Package: inca (via r-universe)

October 25, 2026

Type Package

Title Integer Calibration

Version 0.1.0

Date 2025-05-28

Maintainer Luca Sartore <drwolf85@gmail.com>

Description Specific functions are provided for rounding real weights to integers and performing an integer programming algorithm for calibration problems. These functions are useful for census-weights adjustments, survey calibration, or for performing linear regression with integer parameters <https://www.nass.usda.gov/Education_and_Outreach/Reports,_Presentations_and_Conferences/reports/New_Integer_Cal This research was supported in part by the U.S. Department of Agriculture, National Agriculture Statistics Service. The findings and conclusions in this publication are those of the authors and should not be construed to represent any official USDA, or US Government determination or policy.

License GPL (>= 2)

Depends R (>= 3.5.0), stats, Matrix

Imports Rcpp (>= 0.12.1)

Suggests survey

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation yes

Author Luca Sartore [aut] (ORCID = ``0000-0002-0446-1328"), Luca Sartore [cre] (ORCID = ``0000-0002-0446-1328"), Kelly Toppin [aut]

Repository https://cran.r-universe.dev

Date/Publication 2025-05-28 05:20:02 UTC

RemoteUrl https://github.com/cran/inca

RemoteRef HEAD

RemoteSha 9a93a60ec552b239f9584017e47c14bfe08623d8

Contents

inca-package 2
adjWeights 3
intcalibrate 4
roundWeights 7

Index **9**

`inca-package` *inca: Integer Calibration*

Description

Specific functions are provided for rounding real weights to integers and performing an integer programming algorithm for calibration problems. These functions are useful for census-weights adjustments, survey calibration, or for performing linear regression with integer parameters https://www.nass.usda.gov/Education_and_Outreach/Reports,_Presentations_and_Conferences/reports/New_Integer_Calibration_%20Procedure_2016.pdf. This research was supported in part by the U.S. Department of Agriculture, National Agriculture Statistics Service. The findings and conclusions in this publication are those of the authors and should not be construed to represent any official USDA, or US Government determination or policy.

Specific functions are provided for rounding real weights to integers and performing integer programming algorithms for calibration problems.

Details

Package:	inca
Type:	Package
Version:	0.1.0
Date:	2025-05-28
License:	GPL (>= 2)

Calibration forces the weighted estimates of calibration variables to match known totals. This improves the quality of the design-weighted estimates. It is used to adjust for non-response and/or under-coverage. The commonly used methods of calibration produce non-integer weights. In cases where weighted estimates must be integers, one must "integerize" the calibrated weights. However, this procedure often produces final weights that are very different for the "sample" weights. To counter this problem, the **inca** package provides specific functions for rounding real weights to integers, and performing an integer programming algorithm for calibration problems with integer weights.

For a complete list of exported functions, use `library(help = "inca")`.

This research was supported in part by the U.S. Department of Agriculture, National Agriculture Statistics Service. The findings and conclusions in this publication are those of the authors and should not be construed to represent any official USDA or U.S. Government determination or policy.

Author(s)

Maintainer: Luca Sartore <drwolf85@gmail.com> (ORCID = "0000-0002-0446-1328")

Authors:

- Luca Sartore <luca.sartore@usda.gov> (ORCID = "0000-0002-0446-1328")
- Kelly Toppin <kelly.toppin@nass.usda.gov>

Luca Sartore <luca.sartore@usda.gov> and Kelly Toppin <kelly.toppin@nass.usda.gov>

Maintainer: Luca Sartore <drwolf85@gmail.com>

References

Theberge, A. (1999). Extensions of calibration estimators in survey sampling. *Journal of the American Statistical Association*, **94**(446), 635-644.

Little, R. J., & Vartivarian, S. (2003). On weighting the rates in non-response weights.

Kish, L. (1992). Weighting for unequal Pi. *Journal of Official Statistics*, **8**(2), 183.

Rao, J. N. K., & Singh, A. C. (1997). A ridge-shrinkage method for range-restricted weight calibration in survey sampling. *In Proceedings of the section on survey research methods* (pp. 57-65). American Statistical Association Washington, DC.

Horvitz, D. G., & Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, **47**(260), 663-685.

Kalton, G., & Flores-Cervantes, I. (2003). Weighting methods. *Journal of Official Statistics*, **19**(2), 81-98.

Sartore, L., Toppin, K., Young, L., Spiegelman, C. (2019). Developing integer calibration weights for the Census of Agriculture. *Journal of Agricultural, Biological, and Environmental Statistics*, **24**(1), 26-48.

Examples

```
library(inca)
```

---

adjWeights *Function for Weights Adjustments*

Description

This function provides a trimming procedure to force the weights to be within the provided boundaries

Usage

```
adjWeights(weights, lower = -Inf, upper = +Inf)
```

Arguments

`weights`	A numerical vector of weights
`lower`	A numerical vector of lower bounds
`upper`	A numerical vector of upper bounds

Details

The function produces trimmed weights, which will be the input for the rounding technique before integer calibration. When the weights are bounded, the function rounds-up the lower bounds and rounds-down the upper. If the condition `ceiling(lower) > floor(upper)` is true, an error is returned.

Value

A vector of adjusted weights

Examples

```
library(inca)
w <- rnorm(10, 0, 2)
aw <- adjWeights(w, runif(10, -3, -1), runif(10, 1, 3))
hist(aw, main = "Adjusted weights")
```

`intcalibrate` *Integer Calibration Function*

Description

This function performs an integer programming algorithm developed for calibrating integer weights, in order to reduce a specific objective function

Usage

```
intcalibrate(
  weights,
  formula,
  targets,
 objective = c("L1", "aL1", "rL1", "LB1", "rB1", "L2", "aL2", "rL2", "LB2", "rB2", "LC",
    "aLC", "rLC", "SE", "aSE", "rSE"),
  tgtBnds = NULL,
  lower = -Inf,
  upper = Inf,
  scale = NULL,
  sparse = FALSE,
 penalty = c("null", "l0norm", "lasso", "ridge", "raking", "minentropy", "quadrat",
    "quadmod", "hellinger", "mcp", "scad", "relasso", "modrelasso", "rehuber",
```

```
    "modrehuber"),
  tuning = 0,
  data
)
```

Arguments

weights	A numerical vector of real or integer weights to be calibrated. If real values are provided, they will be rounded before applying the calibration algorithm
formula	A formula to express a linear system for hitting the targets
targets	A numerical vector of point-targets to hit
objective	A character specifying the objective function used for calibration. By default "L1". See details for more information
tgtBnds	A two-column matrix containing the bounds for the point-targets
lower	A numerical vector or value defining the lower bounds of the weights
upper	A numerical vector or value defining the upper bounds of the weights
scale	A numerical vector of positive values
sparse	A logical value denoting if the linear system is sparse or not. By default it is FALSE
penalty	A character specifying the penalty function. By default "null" (for backward compatibility). See details for more information
tuning	A positive value denoting the tuning parameter to control the intensity of the penalty function
data	A data.frame or matrix object containing the data to be used for calibration

Details

The integer programming algorithm for calibration can be performed by considering one of the following objective functions:

"L1" for the summation of absolute errors

"aL1" for the asymmetric summation of absolute errors

"rL1" for the summation of absolute relative errors

"LB1" for the summation of absolute errors if outside the boundaries

"rB1" for the summation of absolute relative errors if outside the boundaries

"L2" for the summation of square errors

"aL2" for the asymmetric summation of square errors

"rL2" for the summation of square relative errors

"LB2" for the summation of square errors if outside the boundaries

"rB2" for the summation of square relative errors if outside the boundaries

"LC" for the summation of the logcosh errors

"aLC" for the asymmetric summation of the logcosh errors

"rLC" for the summation of the logcosh relative errors

"SE" for the summation of the exponential absolute errors

"aSE" for the asymmetric summation of the exponential absolute errors

"rSE" for the summation of the exponential absolute relative errors

The calibrated weights can also be restricted further using one of the following penalty functions:

"null" does not penalize, and it is used for backward compatibility

"l0norm" counts the number of non-zero adjustments

"lasso" sums the absolute values of the adjustments

"ridge" sums the adjustments squared

"raking" uses raking ratios

"minentropy" uses the minimum entropy

"quadrat" uses a nomalized euclidean distance

"quadmod" uses a modified normalization in the euclidean distance

"hellinger" uses the Hellinger's distance

"mcp" uses a variation of the minimax concave penalty

"scad" uses a variation of the smoothly clipped absolute deviations

"relasso" sums the absolute value of the relative adjustments

"modrelasso" sums the absolute value of the modified relative adjustments

"rehuber" uses the Huber loss on the relative adjustments

"modrehuber" uses the Huber loss on the modified relative adjustmnets

In particular, the adjustments are considered from the initial rounded weights rather than the input vector of weights with real numbers.

A two-column matrix must be provided to tgtBnds when objective = "aL1", objective = "LB1", objective = "rB1", objective = "aL2", objective = "LB2", objective = "rB2", objective = "aLC", and objective = "aSE"..

The argument scale must be specified with a vector of positive real numbers when objective = "rL1", objective = "rL2", objective = "rLC", or objective = "rSE".

Value

A numerical vector of calibrated integer weights.

Examples

```
library(inca)
set.seed(0)
w <- rpois(10, 4)
data <- matrix(rbinom(1000, 1, .3) * rpois(1000, 4), 100, 10)
y <- data %*% w
w <- runif(10, 0, 7.5)
print(sum(abs(y - data %*% w)))
cw <- intcalibrate(w, ~. + 0, y, lower = 1, upper = 7, sparse = TRUE, data = data)
```

```
print(sum(abs(y - data %*% cw)))
qw <- intcalibrate(w, ~. + 0, y, lower = 1, upper = 7, sparse = TRUE, data = data,
                   penalty = "quadrat", tuning = 0.7)
print(sum(abs(y - data %*% qw)))
barplot(table(cw), main = "Calibrated integer weights")
barplot(table(qw), main = "Calibrated integer weights")
```

`roundWeights` *Function for Rounding Weights*

Description

This function performs an optimal rounding of the provided real weights, in order to reduce a specific objective function

Usage

```
roundWeights(
  weights,
  formula,
  targets,
 objective = c("L1", "aL1", "rL1", "LB1", "rB1", "L2", "aL2", "rL2", "LB2", "rB2", "LC",
    "aLC", "rLC", "SE", "aSE", "rSE"),
  tgtBnds = NULL,
  lower = -Inf,
  upper = Inf,
  scale = NULL,
  sparse = FALSE,
  data
)
```

Arguments

`weights`	A numerical vector of real weights to be rounded
`formula`	A formula to express a linear system for hitting the `targets`
`targets`	A numerical vector of point-targets to hit
`objective`	A character specifying the objective function used for calibration. By default, it is `"L1"`. See details for more information
`tgtBnds`	A two-column matrix containing the bounds for the point-targets
`lower`	A numerical vector or value defining the lower bounds of the weights
`upper`	A numerical vector or value defining the upper bounds of the weights
`scale`	A numerical vector of positive values
`sparse`	A logical value denoting if the linear system is sparse or not. By default, it is `FALSE`
`data`	A `data.frame` or `matrix` object containing the data to be used for calibration

Details

The optimal rounding can be performed by considering one of the following objective functions:

`"L1"` for the summation of absolute errors

`"aL1"` for the asymmetric summation of absolute errors

`"rL1"` for the summation of absolute relative errors

`"LB1"` for the summation of absolute errors if outside the boundaries

`"rB1"` for the summation of absolute relative errors if outside the boundaries

`"L2"` for the summation of square errors

`"aL2"` for the asymmetric summation of square errors

`"rL2"` for the summation of square relative errors

`"LB2"` for the summation of square errors if outside the boundaries

`"rB2"` for the summation of square relative errors if outside the boundaries

`"LC"` for the summation of the logcosh errors

`"aLC"` for the asymmetric summation of the logcosh errors

`"rLC"` for the summation of the logcosh relative errors

`"SE"` for the summation of the exponential absolute errors

`"aSE"` for the asymmetric summation of the exponential absolute errors

`"rSE"` for the summation of the exponential absolute relative errors

A two-column matrix must be provided to `tgtBnds` when `objective = "aL1"`, `objective = "LB1"`, `objective = "rB1"`, `objective = "aL2"`, `objective = "LB2"`, `objective = "rB2"`, `objective = "aLC"`, and `objective = "aSE"`.

The argument `scale` must be specified with a vector of positive real numbers when `objective = "rL1"`, `objective = "rL2"`, `objective = "rLC"`, or `objective = "rSE"`.

Value

A vector of integer weights to be the input of the calibration algorithm

Examples

```
library(inca)
set.seed(0)
w <- rpois(10, 4)
data <- matrix(rbinom(1000, 1, .3) * rpois(1000, 4), 100, 10)
y <- data %*% w
w <- runif(10, 0, 7.5)
rw <- roundWeights(w, ~. + 0, y, lower = 1, upper = 7, sparse = TRUE, data = data)
barplot(table(rw), main = "Rounded weigths")
```

Index

∗ **calibration**
inca-package, 2
∗ **integer**
inca-package, 2
∗ **rounding**
inca-package, 2

adjWeights, 3

inca *(*inca-package*)*, 2
inca-package, 2
intcalibrate, 4

roundWeights, 7