Package ‘SmallCountRounding’

September 29, 2021

Type Package

Title Small Count Rounding of Tabular Data

Version 0.9.0

Date 2021-09-29

Author Øyvind Langsrud [aut, cre],
    Johan Heldal [aut]

Maintainer Øyvind Langsrud <oyl@ssb.no>

Depends Matrix, SSBtools

Imports methods

VignetteBuilder knitr

Suggests knitr, rmarkdown, kableExtra, sdcHierarchies, testthat

Description A statistical disclosure control tool to protect frequency tables in cases where small values are sensitive. The function PLSrounding() performs small count rounding of necessary inner cells so that all small frequencies of cross-classifications to be published (publishable cells) are rounded. This is equivalent to changing micro data since frequencies of unique combinations are changed. Thus, additivity and consistency are guaranteed. The methodology is described in Langsrud and Heldal (2018) <https://www.researchgate.net/publication/327768398_An_Algorithm_for_Small_Count_Rounding_ofTabla

License Apache License 2.0

URL https://github.com/statisticsnorway/SmallCountRounding

BugReports https://github.com/statisticsnorway/SmallCountRounding/issues

RoxygenNote 7.1.2

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2021-09-29 12:00:02 UTC
R topics documented:

SmallCountRounding-package .................................................. 2
HD ................................................................. 2
PLS2way ................................................................. 3
PLSrounding ................................................................. 4
PLSroundingFits ............................................................... 7
print.PLSrounded ............................................................. 9
RoundViaDummy ............................................................. 10
SmallCountData ............................................................... 13

Index 15

SmallCountRounding-package

Small Count Rounding of Tabular Data

Description

A statistical disclosure control tool to protect frequency tables in cases where small values are sensitive. The main function, PLSrounding, performs small count rounding of necessary inner cells (Heldal, 2017) so that all small frequencies of cross-classifications to be published (publishable cells) are rounded. This is equivalent to changing micro data since frequencies of unique combinations are changed. Thus, additivity and consistency are guaranteed. This is performed by an algorithm inspired by partial least squares regression (Langsrud and Heldal, 2018).

References


HD

Hellinger Distance (Utility)

Description

Hellinger distance (HD) and a related utility measure (HDutility) described in the reference below. The utility measure is made to be bounded between 0 and 1.
Usage

HD(f, g)

HDutility(f, g)

Arguments

f Vector of original counts
g Vector of perturbed counts

Details

HD is defined as \( \sqrt{\text{sum}((\sqrt{f} - \sqrt{g})^2)/2} \) and HDutility is defined as \( 1 - \text{HD}(f, g)/\sqrt{\text{sum}(f)} \).

Value

Hellinger distance or related utility measure

References


Examples

f <- 1:6
g <- c(0, 3, 3, 3, 6, 6)
print(c(
    HD = HD(f, g),
    HDutility = HDutility(f, g),
    maxdiff = max(abs(g - f)),
    meanAbsDiff = mean(abs(g - f)),
    rootMeanSquare = sqrt(mean((g - f)^2))
))

Description

Output from PLSrounding is presented as two-way table(s) in cases where this is possible. A requirement is that the number of main dimensional variables is two.

Usage

PLS2way(obj, variable = c("rounded", "original", "difference", "code"))
PLSrounding

Arguments

obj Output object from PLSrounding

variable One of "rounded" (default), "original", "difference" or "code".

Details

When parameter "variable" is "code", output is coded as "#" (publish), "." (inner) and "&" (both).

Value

A data frame

Examples

# Making tables from PLSrounding examples
z <- SmallCountData("e6")
a <- PLSrounding(z, "freq", formula = ~eu * year + geo)
PLS2way(a, "original")
PLS2way(a, "difference")
PLS2way(a, "code")
PLS2way(PLSrounding(z, "freq", formula = ~eu * year + geo * year), "code")
eHrc2 <- list(geo = c("EU", "@Portugal", "@Spain", "Iceland"), year = c("2018", "2019"))
PLS2way(PLSrounding(z, "freq", hierarchies = eHrc2))

Description

Small count rounding of necessary inner cells are performed so that all small frequencies of cross-classifications to be published (publishable cells) are rounded. The publishable cells can be defined from a model formula, hierarchies or automatically from data.

Usage

PLSrunding(
  data,
  freqVar = NULL,
  roundBase = 3,
  hierarchies = NULL,
  formula = NULL,
  dimVar = NULL,
  maxRound = roundBase - 1,
  printInc = nrow(data) > 1000,
  output = NULL,
  preAggregate = is.null(freqVar),
)
PLSrounding

... )

PLSroundingInner(..., output = "inner")

PLSroundingPublish(..., output = "publish")

Arguments

data          Input data as a data frame (inner cells)
dfreqVar      Variable holding counts (inner cells frequencies). When NULL (default), micro-data is assumed.
roundBase     Rounding base
hierarchies   List of hierarchies
formula       Model formula defining publishable cells
dimVar        The main dimensional variables and additional aggregating variables. This parameter can be useful when hierarchies and formula are unspecified.
maxRound      Inner cells contributing to original publishable cells equal to or less than maxRound will be rounded
printInc      Printing iteration information to console when TRUE
output        Possible non-NULL values are "input", "inner" and "publish". Then a single data frame is returned.
preAggregate  When TRUE, the data will be aggregated beforehand within the function by the dimensional variables.
...           Further parameters sent to RoundViaDummy

Details

This function is a user-friendly wrapper for RoundViaDummy with data frame output and with computed summary of the results. See RoundViaDummy for more details.

Value

Output is a four-element list with class attribute "PLSrounded" (to ensure informative printing).

inner          Data frame corresponding to input data with the main dimensional variables and with cell frequencies (original, rounded, difference).
publish        Data frame of publishable data with the main dimensional variables and with cell frequencies (original, rounded, difference).
metrics        A named character vector of various statistics calculated from the two output data frames ("inner_" used to distinguish). See examples below and the function HDutility.
freqTable      Matrix of frequencies of cell frequencies and absolute differences. For example, row "rounded" and column "inn.4+" is the number of rounded inner cell frequencies greater than or equal to 4.
References


See Also

RoundViaDummy, PLS2way, ModelMatrix

Examples

# Small example data set
z <- SmallCountData("e6")
print(z)

# Publishable cells by formula interface
a <- PLSrounding(z, "freq", roundBase = 5, formula = ~geo + eu + year)
predict(a)
predict(a$inner)
predict(a$publish)
predict(a$metrics)
predict(a$freqTable)

# Recalculation of maxdiff, HDutility, meanAbsDiff and rootMeanSquare
max(abs(a$publish[, "difference"]))
HDutility(a$publish[, "original"], a$publish[, "rounded"])
mean(abs(a$publish[, "difference"]))
sqrt(mean((a$publish[, "difference"])^2))

# Six lines below produce equivalent results
# Ordering of rows can be different
PLSrounding(z, "freq") # All variables except "freq" as dimVar
PLSrounding(z, "freq", dimVar = c("geo", "eu", "year"))
PLSrounding(z, "freq", formula = ~eu * year + geo * year)
PLSrounding(z[, -2], "freq", hierarchies = SmallCountData("eHrc"))
PLSrounding(z[, -2], "freq", hierarchies = SmallCountData("eDimList"))

# Single data frame output
PLSroundingInner(z, "freq", roundBase = 5, formula = ~geo + eu + year)
PLSroundingPublish(z, roundBase = 5, formula = ~geo + eu + year)
# Microdata input  
PLSroundingInner(rbind(z, z), roundBase = 5, formula = ~geo + eu + year)

# Parameter avoidHierarchical (see RoundViaDummy and ModelMatrix)  
PLSroundingPublish(z, roundBase = 5, formula = ~geo + eu + year, avoidHierarchical = TRUE)

# Package sdcHierarchies can be used to create hierarchies.  
The small example code below works if this package is available.  
if (require(sdcHierarchies)) {
  z2 <- cbind(geo = c("11", "21", "22"), z[, 3:4], stringsAsFactors = FALSE)
  h2 <- list(  
    geo = hier_compute(inp = unique(z2$geo), dim_spec = c(1, 1), root = "Tot", as = "df"),
    year = hier_convert(hier_create(root = "Total", nodes = c("2018", "2019")), as = "df"))
  PLSrounding(z2, "freq", hierarchies = h2)
}

# Use PLS2way to produce tables as in Langsrud and Heldal (2018) and to demonstrate 
# parameters maxRound, zeroCandidates and identifyNew (see RoundViaDummy).  
# Parameter rndSeed used to ensure same output as in reference.  
exPSD <- SmallCountData("exPSD")  
a <- PLSrounding(exPSD, "freq", 5, formula = ~rows + cols, rndSeed=124)
PLS2way(a, "original")  
  # Table 1
PLS2way(a)  
  # Table 2
a <- PLSrounding(exPSD, "freq", 5, formula = ~rows + cols, identifyNew = FALSE, rndSeed=124)
PLS2way(a)  
  # Table 3
a <- PLSrounding(exPSD, "freq", 5, formula = ~rows + cols, maxRound = 7)
PLS2way(a)  
  # Values in coll rounded
a <- PLSrounding(exPSD, "freq", 5, formula = ~rows + cols, zeroCandidates = TRUE)
PLS2way(a)  
  # (row3, col4): original is 0 and rounded is 5

PLSroundingFits  
Small count rounding with post-processing to expected frequencies

Description

The counts rounded by PLSrounding Thereafter, based on the publishable rounded data, expected inner cell frequencies are generated by iterative proportional fitting using Mipf. To ensure that empty cells missing in input data are included in the fitting process, the data is first extended using Extend0.

Usage

PLSroundingFits(  
data,
freqVar = NULL,
roundBase = 3,
hierarchies = NULL,
formula = NULL,
dimVar = NULL,
preAggregate = is.null(freqVar),
xReturn = FALSE,
extend0 = TRUE,
limit = 1e-10,
viaQR = FALSE,
itern = 1000,
eps = 0.01,
tol = 1e-13,
reduceBy0 = TRUE,
reduceByColSums = TRUE,
reduceByLeverage = FALSE,
...
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>data frame (inner cells)</td>
</tr>
<tr>
<td>freqVar</td>
<td>Variable holding counts</td>
</tr>
<tr>
<td>roundBase</td>
<td>Rounding base</td>
</tr>
<tr>
<td>hierarchies</td>
<td>List of hierarchies</td>
</tr>
<tr>
<td>formula</td>
<td>Model formula</td>
</tr>
<tr>
<td>dimVar</td>
<td>Dimensional variables</td>
</tr>
<tr>
<td>preAggregate</td>
<td>Aggregation</td>
</tr>
<tr>
<td>xReturn</td>
<td>Dummy matrix in output when TRUE. To return crossTable as well, use xReturn = 2.</td>
</tr>
<tr>
<td>extend0</td>
<td>Data is automatically extended by Extend0 when TRUE. Can also be specified as a list meaning parameter varGroups to Extend0.</td>
</tr>
<tr>
<td>limit</td>
<td>LSfitNonNeg parameter</td>
</tr>
<tr>
<td>viaQR</td>
<td>LSfitNonNeg parameter</td>
</tr>
<tr>
<td>iter</td>
<td>Mipf parameter</td>
</tr>
<tr>
<td>eps</td>
<td>Mipf parameter</td>
</tr>
<tr>
<td>tol</td>
<td>Mipf parameter</td>
</tr>
<tr>
<td>reduceBy0</td>
<td>Mipf parameter</td>
</tr>
<tr>
<td>reduceByColSums</td>
<td>Mipf parameter</td>
</tr>
<tr>
<td>reduceByLeverage</td>
<td>Mipf parameter</td>
</tr>
<tr>
<td>...</td>
<td>Further parameters to PLSrounding.</td>
</tr>
</tbody>
</table>

Details

The seven first parameters is documented in more detail in PLSrounding. If iterative proportional fitting succeeds, the maximum difference between rounded counts and ipFit is less than input parameter eps.
Value
Output from `PLSrounding` (class attribute "PLSrounded") with modified versions of `inner` and `publish`:

- **inner**: Extended with more input data variables and with expected frequencies (`ipFit`).
- **publish**: Extended with aggregated expected frequencies (`ipFit`).

Examples
```r
z <- data.frame(geo = c("Iceland", "Portugal", "Spain"),
                 eu = c("nonEU", "EU", "EU"),
                 year = rep(c("2018","2019"), each = 3),
                 freq = c(2,3,7,1,5,6), stringsAsFactors = FALSE)
z4 <- z[-c(1:2), ]
PLSroundingFits(z4, "freq", formula = ~eu * year + geo, extend0 = FALSE)[c("inner", "publish")]
PLSroundingFits(z4, "freq", formula = ~eu * year + geo)[c("inner", "publish")]

my_km2 <- SSBtools::SSBtoolsData("my_km2")
# Default automatic extension (extend0 = TRUE)
PLSroundingFits(my_km2, "freq",
                formula = ~(Sex + Age) * Municipality * Square1000m + Square250m)[c("inner", "publish")]

# Manual specification to avoid Nittedal combined with another_km
PLSroundingFits(my_km2, "freq", formula = ~(Sex + Age) * Municipality * Square1000m + Square250m,
                extend0 = list(c("Sex", "Age"),
                              c("Municipality", "Square1000m", "Square250m")))[c("inner", "publish")]
```

print.PLSrounded

`print` method for `PLSrounded`

Description
Print method for `PLSrounded`

Usage
```r
## S3 method for class 'PLSrounded'
print(x, digits = max(getOption("digits") - 3, 3), ...)
```

Arguments
- `x`: `PLSrounded` object
- `digits`: positive integer. Minimum number of significant digits to be used for printing most numbers.
- `...`: further arguments sent to the underlying
Value

Invisibly returns the original object.

Description

Small count rounding via a dummy matrix and by an algorithm inspired by PLS

Usage

RoundViaDummy(
  data,
  freqVar,
  formula = NULL,
  roundBase = 3,
  singleRandom = FALSE,
  crossTable = TRUE,
  total = "Total",
  maxIterRows = 1000,
  maxIter = 1e+07,
  x = NULL,
  hierarchies = NULL,
  xReturn = FALSE,
  maxRound = roundBase - 1,
  zeroCandidates = FALSE,
  forceInner = FALSE,
  identifyNew = TRUE,
  step = 0,
  preRounded = NULL,
  leverageCheck = FALSE,
  easyCheck = TRUE,
  printInc = TRUE,
  rndSeed = 123,
  dimVar = NULL,
  plsWeights = NULL,
  ...}

Arguments

data  Input data as a data frame (inner cells)
freqVar  Variable holding counts (name or number)
formula  Model formula defining publishable cells. Will be used to calculate x (via ModelMatrix). When NULL, x must be supplied.
roundBase  Rounding base
singleRandom Single random draw when TRUE (instead of algorithm)
crossTable  When TRUE, cross table in output and calculations via FormulaSums()
total  String used to name totals
maxIterRows  See details
maxIter  Maximum number of iterations
x  Dummy matrix defining publishable cells
hierarchies  List of hierarchies, which can be converted by AutoHierarchies. Thus, a single string as hierarchy input is assumed to be a total code. Exceptions are "rowFactor" or "", which correspond to only using the categories in the data.
xReturn  Dummy matrix in output when TRUE (as input parameter x)
maxRound  Inner cells contributing to original publishable cells equal to or less than maxRound will be rounded.
zeroCandidates  When TRUE, inner cells in input with zero count (and multiple of roundBase when maxRound is in use) contributing to publishable cells will be included as candidates to obtain roundBase value. With vector input, the rule is specified individually for each cell.
forceInner  When TRUE, all inner cells will be rounded. Use vector input to force individual cells to be rounded. Can be combined with parameter zeroCandidates to allow zeros and roundBase multiples to be rounded up.
identifyNew  When TRUE, new cells may be identified after initial rounding to ensure that no nonzero rounded publishable cells are less than roundBase.
step  When step>1, the original forward part of the algorithm is replaced by a kind of stepwise. After step steps forward, backward steps may be performed. The step parameter is also used for backward-forward iteration at the end of the algorithm; step backward steps may be performed.
preRounded  A vector or a variable in data that contains a mixture of missing values and predetermined values of rounded inner cells.
leverageCheck  When TRUE, all inner cells that depends linearly on the published cells and with small frequencies (<=maxRound) will be rounded. The computation of leverages can be very time and memory consuming. The function Reduce0exact is called. The default leverage limit is 0.999999. Another limit can be sent as input instead of TRUE. Checking is performed before and after (since new zeros) rounding. Extra iterations are performed when needed.
easyCheck  A light version of the above leverage checking. Checking is performed after rounding. Extra iterations are performed when needed. Reduce0exact is called with reduceByLeverage=FALSE and reduceByColSums=TRUE.
printInc  Printing iteration information to console when TRUE
rndSeed  If non-NULL, a random generator seed to be used locally within the function without affecting the random value stream in R.
dimVar  The main dimensional variables and additional aggregating variables. This parameter can be useful when hierarchies and formula are unspecified.
plsWeights  A vector of weights for each cell to be published or a function generating it. For use in the algorithm criterion. The supplied function takes the following arguments: yPublish, yInner, crossTable, x, roundBase, maxBase, and ..., where the first two are numeric vectors of original counts.

Further parameters sent to ModelMatrix. In particular, one can specify removeEmpty=TRUE to omit empty combinations. The parameter inputInOutput can be used to specify whether to include codes from input. The parameter avoidHierarchical (Formula2ModelMatrix) can be combined with formula input.

Details

Small count rounding of necessary inner cells are performed so that all small frequencies of cross-classifications to be published (publishable cells) are rounded. This is equivalent to changing micro data since frequencies of unique combinations are changed. Thus, additivity and consistency are guaranteed. The matrix multiplication formula is: yPublish = t(x) %*% yInner, where x is the dummy matrix.

Value

A list where the two first elements are two column matrices. The first matrix consists of inner cells and the second of cells to be published. In each matrix the first and the second column contains, respectively, original and rounded values. By default the cross table is the third element of the output list.

Note

Iterations are needed since after initial rounding of identified cells, new cells are identified. If cases of a high number of identified cells the algorithm can be too memory consuming (unless singleRandom=TRUE). To avoid problems, not more than maxIterRows cells are rounded in each iteration. The iteration limit (maxIter) is by default set to be high since a low number of maxIterRows may need a high number of iterations.

See Also

See the user-friendly wrapper PLSrounding and see Round2 for rounding by other algorithm

Examples

# See similar and related examples in PLSrounding documentation
RoundViaDummy(SmallCountData("e6"), "freq")
RoundViaDummy(SmallCountData("e6"), "freq", formula = ~eu * year + geo)
RoundViaDummy(SmallCountData("e6"), "freq", hierarchies =
  list(geo = c("EU", "@Portugal", "@Spain", "Iceland"),
       year = c("2018", "2019")))

RoundViaDummy(SmallCountData('z2'),
  'ant', ~region + hovedint + fylke*hovedint + kostragr*hovedint, 10)
mf <- ~region*mnd + hovedint*mnd + fylke*hovedint*mnd + kostragr*hovedint*mnd
a <- RoundViaDummy(SmallCountData('z3'), 'ant', mf, 5)
b <- RoundViaDummy(SmallCountData('sosialFiktiv'), 'ant', mf, 4)
print(cor(b[[2]]),digits=12) # Correlation between original and rounded
# Demonstrate parameter leverageCheck
# The 42nd inner cell must be rounded since it can be revealed from the published cells.
mf2 <- ~region + hovedint + fylke * hovedint + kostragr * hovedint
RoundViaDummy(SmallCountData("z2"), "ant", mf2, leverageCheck = FALSE)$yInner[42, ]
RoundViaDummy(SmallCountData("z2"), "ant", mf2, leverageCheck = TRUE)$yInner[42, ]

## Not run:
# Demonstrate parameters maxRound, zeroCandidates and forceInner
# by tabulating the inner cells that have been changed.
z4 <- SmallCountData("sosialFiktiv")
for (forceInner in c("FALSE", "z4$ant < 10"))
  for (zeroCandidates in c(FALSE, TRUE))
    for (maxRound in c(2, 5)) {
      set.seed(123)
      a <- RoundViaDummy(z4, "ant", formula = mf, maxRound = maxRound,
                          zeroCandidates = zeroCandidates,
                          forceInner = eval(parse(text = forceInner)))
      change <- a$yInner[, "original"] != a$yInner[, "rounded"]
      cat("n\n
                 maxRound:", maxRound, "n")
      cat("zeroCandidates:", zeroCandidates, "n")
      cat("forceInner:", forceInner, "n")
      print(table(original = a$yInner[change, "original"], rounded = a$yInner[change, "rounded"]))
      cat("------------------------------\n")
    }
## End(Not run)

SmallCountData

Function that returns a dataset

Description

Function that returns a dataset

Usage

SmallCountData(dataset, path = NULL)

Arguments

dataset Name of data set within the SmallCountRounding package

path When non-NULL the data set is read from "path/dataset.RData"

Value

The dataset
SmallCountData

Note

Except for "europe6", "eHrc", "eDimList" and "exPSD", the function returns the same datasets as SSBtoolsData.

See Also

SSBtoolsData, Hrc2DimList

Examples

SmallCountData("z1")
SmallCountData("e6")
SmallCountData("eHrc") # TauArgus coded hierarchies
SmallCountData("eDimList") # sdcTable coded hierarchies
SmallCountData("exPSD") # Example data in presentation at Privacy in statistical databases
Index

* print
  print.PLSrounded, 9

AutoHierarchies, 11

Extend0, 7

Formula2ModelMatrix, 12

HD, 2
HDutility, 5
HDutility (HD), 2
Hrc2DimList, 14

Mipf, 7
ModelMatrix, 6, 10, 12

PLS2way, 3, 6
PLSrounding, 2–4, 4, 7–9, 12
PLSroundingFits, 7
PLSroundingInner (PLSrounding), 4
PLSroundingPublish (PLSrounding), 4
print.PLSrounded, 9

Reduce0exact, 11
RoundViaDummy, 5, 6, 10

SmallCountData, 13
SmallCountRounding
  (SmallCountRounding-package), 2
SmallCountRounding-package, 2
SSBtoolsData, 14