling Schemes***Description****[Stable]**

A classical problem in survey methodology in stratified sampling is an optimum sample allocation problem. This problem is formulated as determination of a vector of strata sample sizes that minimizes the variance of the pi-estimator of the population total of a given study variable, under constraint on total sample size.

The `dopt()` function solves the problem of optimum sample allocation under lower or upper bounds constraints, optionally imposed on strata sample sizes. The allocation computed is valid for all stratified sampling schemes for which the variance of the stratified pi-estimator is of the form:

$$D(x_1, \dots, x_H) = a_1^2/x_1 + \dots + a_H^2/x_H - b,$$

where  $H$  denotes total number of strata,  $x_1, \dots, x_H$  are the strata sample sizes, and  $b, a_w > 0$  do not depend on  $x_w, w = 1, \dots, H$ .

The `dopt()` function makes use of the following allocation algorithms: `rna`, `sga`, `sgaplus`, `coma` for optimal sample allocation under one-sided upper bounds constraints, and `lrna` for optimal sample allocation under one-sided lower bounds constraints. For the allocation under box-constraints, the `rnabox` algorithm is used. The `rna`, `sga`, and `coma` are described in Wesołowski et al. (2021), while the `sgaplus` is in Wójciak (2019). The `lrna` is introduced in Wójciak (2022). The `rnabox` algorithm is a new optimal allocation algorithm that was developed by the authors of this package and will be published soon.

**Usage**

```
dopt(n, a, m = NULL, M = NULL, M_method = "rna")
```

**Arguments**

`n` (number)  
total sample size. A strictly positive scalar.

a	(numeric) parameters $a_1, \dots, a_H$ of variance function $D$ . Strictly positive numbers.
m	(numeric or NULL) lower bounds constraints optionally imposed on strata sample sizes. If not NULL, it is then required that $n \geq \text{sum}(m)$ . Strictly positive numbers.
M	(numeric or NULL) upper bounds constraints optionally imposed on strata sample sizes. If not NULL, it is then required that $n \leq \text{sum}(M)$ . Strictly positive numbers.
M_method	(string) the name of the underlying algorithm to be used for computing a sample allocation under one-sided upper bounds constraints One of the following: rna (default), sga, sgaplust, coma. This parameter is used only in case when m argument is NULL and M is not NULL.

### Details

The `dopt()` function computes:

$$\text{argmin}D(x_1, \dots, x_H),$$

under the equality constraint imposed on total sample size:

$$x_1 + \dots + x_H = n,$$

and inequality constraints (optimally) imposed on strata sample size:

$$m_w \leq x_w \leq M_w, w = 1, \dots, H.$$

Here,  $H$  denotes total number of strata,  $x_1, \dots, x_H$  are the strata sample sizes, and  $n > 0$ ,  $b_w > 0, w = 1, \dots, H$ , are given numbers. Furthermore,  $m_w > 0$  and  $M_w > 0, w = 1, \dots, H$  are lower and upper bounds respectively, optionally imposed on sample sizes in strata.

User of `dopt()` can choose whether the inequality constraints will be added to the optimization problem or not. This is achieved with the proper use of `m` and `M` arguments of the function. In case of no inequality constraints to be added, `m` and `M` must be both specified as `NULL` (default). If only upper bounds constraints should be added, it should be specified using `M` argument, while leaving `m` as `NULL`. If only lower bounds constraints should be added, user must specify it with `m` argument, while leaving `M` as `NULL`. Finally, in case of box-constraints, both parameters `m` and `M` must be specified.

For the case of one-sided upper bounds constraints only, there are four different underlying algorithms available to use. These are abbreviated as: "rna" (`rna_onesided()`), "sga" (`sga()`), "sgaplust" (`sgaplust()`), and "coma" (`coma()`). Functions names that perform given algorithms are given in the brackets. See its help page for more details. For the case of one-sided lower bounds constraints only, the "rna" (`rna_onesided()`) is used. Finally, for box-constraints, the "rnabox" algorithm is used (`rnabox()`).

### Value

Numeric vector with optimal sample allocation in strata.

**Note**

For simple random sampling without replacement design in each stratum, parameters of the variance function  $D$  are  $b = N_1 * S_1^2 + \dots + N_H * S_H^2$ , and  $a_w = N_w * S_w$ , where  $N_w, S_w, w = 1, \dots, H$ , are strata sizes and standard deviations of a study variable in strata respectively.

Note that if no inequality constraints are imposed, the optimal allocation is given as a closed form expression, known as Neyman allocation:

$$x_w = a_w * n / (a_1 + \dots + a_H), w = 1, \dots, H.$$

**References**

Wesołowski, J., Wieczorkowski, R., Wójciak, W. (2021), Optimality of the recursive Neyman allocation, *Journal of Survey Statistics and Methodology*, doi: [10.1093/jssam/smab018](https://doi.org/10.1093/jssam/smab018), doi: [10.48550/arXiv.2105.14486](https://doi.org/10.48550/arXiv.2105.14486)

Wójciak, W. (2022), Minimum sample size allocation in stratified sampling under constraints on variance and strata sample sizes, doi: [10.48550/arXiv.2204.04035](https://doi.org/10.48550/arXiv.2204.04035)

Wójciak, W. (2019), Optimal allocation in stratified sampling schemes, *MSc Thesis*, Warsaw University of Technology, Warsaw, Poland. [http://home.elka.pw.edu.pl/~wojciak/msc\\_optimal\\_allocation.pdf](http://home.elka.pw.edu.pl/~wojciak/msc_optimal_allocation.pdf)

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

**See Also**

[nopt\(\)](#), [rna\\_onesided\(\)](#), [sga\(\)](#), [sgaplust\(\)](#), [coma\(\)](#), [rnabox\(\)](#).

**Examples**

```
a <- c(3000, 4000, 5000, 2000)
m <- c(100, 90, 70, 50)
M <- c(300, 400, 200, 90)

# Only lower bounds.
dopt(n = 340, a = a, m = m)
dopt(n = 400, a = a, m = m)
dopt(n = 700, a = a, m = m)

# Only upper bounds.
dopt(n = 190, a = a, M = M)
dopt(n = 700, a = a, M = M)

# Box-constraints.
dopt(n = 340, a = a, m = m, M = M)
dopt(n = 500, a = a, m = m, M = M)
dopt(n = 800, a = a, m = m, M = M)
```

```

# Example of execution-time comparison of different algorithms
# using bench R package.
## Not run:
N <- pop969[, "N"]
S <- pop969[, "S"]
a <- N * S
nfrac <- seq(0.01, 0.9, 0.05)
n <- setNames(as.integer(nfrac * sum(N)), nfrac)
lapply(
  n,
  function(ni) {
    bench::mark(
      dopt(ni, a, M = N, M_method = "rna"),
      dopt(ni, a, M = N, M_method = "sga"),
      dopt(ni, a, M = N, M_method = "sgaplus"),
      dopt(ni, a, M = N, M_method = "coma"),
      iterations = 200
    )[c(1, 3)]
  }
)

## End(Not run)

```

dopt\_upper

*Algorithms for Optimum Sample Allocation in Stratified Sampling Under One-Sided Upper Bounds Constraints*

## Description

### [Stable]

Internal functions that implement the optimal sample allocation algorithms: `sga`, `sgaplus` and `coma`. Functions from this family compute:

$$\operatorname{argmin} D(x_1, \dots, x_H) = a_1^2/x_1 + \dots + a_H^2/x_H - b,$$

under the equality constraint imposed on total sample size:

$$x_1 + \dots + x_H = n,$$

and upper bounds constraints imposed on strata sample sizes:

$$x_w \leq M_w, w = 1, \dots, H.$$

Here,  $H$  denotes total number of strata,  $x_1, \dots, x_H$  are the strata sample sizes, and  $n > 0$ ,  $b, a_w > 0$ ,  $M_w > 0$ ,  $w = 1, \dots, H$  are given numbers.

The `sga()`, `sgaplus()` and `coma()` are internal implementations of the algorithms in subject, and hence, users should not use any of these functions directly. Instead, the `dopt()` should be used.

**Usage**

sga(n, a, M)

sgaplus(n, a, M)

coma(n, a, M)

**Arguments**

- |   |  |
|---|--|
| n | (number)   |
|   | total sample size. A strictly positive scalar.   |
| a | (numeric)  |
|   | parameters $a_1, \dots, a_H$ of variance function $D$ . Strictly positive numbers.   |
| M | (numeric)  |
|   | upper bounds constraints imposed on strata sample sizes. It is required that $n \leq \sum(M)$ . Strictly positive numbers. |

**Value**

Numeric vector with optimal sample allocations in strata.

**Functions**

- sga: implementation of the Stenger-Gabler type algorithm SGa, described in Wesołowski et al. (2021) and in Stenger and Gabler (2005).
- sgaplus: implementation of the modified Stenger-Gabler type algorithm, described in Wójciak (2019) as Sequential Allocation (version 1) algorithm.
- coma: implementation of the Change of Monotonicity Algorithm, or coma, described in Wesołowski et al. (2021).

**Note**

For simple random sampling without replacement design in each stratum, parameters of the variance function  $D$  are  $b = N_1 * S_1^2 + \dots + N_H * S_H^2$ , and  $a_w = N_w * S_w$ , where  $N_w, S_w, w = 1, \dots, H$ , are strata sizes and standard deviations of a study variable in strata respectively.

**References**

Wesołowski, J., Wieczorkowski, R., Wójciak, W. (2021), Optimality of the recursive Neyman allocation, *Journal of Survey Statistics and Methodology*, doi: [10.1093/jssam/smab018](https://doi.org/10.1093/jssam/smab018), doi: [10.48550/arXiv.2105.14486](https://doi.org/10.48550/arXiv.2105.14486)

Stenger, H., Gabler, S. (2005), Combining random sampling and census strategies - Justification of inclusion probabilities equal 1, *Metrika*, 61, 137-156

Wójciak, W. (2019), Optimal allocation in stratified sampling schemes, *MSc Thesis*, Warsaw University of Technology, Warsaw, Poland. [http://home.elka.pw.edu.pl/~wojciak/msc\\_optimal\\_](http://home.elka.pw.edu.pl/~wojciak/msc_optimal_)



[allocation.pdf](#)

### See Also

[dopt\(\)](#), [rna\\_onesided\(\)](#).

### Examples

```
a <- c(3000, 4000, 5000, 2000)
M <- c(100, 90, 70, 80)
sga(n = 190, a = a, M = M)
sgaplus(n = 190, a = a, M = M)
coma(n = 190, a = a, M = M)
```

---

h\_get\_which\_violates *Get Proper Version of which\_violates Function*

---

### Description

#### [Stable]

An internal function that prepares a simple 2-arguments wrapper of `base::which()` that checks whether its first argument exceeds the second one. Both arguments are numeric. This excess is hard coded in the returned wrapper function and it is defined either as `>=` ("greater or equal") or `<=` ("lower or equal"), depending on the value of the `geq` flag.

### Usage

```
h_get_which_violates(geq = TRUE)
```

### Arguments

<code>geq</code>	(flag) if TRUE, then "greater or equal" condition is set. Otherwise, "less then or equal" is set.
------------------	--

### Value

2-arguments function that checks whether its first argument exceeds the second one. Both arguments must be numeric.

### See Also

[rna\\_onesided\(\)](#).

**Examples**

```

which_violates <- stratallo:::h_get_which_violates()
which_violates(1:3, 3:1)

which_violates <- stratallo:::h_get_which_violates(geq = FALSE)
which_violates(1:3, 3:1)

```

---

nopt

---

*Minimum Sample Size Allocation in Stratified Sampling Schemes*


---

**Description****[Stable]**

User function that determines fixed strata sample sizes that minimize total sample size, under assumed level of the variance of the stratified pi-estimator of the total and optional one-sided upper bounds imposed on strata sample sizes. The algorithm used by `nopt()` is described in Wójciak (2022). The allocation computed is valid for all stratified sampling schemes for which the variance of the stratified pi-estimator is of the form:

$$D(x_1, \dots, x_H) = a_1^2/x_1 + \dots + a_H^2/x_H - b,$$

where  $H$  denotes total number of strata,  $x_1, \dots, x_H$  are the strata sample sizes, and  $b, a_w > 0$  do not depend on  $x_w, w = 1, \dots, H$ .

**Usage**

```
nopt(D, a, b, M = NULL)
```

**Arguments**

D	(number) variance equality constraint value. A strictly positive scalar.
a	(numeric) parameters $a_1, \dots, a_H$ of variance function $D$ . Strictly positive numbers.
b	(number) parameter $b$ of variance function $D$ .
M	(numeric or NULL) upper bounds constraints optionally imposed on strata sample sizes. If different than NULL, it is then required that $D \geq \sum(a/M) - b > 0$ . Strictly positive numbers.

**Details**

The `nopt()` function computes:

$$\operatorname{argminn}(x_1, \dots, x_H) = x_1 + \dots + x_H,$$

under the equality constraint imposed on the variance:

$$a_1^2/x_1 + \dots + a_H^2/x_H - b = D.$$

Optionally, the following set of one-sided inequality constraints can be added:

$$x_w \leq M_w, w = 1, \dots, H,$$

where  $D > 0$  is a given number and  $M_w > 0, w = 1, \dots, H$ , are the upper bounds imposed on sample sizes in strata.

**Value**

Numeric vector with optimal sample allocation in strata.

**Note**

For simple random sampling without replacement design in each stratum, parameters of the variance function  $D$  are  $b = N_1 * S_1^2 + \dots + N_H * S_H^2$ , and  $a_w = N_w * S_w$ , where  $N_w, S_w, w = 1, \dots, H$ , are strata sizes and standard deviations of a study variable in strata respectively.

**References**

Wójciak, W. (2022), Minimum sample size allocation in stratified sampling under constraints on variance and strata sample sizes, doi: [10.48550/arXiv.2204.04035](https://doi.org/10.48550/arXiv.2204.04035)

**See Also**

[rna\\_onesided\(\)](#), [dopt\(\)](#).

**Examples**

```
a <- c(3000, 4000, 5000, 2000)
M <- c(100, 90, 70, 80)
nopt(1017579, a = a, b = 579, M = M)
```

---

pop10\_mM

*Example Population with 10 Strata and Lower and Upper Bounds*

---

**Description**

Artificial example population with 10 strata defined by strata sizes and standard deviations of some study variable. Additionally, the lower and upper bounds have been specified for the sample in strata.

**Usage**

pop10\_mM

**Format**

A matrix with 10 rows and 5 columns: N (strata sizes), S (standard deviations of a variable under study in strata), m (lower bounds), M (upper bounds).

---

pop507

*Example Population with 507 Strata*

---

**Description**

Artificial example population with 507 strata defined by strata sizes and standard deviations of some study variable.

**Usage**

pop507

**Format**

A matrix with 507 rows and 2 columns: N with strata sizes and S standard deviations of a variable under study in strata.

pop969

*Example Population with 969 Strata***Description**

Artificial example population with 969 strata defined by strata sizes and standard deviations of some study variable.

**Usage**

pop969

**Format**

A matrix with 969 rows and 2 columns: N with strata sizes and S standard deviations of a variable under study in strata.

rnabox

*Recursive Neyman Algorithm for Optimal Sample Allocation Under Box-Constraints in Stratified Sampling Schemes***Description****[Stable]**

An internal function that implements recursive Neyman optimal allocation algorithm, rnabox. The rnabox() should not be used directly. Instead, user function `dopt()` should be used.

The rnabox() function computes:

$$\operatorname{argmin} D(x_1, \dots, x_H) = a_1^2/x_1 + \dots + a_H^2/x_H - b,$$

under the equality constraint imposed on total sample size:

$$x_1 + \dots + x_H = n,$$

and (optionally) the following set of inequality constraints:

$$m_w \leq x_w \leq M_w, w = 1, \dots, H.$$

Here,  $H$  denotes total number of strata,  $x_1, \dots, x_H$  are the strata sample sizes, and  $m_w > 0, M_w > 0, w = 1, \dots, H$  are the lower and upper bounds respectively, optionally imposed on sample sizes in strata.

User of rnabox() can choose which or whether the inequality constraints will be added to the optimization problem or not. In case of no inequality constraints to be added,  $m$  or  $M$  (or both) must be specified as NULL (default). If any bounds should be imposed on sample strata sizes, user must specify these with  $m$  and  $M$  arguments.

**Usage**

```
rnabox(n, a, m = NULL, M = NULL)
```

**Arguments**

**n** (number)  
total sample size. A strictly positive scalar.

**a** (numeric)  
parameters  $a_1, \dots, a_H$  of variance function  $D$ . Strictly positive numbers.

**m** (numeric or NULL)  
optional lower bounds constraints imposed on strata sample sizes. If it is not NULL, it is required that  $n \geq \text{sum}(m)$ . Strictly positive numbers.

**M** (numeric or NULL)  
optional upper bounds constraints imposed on strata sample sizes. If it is not NULL, it is required that  $n \leq \text{sum}(M)$ . Strictly positive numbers.

**Value**

Numeric vector with optimal sample allocations in strata.

**Note**

For simple random sampling without replacement design in each stratum, parameters of the variance function  $D$  are  $b = N_1 * S_1^2 + \dots + N_H * S_H^2$ , and  $a_w = N_w * S_w$ , where  $N_w, S_w, w = 1, \dots, H$ , are strata sizes and standard deviations of a study variable in strata respectively.

The `rnabox` is a generalization of a popular recursive Neyman allocation procedure that is described in Remark 12.7.1 in Sarndal et al. (1992). It is a procedure of optimal sample allocation in stratified sampling scheme with simple random sampling without replacement design in each stratum while not exceeding strata sizes.

Note that in case when no inequality constraints are added, the allocation is given as a closed form expression, known as Neyman allocation:

$$x_w = a_w * n / (a_1 + \dots + a_H), w = 1, \dots, H.$$

**References**

To be added soon.

**See Also**

[dopt\(\)](#), [nopt\(\)](#), [sga\(\)](#), [sgaplus\(\)](#), [coma\(\)](#).

**Examples**

```

# Artificial population and the bounds that to be imposed.
N <- c(454, 10, 116, 2500, 2240, 260, 39, 3000, 2500, 400)
S <- c(0.9, 5000, 32, 0.1, 3, 5, 300, 13, 20, 7)
a <- N * S
m <- c(322, 3, 57, 207, 715, 121, 9, 1246, 1095, 294)
M <- N

# An example of a regular allocation.
n <- 6000
opt_regular <- rnabox(n, a, m, M)

# An example of a vertex allocation.
n <- 4076
opt_vertex <- rnabox(n, a, m, M)

```

rna\_onesided

*Recursive Neyman Algorithm for Optimal Sample Allocation in Stratified Sampling Schemes Under One-Sided Constraints*

**Description****[Stable]**

An internal function that implements the recursive Neyman optimal allocation algorithms, `rna` and `lrna`, described in Wesolowski et al. (2021) and Wójciak (2022) respectively. The `rna_onesided()` should not be used directly. Instead, user function `dopt()` or `nopt()` should be used.

The `rna_onesided()` function computes:

$$\operatorname{argmin} D(x_1, \dots, x_H) = a_1^2/x_1 + \dots + a_H^2/x_H - b,$$

under the equality constraint imposed on total sample size:

$$x_1 + \dots + x_H = n,$$

and optionally, under either

$$x_w \leq M_w, w = 1, \dots, H, (\text{Case I})$$

or

$$x_w \geq m_w, w = 1, \dots, H, (\text{Case II})$$

where  $H$  denotes total number of strata,  $x_1, \dots, x_H$  are the strata sample sizes, and  $m_w > 0$  and  $M_w > 0, w = 1, \dots, H$  are lower and upper bounds respectively, optionally imposed on samples sizes in strata.

User of `rna_onesided()` can choose whether the inequality constraints will be added to the optimization problem or not. This is achieved with the proper use of bounds and upper arguments of the function. In case of no inequality constraints to be added, bounds must be specified as `NULL`

(default). If any bounds should be imposed on sample strata sizes, user must specify these with bounds argument. For the Case I of the upper bounds, upper flag must be set to TRUE (default) and then the rna\_onesided() performs the rna. For the Case II of lower bounds, upper flag must be set to FALSE and then the rna\_onesided() performs the lrna algorithm. The upper flag is ignored when bounds is NULL.

### Usage

```
rna_onesided(n, a, bounds = NULL, upper = TRUE, assignments = FALSE)
```

### Arguments

n	(number) total sample size. A strictly positive scalar.
a	(numeric) parameters $a_1, \dots, a_H$ of variance function $D$ . Strictly positive numbers.
bounds	(numeric or NULL) optional one-sided lower or upper bounds constraints imposed on strata sample sizes. If bounds is not NULL, it is required that $n \leq \text{sum}(\text{bounds})$ in case of upper = TRUE, and $n \geq \text{sum}(\text{bounds})$ , in case of upper = FALSE. Strictly positive numbers.
upper	(flag) should values of bounds be treated as one-sided upper bounds constraints (default)? Otherwise, they are treated as lower bounds.
assignments	(flag) should information about strata assignments (either to take-Neyman or take-bound) be added to the output?

### Value

Numeric vector with optimal sample allocations in strata or list with optimal sample allocations in strata and strata assignments (if assignments is TRUE).

### Note

For simple random sampling without replacement design in each stratum, parameters of the variance function  $D$  are  $b = N_1 * S_1^2 + \dots + N_H * S_H^2$ , and  $a_w = N_w * S_w$ , where  $N_w, S_w, w = 1, \dots, H$ , are strata sizes and standard deviations of a study variable in strata respectively.

The rna and lrna are kind of more general versions of a popular recursive Neyman allocation procedure that is described in Remark 12.7.1 in Sarndal et al. (1992). It is a procedure of optimal sample allocation in stratified sampling scheme with simple random sampling without replacement design in each stratum while not exceeding strata sizes.

Note that in case no inequality constraints are added, the allocation is given as a closed form expression, known as Neyman allocation:

$$x_w = a_w * n / (a_1 + \dots + a_H), w = 1, \dots, H.$$



## References

Wesołowski, J., Wieczorkowski, R., Wójciak, W. (2021), Optimality of the recursive Neyman allocation, *Journal of Survey Statistics and Methodology*, doi: [10.1093/jssam/smab018](https://doi.org/10.1093/jssam/smab018), doi: [10.48550/arXiv.2105.14486](https://doi.org/10.48550/arXiv.2105.14486)

Wójciak, W. (2022), Minimum sample size allocation in stratified sampling under constraints on variance and strata sample sizes, doi: [10.48550/arXiv.2204.04035](https://doi.org/10.48550/arXiv.2204.04035)

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

## See Also

[dopt\(\)](#), [nopt\(\)](#), [sga\(\)](#), [sgaplus\(\)](#), [coma\(\)](#), [rnabox\(\)](#).

## Examples

```
a <- c(3000, 4000, 5000, 2000)
bounds <- c(100, 90, 70, 80)
rna_onesided(n = 190, a = a, bounds = bounds)
```

---

var\_tst

*Variance of Stratified Pi-estimator of the Total*

---

## Description

**[Stable]**

Compute the variance of the stratified pi-estimator of the population total, that is of the following generic form:

$$D(x_1, \dots, x_H) = a_1^2/x_1 + \dots + a_H^2/x_H - b,$$

where  $H$  denotes total number of strata,  $x_1, \dots, x_H$  are the strata sample sizes, and  $b, a_w > 0$  do not depend on  $x_w, w = 1, \dots, H$ .

## Usage

```
var_tst(x, a, b)
```

```
var_tst_si(x, N, S)
```

**Arguments**

x	(numeric) sample allocations in strata. Strictly positive numbers.
a	(numeric) parameters $a_1, \dots, a_H$ of variance function $D$ . Strictly positive numbers.
b	(numeric) parameter $b$ of variance function $D$ .
N	(numeric) strata sizes. Strictly positive numbers.
S	(numeric) strata standard deviations of a given study variable. Strictly positive numbers.

**Value**

Value of the variance  $D$  for a given allocation vector  $x$ .

**Functions**

- `var_tst_si`: computes variance of stratified pi-estimator of the total for simple random sampling without replacement design in each stratum. Under this design, parameters of the variance function  $D$  take the following form:

$$a_w = N_w * S_w, w = 1, \dots, H,$$

and

$$b = N_1 * S_1^2 + \dots + N_H * S_H^2,$$

where  $N_w, S_w, w = 1, \dots, H$ , are strata sizes and standard deviations of a study variable in strata respectively.

**References**

Sarndal, C.-E., Swensson, B., and Wretman, J. (1992), *Model Assisted Survey Sampling*, Chapter 3.7 *Stratified Sampling*, New York, NY: Springer.

**Examples**

```
N <- c(3000, 4000, 5000, 2000)
S <- rep(1, 4)
M <- c(100, 90, 70, 80)
opt <- dopt(n = 190, a = N * S, M = M)
var_tst_si(x = opt, N, S)
```

# Index

- \* **datasets**
  - pop10\_mM, [12](#)
  - pop507, [12](#)
  - pop969, [13](#)
- \* **package**
  - stratallo-package, [2](#)
- allocation\_summary, [3](#)
- base::which(), [9](#)
- coma (dopt\_upper), [7](#)
- coma(), [5](#), [6](#), [14](#), [17](#)
- data.frame, [3](#)
- dopt, [4](#)
- dopt(), [3](#), [7](#), [9](#), [11](#), [13–15](#), [17](#)
- dopt\_upper, [7](#)
- h\_get\_which\_violates, [9](#)
- nopt, [10](#)
- nopt(), [3](#), [6](#), [14](#), [15](#), [17](#)
- pop10\_mM, [12](#)
- pop507, [12](#)
- pop969, [13](#)
- rna\_onesided, [15](#)
- rna\_onesided(), [5](#), [6](#), [9](#), [11](#)
- rnabox, [13](#)
- rnabox(), [5](#), [6](#), [17](#)
- sga (dopt\_upper), [7](#)
- sga(), [5](#), [6](#), [14](#), [17](#)
- sgaplus (dopt\_upper), [7](#)
- sgaplus(), [5](#), [6](#), [14](#), [17](#)
- stratallo (stratallo-package), [2](#)
- stratallo-package, [2](#)
- var\_tst, [17](#)
- var\_tst\_si (var\_tst), [17](#)