Package ‘icarus’

August 20, 2019

**Title**  Calibrates and Reweights Units in Samples

**Description**  Provides user-friendly tools for calibration in survey sampling.

The package is production-oriented, and its interface is inspired by the famous popular macro 'Calmar' for SAS, so that 'Calmar' users can quickly get used to 'icarus'. In addition to calibration (with linear, raking and logit methods), 'icarus' features functions for calibration on tight bounds and penalized calibration.

**Version**  0.3.1

**Maintainer**  Antoine Rebecq <antoine.rebecq@m4x.org>

**Depends**  R (>= 3.1.1)

**License**  GPL-3

**LazyData**  true

**Suggests**  testthat, ggplot2, Rglpk, slam, xtable, magrittr

**RoxygenNote**  6.1.1

**NeedsCompilation**  no

**Author**  Antoine Rebecq [aut, cre]

**Repository**  CRAN

**Date/Publication**  2019-08-20 05:00:02 UTC

**R topics documented:**

- addMargin .................................................. 2
- calibration .................................................. 2
- calibrationMarginStats ................................. 5
- calWeights_movies ........................................ 6
- colToDummies ................................................. 6
- dataPop .......................................................... 7
- data_employees ............................................ 7
- HTmean ........................................................ 8
- HTtotal ....................................................... 9
- marginStats ................................................ 9
- newMarginMatrix .......................................... 10
addMargin

Adds a margin to marginMatrix

Description

Adds a margin to marginMatrix

Usage

addMargin(marginMatrix, varName, vecTotals, adjustToOne = TRUE, thresholdAdjustToOne = 0.01)

Arguments

marginMatrix The matrix of margins to add the new margin to
varName Name of variable in calibration matrix corresponding to the new margin
vecTotals values of margins (Calmar style) for the variable. Note: if length(vecTotals) > 1, then sum(thresholdAdjustToOne) has to be 1.
adjustToOne if TRUE and sum(vecTotals) is nearly 1, modify values of vecTotals so that sum is 1.
thresholdAdjustToOne adjust sum(vecTotals) to 1 if difference is under thresholdAdjustToOne

calibration

Calibration on margins

Description

Performs calibration on margins with several methods and customizable parameters
Usage

calibration(data, marginMatrix, colWeights, method = "linear", 
  bounds = NULL, q = NULL, costs = NULL, gap = NULL, 
  popTotal = NULL, pct = FALSE, scale = NULL, description = TRUE, 
  maxIter = 2500, check = TRUE, calibTolerance = 1e-06, 
  uCostPenalized = 1, lambda = NULL, precisionBounds = 1e-04, 
  forceSimplex = FALSE, forceBisection = FALSE, colCalibratedWeights, 
  exportDistributionImage = NULL, exportDistributionTable = NULL)

Arguments

data The dataframe containing the survey data
marginMatrix The matrix giving the margins for each column variable included in the calibration problem
colWeights The name of the column containing the initial weights in the survey dataframe
method The method used to calibrate. Can be "linear", "raking", "logit", "truncated"
bounds Two-element vector containing the lower and upper bounds for bounded methods ("truncated" and "logit")
q Vector of q_k weights described in Deville and Sarndal (1992)
costs The penalized calibration method will be used, using costs defined by this vector. Must match the number of rows of marginMatrix. Negative of non-finite costs are given an infinite cost (coefficient of C^-1 matrix is 0)
gap Only useful for penalized calibration. Sets the maximum gap between max and min calibrated weights / initial weights ratio (and thus is similar to the "bounds" parameter used in regular calibration)
popTotal Precise the total population if margins are defined by relative value in marginMatrix (percentages)
pct If TRUE, margins for categorical variables are considered to be entered as percentages. popTotal must then be set. (FALSE by default)
scale If TRUE, stats (including bounds) on ratio calibrated weights / initial weights are done on a vector multiplied by the weighted non-response ratio (ratio population total / total of initial weights). Has same behavior as "ECHELLE=0" in Calmar.
description If TRUE, output stats about the calibration process as well as the graph of the density of the ratio calibrated weights / initial weights
maxIter The maximum number of iterations before stopping
check performs a few check about the dataframe. TRUE by default
calibTolerance Tolerance for the distance to an exact solution. Could be useful when there is a huge number of margins as the risk of inadvertently setting incompatible constraints is higher. Set to 1e-06 by default.
uCostPenalized Unary cost by which every cost is "costs" column is multiplied
lambda The initial ridge lambda used in penalized calibration. By default, the initial lambda is automatically chosen by the algorithm, but you can speed up the search for the optimum if you already know a lambda close to the lambda_opt
corresponding to the gap you set. Be careful, the search zone is reduced when a
lambda is set by the user, so the program may not converge if the lambda set is
too far from the lambda_opt.

precisionBounds
Only used for calibration on minimum bounds. Desired precision for lower and
upper reweighting factor, both bounds being as close to 1 as possible

forceSimplex
Only used for calibration on tight bounds. Bisection algorithm is used for matrices
whose size exceed 1e8. forceSimplex = TRUE forces the use of the simplex
algorithm whatever the size of the problem (you might want to set this parameter
to TRUE if you have a large memory size)

forceBisection
Only used for calibration on tight bounds. Forces the use of the bisection algo-
rithm to solve calibration on tight bounds

colCalibratedWeights
Deprecated. Only used in the scope of calibration function

exportDistributionImage
File name to which the density plot shown when description is TRUE is ex-
ported. Requires package "ggplot2"

exportDistributionTable
File name to which the distribution table of before/after weights shown when
description is TRUE is exported. Requires package "xtable"

Value
column containing the final calibrated weights

References
Vanderhoeft, Camille. Generalised calibration at statistics Belgium: SPSS Module G-CALIB-S and
Le Guennec, Josiane, and Olivier Sautory. "Calmar 2: Une nouvelle version de la macro calmar
der redressement d’échantillon par calage." Journees de Methodologie Statistique, Paris. INSEE
(2002).

Examples
N <- 300 ## population total
## Horvitz Thompson estimator of the mean: 1.666667
weightedMean(data_employees$movies, data_employees$weight, N)
## Enter calibration margins:
mar1 <- c("category",3,80,90,60)
mar2 <- c("sex",2,140,90,0)
mar3 <- c("department",2,100,130,0)
mar4 <- c("salary", 0, 470000,0,0)
margins <- rbind(mar1, mar2, mar3, mar4)
## Compute calibrated weights with raking ratio method
calibrationMarginStats

wCal <- calibration(data=data_employees, marginMatrix=margins, colWeights="weight", method="raking", description=FALSE)
## Calibrated estimate: 2.471917
weightedMean(data_employees$movies, wCal, N)

calibrationMarginStats

Stats for initial weights, calibrated weights, and margins.

Description

Gives stats about the calibration process: differences between totals after/before calibration and margins. Totals for categorical variables are displayed in percentages. (same as first panels output in Calmar/Calmar 2) Output is a list, which might not be convenient for exports (e.g. for integration into a scientific report). In such cases, use function marginStats, which outputs a dataframe.

Usage

calibrationMarginStats(data, marginMatrix, popTotal = NULL, pct = FALSE, colWeights, colCalibratedWeights = NULL, calibThreshold = 1)

Arguments

data        dataframe containing the survey data
marginMatrix  matrix of margins
popTotal     total of population, useful if margins are entered in relative value
pct          Set this to true if margins for categorical variables are written in percentages
colWeights   name of weights column in the dataframe
colCalibratedWeights  name of calibrated weights column in the dataframe (if applicable)
calibThreshold If difference between calibration estimate and margin differ more than this parameter, calibration is considered to have failed

Value

List containing stats on weights and margins

See Also

marginStats
calWeights_movies  
*Calibration weights for data_employees*

Description

Calibration weights computed with Calmar2 for the small example data_employees.

Usage

calWeights_movies

Format

1 column "id", unique id for each of the 15 units in sample. 3 columns with calibration weights using 3 different methods (linear, raking, and logit with bounds LO=0.4, UP=2.2)

Author(s)

Antoine Rebecq

colToDummies  
*Changes a column containing multiple values to a matrix of columns containing the dummies corresponding to each value.*

Description

Changes a column containing multiple values to a matrix of columns containing the dummies corresponding to each value.

Usage

colToDummies(col, nameCol, modalities = NULL, keepValue = FALSE)

Arguments

col  
input column

nameCol  
name that will be used as a prefix for dummies column name in the output matrix

modalities  
if a vector is entered, dummies produced will only be the ones corresponding to the values in the "modalities" input column + another one containing all the other modalities.

keepValue  
Logical. If TRUE, puts not "1"s in the dummies output columns but the real values in the "col" column (except if values are non-numeric)

Value

Matrix containing the dummy columns
**dataPop**

Test population for Icarus.

**Description**

This data set features a generated population of 50000 units. 11 characteristics of interest for all units in population are featured. These characteristics of interest are variously correlated to one another. A stratified random sampling (with a proportional allocation on variable Y3) of fixed size 1000 is selected. Among the 1000 units in the selected sample, only 718 are respondent to the survey. These responding units are selected using a dummy logit model.

**Usage**

`dataPop`

**Format**

1 column "ident" with unique id for all units. 11 columns with various characteristics of interest for units in the population. 1 column "weight", with sampling weights . Weights equal to zero means that the unit is not selected in the sample. 1 column "simul_nr" indicates the probability that each unit will respond to the survey. 1 column "responding". For sampled units, indicates whether unit is respondent to survey (1) or not (0). Variable is also equal to 0 for units not selected in sample 1 column "qTest" containing randomly generated q weights used in unit tests 50000 rows, 1 row per unit in the population.

**Author(s)**

Antoine Rebecq

**References**


---

**data_employees**

A small example sample for calibration with Icarus

**Description**

This table features a samples of 15 units (drawn from a population of size 300), used in a small survey to determine how frequently the employees of a firm go the movies (column "cinema"). Some auxiliary variables are given, which allows the use of calibration to improve estimates. Margins for these auxiliary variables are known: categ: 80 (modality 1) ; 90 (modality 2) ; 60 (modality 3) sexe: 140 (modality 1) ; 90 (modality 2) service: 100 (modality 1) ; 130 (modality 2) salaire : 470000
**Usage**

data_employees

**Format**

15 rows, one per unit in sample. 1 column "id", unique id for each unit. 4 columns of auxiliary variables ("service", "categ", "sexe", "salaire"). 1 column "cinema" - the variable of interest 1 column "weight" - the Horvitz-Thompson weights

**Author(s)**

Antoine Rebecq

---

**HTmean**

Weighted estimator for the mean

**Description**

Computes the weighted estimator for the mean of a column. Alias for weightedMean

**Usage**

HTmean(var, weights, popTot = NULL)

**Arguments**

- **var**: column of variable of interest
- **weights**: column of weights matching the variable of interest
- **popTot**: population size, used in Horvitz-Thompson-like estimation. If no value is given for popTot, default value is the sum of weights. In the context of survey sampling, this is equivalent to using an Hajek estimate.

**Value**

Estimated mean

**See Also**

weightedMean
HTtotal  

Weighted estimator for total

Description

Computes the weighted estimator for the total of a column. Alias for weightedTotal.

Usage

HTtotal(var, weights)

Arguments

<table>
<thead>
<tr>
<th>var</th>
<th>column of variable of interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>weights</td>
<td>column of weights matching the variable of interest</td>
</tr>
</tbody>
</table>

Value

Estimated total

See Also

weightedTotal

marginStats  

Stats for initial weights, calibrated weights, and margins.

Description

Just like calibrationMarginStats, gives stats about the calibration process: differences between totals after/before calibration and margins. Totals for categorical variables are displayed in percentages. The last column, named "difference", shows the difference (in percentage points) between initial estimates and margins (if colCalibratedWeights is NULL) or between calibrated estimates and margins (if colCalibratedWeights is not NULL). Output is a dataframe, which might be more convenient to export than a list (e.g. for integration into reports).

Usage

marginStats(data, marginMatrix, pct = FALSE, popTotal = NULL, colWeights, colCalibratedWeights = NULL, calibThreshold = 1)
Arguments

- **data**: dataframe containing the survey data
- **marginMatrix**: matrix of margins
- **pct**: Set this to true if margins for categorical variables are written in percentages
- **popTotal**: total of population, useful if margins are entered in relative value
- **colWeights**: name of weights column in the dataframe
- **colCalibratedWeights**: name of calibrated weights column in the dataframe (if applicable)
- **calibThreshold**: If difference between calibration estimate and margin differ more than this parameter, calibration is considered to have failed

Value

Dataframe containing stats on weights and margins

See Also

calibrationMarginStats

---

newMarginMatrix

*Create empty margin matrix*

Description

Use this to create an empty margin matrix (which facilitates the use of magrittr syntax to enter margins)

Usage

```r
newMarginMatrix()
```

Examples

```r
library(magrittr)
N <- 230 ## population total
## Horvitz Thompson estimator of the mean: 2.174
weightedMean(data_employees$movies, data_employees$weight, N)
## Enter calibration margins:
margins <- newMarginMatrix() %>%
  addMargin("category", c(0.35, 0.40, 0.25)) %>%
  addMargin("sex", c(0.6, 0.4)) %>%
  addMargin("department", c(0.45, 0.55)) %>%
  addMargin("salary", 470000)
## Compute calibrated weights with raking ratio method
wCal <- calibration(data=data_employees, marginMatrix=margins, colWeights="weight",
                    method="raking", pct = TRUE, description=FALSE
```
### poptest_calmar

Calibration on population test - made on Calmar2

**Description**

This data set features calibration weights for the sample test of `dataPop` (using margins tables `table_margins_1` and `table_margins_2`). Calibration is computed using the SAS Macro Calmar2, for test purposes.

**Usage**

`poptest_calmar`

**Format**

1000 rows, one per unit in the sample. 1 column "ident", with a unique id for every unit in the sample. 3 methods of calibration are used (linear, raking, and logit with bounds LO=0.2 and UP=1.3) for two different margins tables `table_margins_1` and `table_margins_2`, which results in 7 columns of weights.

**Author(s)**

Antoine Rebecq

**References**


### poptest_calmar_nr

Calibration with nonresponse on population test - made on Calmar2

**Description**

This data set features calibration weights for the sample test of `dataPop` (using margins tables `table_margins_1` and `table_margins_2`). Calibration is computed using the SAS Macro Calmar2, for test purposes. Only the 718 responding units are taken into account.

**Usage**

`poptest_calmar_nr`
regroupCalibrationModalities

Format

718 rows, one per unit in the sample. 1 column "ident", with a unique id for every unit in the sample.
3 methods of calibration are used (linear, raking, and logit with bounds LO=0.1 and UP=2.0 and parameter ECHELLE=0) for two different margins tables `table_margins_1` and `table_margins_2`, which results in 7 columns of weights.

Author(s)

Antoine Rebecq

References


---

regroupCalibrationModalities

Regroup calibration modalities

Description

Beware, this function modifies the calibrationMatrix and marginMatrix objects entered in parameter? Regroups modalities entered in "vecModalities" into single "newModality" in "calibrationMatrix" and adapts "marginMatrix" to the new concept. Typical usage is right before a calibration (and after computation of marginMatrix), when you realise calibration output is better when several modalities are reduced to one. (typically very rare modalities, on which calibration constraints are very restrictive). Uses pseudo-"call by reference" via eval.parent because 2 objects are modified: calibrationMatrix and marginMatrix

Usage

regroupCalibrationModalities(calibrationMatrix, marginMatrix, calibrationVariable, vecModalities, newModality)

Arguments

calibrationMatrix
calibration matrix
marginMatrix matrix containing the margins to the Icarus format
calibrationVariable name of the calibration variable for which regroupment has to be done
vecModalities Initial modalities of the variable
newModality Regrouped modalities of the variable
Examples

```
## Not run:
## Suppose we have a calibration matrix and a margin matrix containing information
## for two categorical variables "X1" (10 modalities) and "X2" (5 modalities)

matrixCal <- data.frame(matrix(
  c(floor(10*runif(100))+1,floor((5)*runif(100))+1,
    floor(10*runif(100))+1,rep(10,100)),
  ncol=4))

marginMatrix <- matrix(c("X1",10,rep(1/10,10),
                         "X2",5,rep(1/5,5),rep(0,5)), nrow=2, byrow=TRUE)

# table(matrixCal$X1)
# 1 2 3 4 5 6 7 8 9 10
# 9 8 8 11 15 13 6 10 12
# marginMatrix
# [1,] "X1" "10" "0.1" "0.1" "0.1" "0.1" "0.1" "0.1" "0.1" "0.1" "0.1" "0.1"
# [2,] "X2" "5" "0.2" "0.2" "0.2" "0.2" "0.2" "0" "0" "0" "0" "0"

regroupCalibrationModalities(matrixCal, marginMatrix, "X1", c(3,4,8), "0")
# table(matrixCal$X1)
# 0 1 2 3 4 5 6 7 8 9 10
# 22 9 8 11 15 13 10 12
# marginMatrix
# [1,] "X1" "8" "0.3" "0.1" "0.1" "0.1" "0.1" "0.1" "0.1" "0.1"
# [2,] "X2" "5" "0.2" "0.2" "0.2" "0.2" "0.2" "0" "0" "0"

## End(Not run)
```

---

**regroupModalities**

**Regroup elements of a vector**

**Description**

Regroup the contiguous elements of a vector under a single value. Which elements should be regrouped is indicated by the rows of a matrix. Output vector is NOT a factor.

**Usage**

```
regroupModalities(column, regroupMatrix, modalities = NULL)
```

**Arguments**

- `column` Column vector which values are going to be replaced
- `regroupMatrix` Bounds of the values to regroup under the same modality
- `modalities` Specify the values of the modalities to use. Must match number of rows of `regroupMatrix` If not specified, replacement modalities will be 1:length(column)
**Value**

Column vector with regrouped modalities

**Examples**

```r
gregrModalities(c(1:20), rbind(c(0,5),c(6,18),c(19,Inf)))
# Returns : [1] 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3
```

---

**Description**

This table features calibration margins for the sample of the test population of `dataPop`.

**Usage**

```r
tableMargins_1
```

**Format**

A margins table written in the Icarus format.

**Author(s)**

Antoine Rebecq

---

**Description**

This table features calibration margins for the sample of the test population of `dataPop`. Margins for categorical variables are entered in percentages.

**Usage**

```r
tableMargins_2
```

**Format**

A margins table written in the Icarus format.

**Author(s)**

Antoine Rebecq
weightedMean

Weighted estimator for the mean

Description

Computes the weighted estimator for the mean of a column

Usage

weightedMean(var, weights, popTot = NULL)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>column of variable of interest</td>
</tr>
<tr>
<td>weights</td>
<td>column of weights matching the variable of interest</td>
</tr>
<tr>
<td>popTot</td>
<td>population size, used in Horvitz-Thompson-like estimation. If no value is given for popTot, default value is the sum of weights. In the context of survey sampling, this is equivalent to using an Hajek estimate.</td>
</tr>
</tbody>
</table>

Value

Estimated mean

See Also

HTmean

weightedTotal

Weighted estimator for total

Description

Computes the weighted estimator for the total of a column

Usage

weightedTotal(var, weights)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>column of variable of interest</td>
</tr>
<tr>
<td>weights</td>
<td>column of weights matching the variable of interest</td>
</tr>
</tbody>
</table>

Value

Estimated total
See Also

HTtotal
Index

addMargin, 2

calibration, 2
calibrationMarginStats, 5, 9, 10
calWeights_movies, 6
colToDummies, 6

data_employees, 6, 7
dataPop, 7, 11, 14

HTmean, 8, 15
HTtotal, 9, 16

marginStats, 5, 9

newMarginMatrix, 10

poptest_calmar, 11
poptest_calmar_nr, 11

regroupCalibrationModalities, 12
regroupModalities, 13

table_margins_1, 11, 12, 14
table_margins_2, 11, 12, 14

weightedMean, 8, 15
weightedTotal, 9, 15