

Package ‘sdcMicro’

October 22, 2009

Type Package

Title Statistical Disclosure Control methods for the generation of public- and scientific-use files.

Version 2.6.2

Date 2009-10-21

Author Matthias Templ

Maintainer Matthias Templ <Matthias.Templ@statistik.gv.at>

Description Data from statistical agencies and other institutions are mostly confidential. This package can be used for the generation of save (micro)data, i.e. for the generation of public- and scientific-use files. The package includes also a graphical user interface.

Depends car, robustbase, tcltk, cluster, MASS, far, e1071, mclust, gWidgetsRGtk2

License GPL-2

Repository CRAN

Date/Publication 2009-10-22 18:37:42

R topics documented:

sdcMicro-package	2
addNoise	5
cascl	7
CASCrefmicrodata	7
dataGen	9
dRisk	10
dRiskRMD	11
dUtility	12
EIA	14
francdat	17
free1	18
freqCalc	18
globalRecode	20

indivRisk	21
localSupp	22
localSupp2	23
localSupp2Wrapper	25
microaggregation	27
microData	29
plot.indivRisk	30
plot.localSupp2	31
plotMicro	32
pram	33
print.freqCalc	34
print.indivRisk	35
print.localSupp2	36
print.micro	36
print.pram	37
sdcGUI	38
summary.freqCalc	39
summary.micro	40
summary.pram	41
swappNum	42
Tarragona	43
topBotCoding	44
valTable	45
Index	47

sdcMicro-package *Statistical Disclosure Control (SDC) for the generation of protected microdata for researchers and for public use.*

Description

This package includes all methods of the popular software mu-Argus plus several new methods. In comparison with mu-Argus the advantages of this package are that the results are fully reproducible even with the included GUI, that the package can be used in batch-mode from other software, that the functions can be used in a very flexible way, that everybody could look at the source code and that there are no time-consuming meta-data management is necessary. However, the user should have a detailed knowledge about SDC when applying the methods on data.

The implemented graphical user interface (GUI) for microdata protection serves as an easy-to-handle tool for users who want to use the sdcMicro package for statistical disclosure control but are not used to the native R command line interface. In addition to that, interactions between objects which results from the anonymization process are provided within the GUI. This allows an automated recalculation and displaying information of the frequency counts, individual risk, information loss and data utility after each anonymization step. In addition to that, the code for every anonymization step carried out within the GUI is saved in a script which can then be easily modified and reloaded.

Please note, that methods “shuffling”, “robShuffle” (robust shuffling), “gadp” and “robgadp” are not included in the package because method “shuffling” is under a US-patent by other authors, even shuffling consists only of 8 lines of code ...

Details

Package: sdcMicro
 Type: Package
 Version: 2.5.9
 Date: 2009-07-22
 License: GPL 2.0

Author(s)

Matthias Templ

Maintainer: Matthias Templ <templ@statistik.tuwien.ac.at>

References

Templ, M. *Statistical Disclosure Control for Microdata Using the R-Package sdcMicro*, Transactions on Data Privacy, vol. 1, number 2, pp. 67-85, 2008. <http://www.tdp.cat/issues/abs.a004a08.php>

Templ, M. *New Developments in Statistical Disclosure Control and Imputation: Robust Statistics Applied to Official Statistics*, Suedwestdeutscher Verlag fuer Hochschulschriften, 2009, ISBN: 3838108280, 264 pages.

Examples

```
## example from Capobianchi, Poletti and Lucarelli:
data(franmdat)
f <- freqCalc(franmdat, keyVars=c(2,4,5,6),w=8)
f
f$fk
f$Fk
## with missings:
x <- franmdat
x[3,5] <- NA
x[4,2] <- x[4,4] <- NA
x[5,6] <- NA
x[6,2] <- NA
f2 <- freqCalc(x, keyVars=c(2,4,5,6),w=8)
f2$Fk
## individual risk calculation:
indivf <- indivRisk(f)
indivf$rk
## Local Suppression
locals <- localSupp(f, keyVar=2, indivRisk=indivf$rk, threshold=0.25)
```

```

f2 <- freqCalc(localS$freqCalc, keyVars=c(2,4,5,6), w=8)
indivf2 <- indivRisk(f2)
indivf2$rk

## select another keyVar and run localSupp once again, if you think the table is not fully p
data(free1)
f <- freqCalc(free1, keyVars=1:3, w=30)
ind <- indivRisk(f)
## and now you can use the interactive plot for individual risk objects:
## plot(ind)

## Local suppression with localSupp2 and localSupp2Wrapper is more effective:
## example from Capobianchi, Polettoni and Lucarelli:
data(francdat)
l1 <- localSupp2(francdat, keyVars=c(2,4,5,6), w=8)
l1
l1$x
l2 <- localSupp2(francdat, keyVars=c(2,4,5,6), w=8, k=2)
l3 <- localSupp2(francdat, keyVars=c(2,4,5,6), w=8, k=4)
## long computation time:
## l = localSupp2(free1, keyVar=1:3, w=30, k=2, importance=c(0.1,1,0.8))

## we want to avoid missings in column 5:
l1 <- localSupp2Wrapper(francdat, keyVars=c(2,4,5,6), importance=c(1,1,0,1), w=8, kAnon=1)
l1$x
## we want to avoid missings in column 5 and allow missings in 1 only if
## is really necessary:
l1 <- localSupp2Wrapper(francdat, keyVars=c(2,4,5,6), importance=c(0.1,1,0,1), w=8, kAnon=1)
l1$x
plot(l1)

## Data from mu-Argus:
## Global recoding:
data(free1)
free1[, "AGE"] <- globalRecode(free1[, "AGE"], c(1,9,19,29,39,49,59,69,100), labels=1:8)

## Top coding:
topBotCoding(free1[, "DEBTS"], value=9000, replacement=9100, kind="top")

## Numerical Rank Swapping:
## do not use the mu-Argus test data set (free1) since the numerical variables are (probably
data(Tarragona)
Tarragona1 <- swappNum(Tarragona, p=10)

## Microaggregation:
m1 <- microaggregation(Tarragona, method="onedims", aggr=3)
m2 <- microaggregation(Tarragona, method="pca", aggr=3)
# summary(m1)
# valTable(Tarragona, method=c("simple","onedims","pca")) ## approx. 1 minute computation ti

data(microData)
m1 <- microaggregation(microData, method="mdav")
x <- m1$x ### fix me

```

```

summary(m1)
plotMicro(m1, 0.1, which.plot=1) # too less observations...
data(freel)
plotMicro(microaggregation(freel[,31:34], method="onedims"), 0.1, which.plot=1)

## disclosure risk (interval) and data utility:
data(freel)
m1 <- microaggregation(Tarragona, method="onedims", aggr=3)
dRisk(x=Tarragona, xm=m1$blowxm)
dRisk(x=Tarragona, xm=m2$blowxm)
dUtility(x=Tarragona, xm=m1$blowxm)
dUtility(x=Tarragona, xm=m2$blowxm)

## S4 class code for Adding Noise methods will be included in the next version of sdcMicro.

## Fast generation of synthetic data with aprox. the same covariance matrix as the original

data(mtcars)
cov(mtcars[,4:6])
cov(dataGen(mtcars[,4:6],n=200))
pairs(mtcars[,4:6])
pairs(dataGen(mtcars[,4:6],n=200))

## PRAM

set.seed(123)
x <- sample(1:4, 250, replace=TRUE)
pr1 <- pram(x)
length(which(pr1$x == x))
x2 <- sample(1:4, 250, replace=TRUE)
length(which(pram(x2)$x == x2))

data(freel)
marstatPramed <- pram(freel[, "MARSTAT"])

```

addNoise

Adding noise for the perturbation of data

Description

Various adding noise methods for the perturbation of continuous scaled variables can be used.

Usage

```
addNoise(x, noise = 150, method = "additive", p = 0.001, delta=0.1)
```

Arguments

x	data frame or matrix which should be perturbed
noise	amount of noise (in percentages)
method	choose between 'additive', 'correlated', 'correlated2', 'restr', 'ROMM', 'outdetect'
p	multiplication factor for method 'ROMM'
delta	parameter for method 'correlated2', details can be found in the reference below.

Details

Method 'additive' adds noise completely at random to each variable depending on there size and standard deviation. 'correlated' and method 'correlated2' adds noise and preserves the covariances as described in R. Brand (2001) or in the reference given below. Method 'restr' takes the sample size into account when adding noise. Method 'ROMM' is an implementation of the algorithm ROMM (Random Orthogonalized Matrix Masking) (Fienberg, 2004). Method 'outdetect' adds noise only to outliers. The outliers are ididentified with univariate and robust multivariate procedures based on a robust mahalanobis distancs calculated by the MCD estimator.

Value

An object of class "micro" with following entities:

x	the original data
xm	the modified (perturbed) data
method	method used for perturbation
noise	amount of noise

Author(s)

Matthias Templ

References

- Domingo-Ferrer, J. and Sebe, F. and Castella, J., "On the security of noise aliiition for privacy in statistical databases", Lecture Notes in Computer Science, vol. 3050, pp. 149-161, 2004. ISSN 0302-9743. Vol. Privacy in Statistical Databases, eds. J. Domingo-Ferrer and V. Torra, Berlin: Springer-Verlag. <http://vneumann.etse.urv.es/publications/sci/lncs3050OntheSec.pdf>,
- Ting and Fienberg, "ROMM Methodology for Microdata Release" Joint UNECE/Eurostat work session on statistical data confidentiality, Geneva, Switzerland, 2005, [http://www.niss.org/dgii/TR/wp.11.e\(ROMM\).pdf](http://www.niss.org/dgii/TR/wp.11.e(ROMM).pdf)
- Templ, M. and Meindl, B., *Robustification of Microdata Masking Methods and the Comparison with Existing Methods*, Lecture Notes in Computer Science, Privacy in Statistical Databases, vol. 5262, pp. 177-189, 2008.
- Templ, M. *New Developments in Statistical Disclosure Control and Imputation: Robust Statistics Applied to Official Statistics*, Suedwestdeutscher Verlag fuer Hochschulschriften, 2009, ISBN: 3838108280, 264 pages.

See Also

[summary.micro](#)

Examples

```
data(Tarragona)
a1 <- addNoise(Tarragona)
a1
valTable(Tarragona[100:200,], method=c("addNoise: additive", "addNoise: correlated2"))
```

cascl

Small Artificial Data set

Description

Small Example Data set which was used by Sanz-Mateo et.al.

Usage

```
data(cascl)
```

Format

The format is: int [1:13, 1:7] 10 12 17 21 9 12 12 14 13 15 ... - attr(*, "dimnames")=List of 2
..: chr [1 : 13] "1" "2" "3" "4" : chr [1:7] "1" "2" "3" "4" ...

Examples

```
data(cascl)
cascl
```

CASCrefmicrodata

Census data set

Description

This test data set was obtained on July 27, 2000 using the Data Extraction System of the U. S. Bureau of the Census.

Usage

```
data(CASCrefmicrodata)
```

Format

A data frame sampled from year 1995 with 1080 observations on the following 13 variables.

AFNLWGT Final weight (2 implied decimal places)

AGI Adjusted gross income

EMCONTRB Employer contribution for hlth insurance

FEDTAX Federal income tax liability

PTOTVAL Total person income

STATETAX State income tax liability

TAXINC Taxable income amount

POTHVAL Total other persons income

INTVAL Amt of interest income

PEARNVAL Total person earnings

FICA Soc. sec. retirement payroll deduction

WSALVAL Amount: Total Wage and salary

ERNVAL Business or Farm net earnings

Details

Further details are given in the link below.

Source

<http://neon.vb.cbs.nl/casc/testsets.html>

References

Brand, R. and Domingo-Ferrer, J. and Mateo-Sanz, J.M., Reference data sets to test and compare SDC methods for protection of numerical microdata. Unpublished. <http://neon.vb.cbs.nl/casc/CASCrefermicrodata.pdf>

Examples

```
data(CASCrefermicrodata)
head(CASCrefermicrodata)
```

`dataGen`*Fast generation of synthetic data*

Description

Fast generation of synthetic data.

Usage

```
dataGen(x, n = 200)
```

Arguments

<code>x</code>	data.frame or matrix
<code>n</code>	amount of observations for the generated data

Details

Uses the cholesky decomposition to generate synthetic data. For details see at the reference.

Value

the generated synthetic data.

Note

With this method only multivariate normal distributed data with approximately the same covariance as the original data can be generated without reflecting the distribution of real complex data, which are, in general, not follows a multivariate normal distribution.

Author(s)

Matthias Templ

References

Have a look at <http://vneumann.etse.urv.es/publications/sci/lncs3050FastGen.pdf>

Examples

```
data(mtcars)
cov(mtcars[,4:6])
cov(dataGen(mtcars[,4:6]))
pairs(mtcars[,4:6])
pairs(dataGen(mtcars[,4:6]))
```

dRisk *overall disclosure risk*

Description

Distance-based disclosure risk estimation via standard deviation-based intervals.

Usage

```
dRisk(x, xm, k = 0.01)
```

Arguments

x	original data
xm	perturbed data
k	percentage of the standard deviation

Details

An interval is built around each value of the perturbed value with the help of the standard deviation. Then we look if the original values lay in these intervals or not. With parameter k one can enlarge or down scale the interval.

Value

The disclosure risk.

Author(s)

Matthias Templ

References

see method SDID in <http://vneumann.etse.urv.es/publications/sci/lncs3050Outlier.pdf>

See Also

[dUtility](#), [dUtility](#)

Examples

```
data(free1)
m1 <- microaggregation(free1[, 31:34], method="onedims", aggr=3)
m2 <- microaggregation(free1[, 31:34], method="pca", aggr=3)
dRisk(x=free1[, 31:34], xm=m1$blowxm)
dRisk(x=free1[, 31:34], xm=m2$blowxm)
dUtility(x=free1[, 31:34], xm=m1$blowxm)
dUtility(x=free1[, 31:34], xm=m2$blowxm)
```

dRiskRMD

RMD based disclosure risk

Description

Distance-based disclosure risk estimation via robust Mahalanobis Distances.

Usage

```
dRiskRMD(x, xm, k = 0.01, k2=0.05)
```

Arguments

x	original data
xm	masked data
k	weight for adjusting the influence of the robust Mahalanobis distances, i.e. to increase or decrease each of the disclosure risk intervals.
k2	parameter for method RMDID2 to choose a small interval around each masked observation.

Details

This method is an extension of method SDID because it accounts for the “outlyingness” of each observations. This is a quite natural approach since outliers do have a higher risk of re-identification and therefore these outliers should have larger disclosure risk intervals as observations in the center of the data cloud.

The algorithm works as follows:

1. Robust Mahalanobis distances are estimated in order to get a robust multivariate distance for each observation.
2. Intervals are estimated for each observation around every data point of the original data points where the length of the interval is defined/weighted by the squared robust Mahalanobis distance and the parameter k . The higher the RMD of an observation the larger the interval.
3. Check if the corresponding masked values fall into the intervals around the original values or not. If the value of the corresponding observation is within such an interval the whole observation is considered unsafe. So, we get a whole vector indicating which observation is save or not, and we are finished already when using method RMDID1).
4. For method RMDID1w: we return the weighted (via RMD) vector of disclosure risk.
5. For method RMDID2: whenever an observation is considered unsafe it is checked if m other observations from the masked data are very close (defined by a parameter $k2$ for the length of the intervals as for SDID or RSDID) to such an unsafe observation from the masked data, using Euclidean distances. If more than m points are in such a small interval, we conclude that this observation is “save”.

Value

The disclosure risk.

risk1	percentage of sensitive observations according to method RMDID1.
risk2	standardized version of risk1
wrisk1	amount of sensitive observations according to RMDID1 weighted by their corresponding robust Mahalanobis distances.
wrisk2	RMDID2 measure
indexRisk1	index of observations with high risk according to risk1 measure
indexRisk2	index of observations with high risk according to wrisk2 measure

Author(s)

Matthias Templ

References

Templ, M. and Meindl, B., *Robust Statistics Meets SDC: New Disclosure Risk Measures for Continuous Microdata Masking*, Lecture Notes in Computer Science, Privacy in Statistical Databases, vol. 5262, pp. 113-126, 2008.

Templ, M. *New Developments in Statistical Disclosure Control and Imputation: Robust Statistics Applied to Official Statistics*, Suedwestdeutscher Verlag fuer Hochschulschriften, 2009, ISBN: 3838108280, 264 pages.

See Also

[dRisk](#)

Examples

```
data(Tarragona)
x <- Tarragona[, 5:7]
y <- addNoise(x)$xm
dRiskRMD(x, xm=y)
dRisk(x, xm=y)
```

dUtility

data utility

Description

IL1s data utility.

Usage

```
dUtility(x, xm, method="IL1")
```

Arguments

x	original data
xm	perturbed data
method	method IL1 or eigen. More methods are implemented in summary.micro()

Details

The standardised distances of the perturbed data values to the original ones are measured. Measure IL1 measures the distances between the original values and the perturbed ones, scaled by the standard deviation. Method 'eigen' and 'robeigen' compares the eigenvalues and robust eigenvalues from the original data and the perturbed data.

Value

data utility

Author(s)

Matthias Templ

References

for IL1s: see <http://vneumann.etse.urv.es/publications/sci/lncs3050Outlier.pdf>,

Templ, M. and Meindl, B., *Robust Statistics Meets SDC: New Disclosure Risk Measures for Continuous Microdata Masking*, Lecture Notes in Computer Science, Privacy in Statistical Databases, vol. 5262, pp. 113-126, 2008.

See Also

[dRisk](#), [dRiskRMD](#)

Examples

```
data(free1)
m1 <- microaggregation(free1[, 31:34], method="onedims", aggr=3)
m2 <- microaggregation(free1[, 31:34], method="pca", aggr=3)
dRisk(x=free1[, 31:34], xm=m1$blowxm)
dRisk(x=free1[, 31:34], xm=m2$blowxm)
dUtility(x=free1[, 31:34], xm=m1$blowxm)
dUtility(x=free1[, 31:34], xm=m2$blowxm)
data(Tarragona)
x <- Tarragona[, 5:7]
y <- addNoise(x)$xm
dRiskRMD(x, xm=y)
dRisk(x, xm=y)
dUtility(x, xm=y)
dUtility(x, xm=y, method="eigen")
dUtility(x, xm=y, method="robeigen")
```

EIA

*EIA data set***Description**

Data set obtained from the U.S. Energy Information Authority.

Usage

data (EIA)

Format

A data frame with 4092 observations on the following 15 variables.

UTILITYID UNIQUE UTILITY IDENTIFICATION NUMBER

UTILNAME UTILITY NAME. A factor with levels 4-County Electric Power Assn Alabama Power Co Alaska Electric Light & Power Co Anchorage City of Anoka Electric Coop Appalachian Electric Coop Appalachian Power Co Arizona Public Service Co Arkansas Power & Light Co Arkansas Valley Elec Coop Corp Atlantic City Electric Company Baker Electric Coop Inc Baltimore Gas & Electric Co Bangor Hydro-Electric Co Berkeley Electric Coop Inc Black Hills Corp Blackstone Valley Electric Co Bonneville Power Admin Boston Edison Co Bountiful City Light & Power Bristol City of Brookings City of Brunswick Electric Member Corp Burlington City of Carolina Power & Light Co Carroll Electric Coop Corp Cass County Electric Coop Inc Central Illinois Light Company Central Illinois Pub Serv Co Central Louisiana Elec Co Inc Central Maine Power Co Central Power & Light Co Central Vermont Pub Serv Corp Chattanooga City of Cheyenne Light Fuel & Power Co Chugach Electric Assn Inc Cincinnati Gas & Electric Co Citizens Utilities Company City of Boulder City City of Clinton City of Dover City of Eugene City of Gillette City of Groton Dept of Utils City of Idaho Falls City of Independence City of Newark City of Reading City of Tupelo Water & Light D Clarksville City of Cleveland City of Cleveland Electric Illum Co Coast Electric Power Assn Cobb Electric Membership Corp Colorado River Commission Colorado Springs City of Columbus Southern Power Co Commonwealth Edison Co Commonwealth Electric Co Connecticut Light & Power Co Consolidated Edison Co-NY Inc Consumers Power Co Cornhusker Public Power Dist Cuivre River Electric Coop Inc Cumberland Elec Member Corp Dakota Electric Assn Dawson County Public Pwr Dist Dayton Power & Light Company Decatur City of Delaware Electric Coop Inc Delmarva Power & Light Co Detroit Edison Co Duck River Elec Member Corp Duke Power Co Duquesne Light Company East Central Electric Assn Eastern Maine Electric Coop El Paso Electric Co Electric Energy Inc Empire District Electric Co Exeter & Hampton Electric Co Fairbanks City of Fayetteville Public Works Comm First Electric Coop Corp

Florence City of Florida Power & Light Co Florida Power Corp Fort Collins Lgt & Pwr Utility Fremont City of Georgia Power Co Gibson County Elec Member Corp Golden Valley Elec Assn Inc Grand Island City of Granite State Electric Co Green Mountain Power Corp Green River Electric Corp Greeneville City of Gulf Power Company Gulf States Utilities Co Hasting Utilities Hawaii Electric Light Co Inc Hawaiian Electric Co Inc Henderson-Union Rural E C C Homer Electric Assn Inc Hot Springs Rural El Assn Inc Houston Lighting & Power Co Huntsville City of Idaho Power Co IES Utilities Inc Illinois Power Co Indiana Michigan Power Co Indianapolis Power & Light Co Intermountain Rural Elec Assn Interstate Power Co Jackson Electric Member Corp Jersey Central Power & Light Co Joe Wheeler Elec Member Corp Johnson City City of Jones-Onslow Elec Member Corp Kansas City City of Kansas City Power & Light Co Kentucky Power Co Kentucky Utilities Co Ketchikan Public Utilities Kingsport Power Co Knoxville City of Kodiak Electric Assn Inc Kootenai Electric Coop, Inc Lansing Board of Water & Light Lenoir City City of Lincoln City of Long Island Lighting Co Los Angeles City of Louisiana Power & Light Co Louisville Gas & Electric Co Loup River Public Power Dist Lower Valley Power & Light Inc Maine Public Service Company Massachusetts Electric Co Matanuska Electric Assn Inc Maui Electric Co Ltd McKenzie Electric Coop Inc Memphis City of MidAmerican Energy Company Middle Tennessee E M C Midwest Energy, Inc Minnesota Power & Light Co Mississippi Power & Light Co Mississippi Power Co Monongahela Power Co Montana-Dakota Utilities Co Montana Power Co Moon Lake Electric Assn Inc Narragansett Electric Co Nashville City of Nebraska Public Power District Nevada Power Co New Hampshire Elec Coop, Inc New Orleans Public Service Inc New York State Gas & Electric Newport Electric Corp Niagara Mohawk Power Corp Nodak Rural Electric Coop Inc Norris Public Power District Northeast Oklahoma Electric Co Northern Indiana Pub Serv Co Northern States Power Co Northwestern Public Service Co Ohio Edison Co Ohio Power Co Ohio Valley Electric Corp Oklahoma Electric Coop, Inc Oklahoma Gas & Electric Co Oliver-Mercer Elec Coop, Inc Omaha Public Power District Otter Tail Power Co Pacific Gas & Electric Co Pacificorp dba Pacific Pwr & L Palmetto Electric Coop, Inc Pennsylvania Power & Light Co Pennyrile Rural Electric Coop Philadelphia Electric Co Pierre Municipal Electric Portland General Electric Co Potomac Edison Co Potomac Electric Power Co Poudre Valley R E A, Inc Power Authority of State of NY Provo City Corporation Public Service Co of Colorado Public Service Co of IN Inc Public Service Co of NH Public Service Co of NM Public Service Co of Oklahoma Public Service Electric & Gas Co PUD No 1 of Clark County PUD No 1 of Snohomish County Puget Sound Power & Light Co Rappahannock Electric Coop Rochester Public Utilities Rockland Electric Company Rosebud Electric Coop Inc Rutherford Elec Member Corp Sacramento Municipal Util Dist Salmon River Electric Coop Inc Salt River Proj Ag I & P Dist San Antonio City of Savannah Electric & Power Co Seattle City of Sierra Pacific Power Co Singing River Elec Power Assn Sioux Valley Empire E A Inc South Carolina

Electric&Gas Co South Carolina Pub Serv Auth South Kentucky Rural
 E C C Southern California Edison Co Southern Nebraska Rural P P
 D Southern Pine Elec Power Assn Southwest Tennessee E M C Southwestern
 Electric Power Co Southwestern Public Service Co Springfield City
 of St Joseph Light & Power Co State Level Adjustment Tacoma City
 of Tampa Electric Co Texas-New Mexico Power Co Texas Utilities Electric
 Co Tri-County Electric Assn Inc Tucson Electric Power Co Turner-Hutchinsin
 El Coop, Inc TVA U S Bureau of Indian Affairs Union Electric Co Union
 Light Heat & Power Co United Illuminating Co Upper Cumberland E
 M C UtiliCorp United Inc Verdigris Valley Electric Coop Verendrye
 Electric Coop Inc Virginia Electric & Power Co Volunteer Electric
 Coop Wallingford Town of Warren Rural Elec Coop Corp Washington Water
 Power Co Watertown Municipal Utils Dept Wells Rural Electric Co
 West Penn Power Co West Plains Electric Coop Inc West River Electric
 Assn, Inc Western Massachusetts Elec Co Western Resources Inc Wheeling
 Power Company Wisconsin Electric Power Co Wisconsin Power & Light
 Co Wisconsin Public Service Corp Wright-Hennepin Coop Elec Assn
 Yellowstone Vlly Elec Coop Inc

STATE STATE FOR WHICH THE UTILITY IS REPORTING. A factor with levels AK AL AR AZ
 CA CO CT DC DE FL GA HI IA ID IL IN KS KY LA MA MD ME MI MN MO MS MT NC ND NE
 NH NJ NM NV NY OH OK OR PA RI SC SD TN TX UT VA VT WA WI WV WY

YEAR REPORTING YEAR FOR THE DATA

MONTH REPORTING MONTH FOR THE DATA

RESREVENUE REVENUE FROM SALES TO RESIDENTIAL CONSUMERS

RESSALES SALES TO RESIDENTIAL CONSUMERS

COMREVENUE REVENUE FROM SALES TO COMMERCIAL CONSUMERS

COMSALES SALES TO COMMERCIAL CONSUMERS

INDREVENUE REVENUE FROM SALES TO INDUSTRIAL CONSUMERS

INDSALES SALES TO INDUSTRIAL CONSUMERS

OTHEREVENUE REVENUE FROM SALES TO OTHER CONSUMERS

OTHRSALES SALES TO OTHER CONSUMERS

TOTREVENUE REVENUE FROM SALES TO ALL CONSUMERS

TOTSALES SALES TO ALL CONSUMERS

Details

Further details can be found on the link given below.

Source

<http://neon.vb.cbs.nl/casc/testsets.html>

References

Brand, R. and Domingo-Ferrer, J. and Mateo-Sanz, J.M., Reference data sets to test and compare SDC methods for protection of numerical microdata. Unpublished. <http://neon.vb.cbs.nl/casc/CAScrefmicrodata.pdf>

Examples

```
data(EIA)
head(EIA)
```

francdat	<i>data from the casc project</i>
----------	-----------------------------------

Description

Small synthetic data from Capobianchi, Poletti, Lucarelli

Usage

```
data(francdat)
```

Format

A data frame with 8 observations on the following 8 variables.

Num1 a numeric vector

Key1 Key variable 1. A numeric vector

Num2 a numeric vector

Key2 Key variable 2. A numeric vector

Key3 Key variable 3. A numeric vector

Key4 Key variable 4. A numeric vector

Num3 a numeric vector

w The weight vector. A numeric vector

Details

This data set is very similar to that one which are used by the authors of the paper given below. We need this data set only for demonstration effect, i.e. that the package provides the same results as their software.

Source

<http://neon.vb.cbs.nl/casc/Deliv/12d1.pdf>

Examples

```
data(francdat)
francdat
```

`free1`*Demo data set from mu-Argus*

Description

Demo data set from the mu-Argus software for SDC.

Usage

```
data(free1)
```

Format

The format is: num [1:4000, 1:34] 36 36 36 36 36 36 36 36 36 36 ... - attr(*, "dimnames")=List of 2 .. *NULL*.. : chr [1:34] "REGION" "SEX" "AGE" "MARSTAT" ...

Details

Please, see at the link given below. Please note, that the correlation structure of the data is not realistic.

Source

<http://neon.vb.cbs.nl/casc/>

Examples

```
data(free1)
head(free1)
```

`freqCalc`*Frequencies calculation for risk estimation*

Description

Fast computation and estimation of the sample and population frequency counts which is also needed for risk estimation.

Usage

```
freqCalc(x, keyVars = 1:3, w = 4)
```

Arguments

x	data frame or matrix
keyVars	key variables
w	column index of the weight variable. Should be set to NULL if one deal with a population.

Details

The function considers the case of missing values in the data. A missing value stands for any of the possible categories of the variable considered. It is possible to apply this function to large data sets with many (catergorical) key variables, since the computation is done in C.

Value

Object from class freqCalc.

freqCalc	data
keyVars	keyVars
w	index of weight vector. NULL if you do not a sample.
indexG	
fk	the frequency of equal observations in the key variables subset sample given for each observation.
Fk	estimated frequency in the population
n1	amount of observations with fk=1
n2	amount of observations with fk=2

Author(s)

Bernhard Meindl and Matthias Templ

References

look e.g. in <http://neon.vb.cbs.nl/casc/Deliv/12d1.pdf> Templ, M. *Statistical Disclosure Control for Microdata Using the R-Package sdcMicro*, Transactions on Data Privacy, vol. 1, number 2, pp. 67-85, 2008. <http://www.tdp.cat/issues/abs.a004a08.php>

Templ, M. *New Developments in Statistical Disclosure Control and Imputation: Robust Statistics Applied to Official Statistics*, Suedwestdeutscher Verlag fuer Hochschulschriften, 2009, ISBN: 3838108280, 264 pages.

See Also

[indivRisk](#)

Examples

```

data(franccdat)
f <- freqCalc(franccdat, keyVars=c(2,4,5,6),w=8)
f
f$freqCalc
f$fk
f$Fk
## with missings:
x <- franccdat
x[3,5] <- NA
x[4,2] <- x[4,4] <- NA
x[5,6] <- NA
x[6,2] <- NA
f2 <- freqCalc(x, keyVars=c(2,4,5,6),w=8)
f2$Fk

```

globalRecode

Global Recoding

Description

Global recoding

Usage

```
globalRecode(x, breaks, labels, method="equidistant")
```

Arguments

x	vector of class numeric or of class factor with integer labels for recoding
breaks	either a numeric vector of cut points or number giving the number of intervals which x is to be cut into.
labels	labels for the levels of the resulting category. By default, labels are constructed using "(a,b)" interval notation. If labels = FALSE, simple integer codes are returned instead of a factor.
method	method "equidistant" for equal sized intervalls method "logEqui" for equal sized intervalls for log-transformed data method "equalAmount" for intervalls with approximately the same amount of observations

Details

If a labels parameter is specified, its values are used to name the factor levels. If none is specified, the factor level labels are constructed.

Value

A factor is returned, unless labels = FALSE which results in the mere integer level codes.

See Also[cut](#)**Examples**

```

data(free1)
head(globalRecode(free1[, "AGE"], breaks=c(1, 9, 19, 29, 39, 49, 59, 69, 100), labels=1:8))
table(globalRecode(free1[, "AGE"], breaks=c(1, 9, 19, 29, 39, 49, 59, 69, 100), labels=1:8))
table(globalRecode(free1[, "AGE"], breaks=c(1, 9, 19, 29, 39, 49, 59, 69, 100)))
table(globalRecode(free1[, "AGE"], breaks=6))
table(globalRecode(free1[, "AGE"], breaks=6, method="logEqui"))
table(globalRecode(free1[, "AGE"], breaks=6, method="equalAmount"))

```

indivRisk

*Individual Risk computation***Description**

Base individual risk computation.

Usage

```
indivRisk(x, method = "approx", qual = 1, survey=TRUE)
```

Arguments

x	object from class freqCalc
method	approx (default) or exact
qual	final correction factor
survey	TRUE, if one have survey data and FALSE if one deal with the whole population.

Details

Estimation of the risk for each observation. After the risk is computed one can use e.g. the function `localSuppr()` for the protection of values of high risk. Further details can be found at the link given below.

Value

rk	base individual risk
method	method
qual	final correction factor
fk	frequency count
knames	colnames of the key variables

Note

The base individual risk method was developed by Benedetti, Capobianchi and Franconi

Author(s)

Matthias Templ

References

have a look at: <http://neon.vb.cbs.nl/casc/Deliv/12d1.pdf> or http://www.istat.it/dati/pubbsci/contributi/Contributi/contr_2003/2003_14.pdf

See Also

[freqCalc](#)

Examples

```
## example from Capobianchi, Polettini and Lucarelli:
data(franccat)
f <- freqCalc(franccat, keyVars=c(2,4,5,6),w=8)
f
f$fk
f$Fk
## individual risk calculation:
indivf <- indivRisk(f)
indivf$rk
```

localSupp

Local Suppression

Description

A simple method to perform local suppression.

Usage

```
localSupp(x, keyVar, indivRisk, threshold = 0.15)
```

Arguments

x	object from class freqCalc
keyVar	Variable on which some values might be suppressed
indivRisk	object from class indivRisk
threshold	threshold for individual risk

Details

Values of high risk (above the threshold) of a certain variable (parameter keyVar) are suppressed.

Value

Manipulated data with suppressions

Author(s)

Matthias Templ

References

Templ, M. *Statistical Disclosure Control for Microdata Using the R-Package sdcMicro*, Transactions on Data Privacy, vol. 1, number 2, pp. 67-85, 2008. <http://www.tdp.cat/issues/abs.a004a08.php>

See Also

[freqCalc](#), [indivRisk](#)

Examples

```
## example from Capobianchi, Polettini and Lucarelli:
data(franccdat)
f <- freqCalc(franccdat, keyVars=c(2,4,5,6),w=8)
f
f$fk
f$Fk
## individual risk calculation:
indivf <- indivRisk(f)
indivf$rk
## Local Suppression
localS <- localSupp(f, keyVar=2, indivRisk=indivf$rk, threshold=0.25)
f2 <- freqCalc(localS$freqCalc, keyVars=c(4,5,6), w=8)
indivf2 <- indivRisk(f2)
indivf2$rk
## select another keyVar and run localSupp once again, if you think the table is not fully p
```

localSupp2

Local Suppression 2

Description

An Algorithm to perform local suppression to achieve k-anonymity.

Usage

```
localSupp2(x, keyVars, w, importance=rep(1, length(keyVars)), method="minimizeSupp"
```

Arguments

x	data frame or matrix
keyVars	column index of key variables
w	column index of sampling weights
importance	weights for each key variable
method	“minimizeSupp” (default), further methods will be included in future versions of the package
k	parameter for k-anonymity.

Details

With the help of this algorithm you can achieve k-anonymity in an optimized way. The procedure set missings only to those key variables for which the importance is greater than 0. Key variables with higher importance will be preferred to be the variable which will be used for suppression of specific values, i.e. the vector of importance assigns to each key variable a weight which is considered by the algorithm.

To guarantee k-anonymity the wrapper of function localSupp2 should be applied (localSupp2Wrapper())

However, if the importance of some key variables are equal to zero, the algorithm may not find a k-anonymity solution (because there isn't any solution reachable at all, for example). The easiest way to overcome this situation is to re-run the algorithm and allow for NA's in some more key variables, i.e. re-run the algorithm with importance greater than 0 for all entries of importance. This will result in k-anonymized results and leads to only few suppressions in the key variables where the importance of the variables are considered.

Method fastSupp avoids some calculation steps but this method is only significantly faster if there is a large data set with few key variables. However, fastSupp leads to an oversuppression (slightly).

Value

Object from class localSupp2.

xAnon	resulting data with suppressions
supps	number of suppressions in the key variables
totalSupps	total number of suppressions.
anonymity	TRUE, if k-anonymity is achieved
keyVars	index of the key variables.
importance	weight vector for key variables
k	k for k-anonymity

Note

fix me: Implementation in C and interface to R.

Author(s)

Matthias Templ, Bernhard Meindl

References

Templ, M. *Statistical Disclosure Control for Microdata Using the R-Package sdcMicro*, Transactions on Data Privacy, vol. 1, number 2, pp. 67-85, 2008. <http://www.tdp.cat/issues/abs.a004a08.php>

See Also

`freqCalc`, `localSupp`

Examples

```
## example from Capobianchi, Polettini and Lucarelli:
data(franccat)
l1 <- localSupp2(franccat, keyVars=c(2,4,5,6), w=8)
l1
l1$x
l2 <- localSupp2(franccat, keyVars=c(2,4,5,6), w=8, k=2)
l3 <- localSupp2(franccat, keyVars=c(2,4,5,6), w=8, k=4)

## long computation time, wait some seconds to get an information
## about the estimated computing time.
## l = localSupp2(free1, keyVars=1:3, w=30, k=2, importance=c(0.1,1,0.8))
```

localSupp2Wrapper *Local Suppression 2*

Description

A wrapper function for function `localSupp2` in order to guarantee k-anonymity.

Usage

```
localSupp2Wrapper(x, keyVars, w, importance=rep(1, length(keyVars)), method="minimizeSupp")
```

Arguments

<code>x</code>	data frame or matrix
<code>keyVars</code>	column index of key variables
<code>w</code>	column index of sampling weights
<code>importance</code>	weights for each key variable, see ‘ <code>localSupp2()</code> ’
<code>method</code>	“ <code>minimizeSupp</code> ” (default), further methods will be included in future versions of the package
<code>kAnon</code>	parameter for k-anonymity.

Details

This wrapper function guarantees k-anonymity. If function `localSupp2()` cannot reach k-anonymity, `localSupp2` must be re-run on the previous results as long as k-anonymity is reached. If k-anonymity cannot be achieved (because the entries of parameter importance includes too much zeros) the function breaks after a sub-optimal solution is obtained.

Value

Object from class `localSupp2`.

<code>xAnon</code>	resulting data with suppressions
<code>supps</code>	number of suppressions in the key variables
<code>totalSupps</code>	total number of suppressions.
<code>anonymity</code>	TRUE, if k-anonymity is achieved
<code>keyVars</code>	index of the key variables.
<code>importance</code>	weight vector for key variables
<code>kAnon</code>	k for k-anonymity

Note

fix me: Implementation in C and interface to R.

Author(s)

Bernhard Meindl, Matthias Templ

References

Templ, M. *Statistical Disclosure Control for Microdata Using the R-Package sdcMicro*, Transactions on Data Privacy, vol. 1, number 2, pp. 67-85, 2008. <http://www.tdp.cat/issues/abs.a004a08.php>

See Also

[freqCalc](#), [localSupp](#)

Examples

```
## example from Capobianchi, Polettini and Lucarelli:
## same results as localSupp2
data(franccat)
localSupp2Wrapper(franccat, keyVars=c(2,4,5,6), w=8)
localSupp2Wrapper(franccat, keyVars=c(2,4,5,6), w=8, k=2)
localSupp2Wrapper(franccat, keyVars=c(2,4,5,6), w=8, k=4)

## we want to avoid missings in column 5:
l1 <- localSupp2Wrapper(franccat, keyVars=c(2,4,5,6), importance=c(1,1,0,1), w=8, kAnon=1)
l1$x
## we want to avoid missings in column 5 and allow missings in 1 only if
```

```
## is really necessary:
l1 <- localSupp2Wrapper(franccdat, keyVars=c(2,4,5,6), importance=c(0.1,1,0,1), w=8, kAnon=1)
l1$x

## long computation time, wait some seconds to get an information
## about the estimated computing time.
## l = localSupp2(free1, keyVars=1:3, w=30, k=2, importance=c(0.1,1,0.8))
```

microaggregation *Microaggregation*

Description

Function to perform various methods of microaggregation.

Usage

```
microaggregation(x, method = "pca", aggr = 3, nc = 8, clustermethod = "clara", opt
```

Arguments

x	data frame or matrix
method	pca, onedims, single, simple, clustpca, pppca, clustpppca, mdav, clustmcdpca, influence, mcdpca
aggr	aggregation level (default=3)
nc	number of cluster, if the chosen method performs cluster analysis
clustermethod	clustermethod, if necessary
opt	
measure	aggregation statistic, mean, median, trim, onestep (default = mean)
trim	trimming percentage, if measure=trim
varsort	variable for sorting, if method= single
transf	transformation for data x
blow	if TRUE, the microaggregated data will have the same dimension as the original data set
blowxm	the microaggregated data with the same dimension as the original one.

Details

On <http://neon.vb.cbs.nl/casc/Glossary.htm> one can found the “official” definition of microaggregation:

Records are grouped based on a proximity measure of variables of interest, and the same small groups of records are used in calculating aggregates for those variables. The aggregates are released instead of the individual record values.

While for the proximity measure very different concepts can be used, the aggregation itself is naturally done with the arithmetic mean. Nevertheless, other measures of location can be used for aggregation, especially when the group size for aggregation has been taken higher than 3. Since the median seems to be unsuitable for microaggregation because of being highly robust, other measures which are included can be chosen.

This function contains also a method with which the data can be clustered with a variety of different clustering algorithms. Clustering observations before applying microaggregation might be useful. Note, that the data are automatically standardised before clustering.

The usage of clustering method 'Mclust' requires package mclust02, which must be loaded first. The package is not loaded automatically, since the package is not under GPL but comes with a different licence.

There are also some projection methods for microaggregation included. The robust version 'pppca' or 'clustpppca' (clustering at first) are fast implementations and provide almost everytime the best results.

Univariate statistics are preserved best with the individual ranking method (we called them 'onedims', however, often this method is named 'individual ranking'), but multivariate statistics are strongly affected.

With method 'simple' one can apply microaggregation directly on the (unsorted) data. It is useful for the comparison with other methods as a benchmark, i.e. replies the question how much better is a sorting of the data before aggregation.

If `blow` is set to `FALSE`, the result will be a data set with dimension `n` divided by `aggr`.

Value

<code>x</code>	original data
<code>method</code>	method
<code>clustering</code>	TRUE, if a clustering is done before microaggregation
<code>aggr</code>	aggregation level
<code>nc</code>	number of clusters, if a clustering method is chosen
<code>xm</code>	aggregated data set
<code>roundxm</code>	rounded aggregated data set (to integers)
<code>clustermethod</code>	clustermethod, if a cluster method is chosen
<code>measure</code>	proximity measure for aggregation
<code>trim</code>	trimming, if proximity measure 'trim' is chosen
<code>varsort</code>	information about the variable which is chosen when using method 'single'
<code>transf</code>	transformation used, when clustering is applied first
<code>blow</code>	TRUE, <code>blowxm</code> is calculated
<code>blowxm</code>	microaggregated data with the same dimension as the original data set
<code>fot</code>	correction factor, necessary if totals calculated and <code>n</code> divided by <code>aggr</code> is not an integer.

Author(s)

Matthias Templ

References

http://www.springerlink.com/content/v257655u88w2/?sortorder=asc\&p_o=20

Templ, M. and Meindl, B., *Robust Statistics Meets SDC: New Disclosure Risk Measures for Continuous Microdata Masking*, Lecture Notes in Computer Science, Privacy in Statistical Databases, vol. 5262, pp. 113-126, 2008.

Templ, M. *Statistical Disclosure Control for Microdata Using the R-Package sdcMicro*, Transactions on Data Privacy, vol. 1, number 2, pp. 67-85, 2008. <http://www.tdp.cat/issues/abs.a004a08.php>

Templ, M. *New Developments in Statistical Disclosure Control and Imputation: Robust Statistics Applied to Official Statistics*, Suedwestdeutscher Verlag fuer Hochschulschriften, 2009, ISBN: 3838108280, 264 pages.

See Also

[summary.micro](#), [plotMicro](#), [valTable](#)

Examples

```
data(Tarragona)
m1 <- microaggregation(Tarragona, method="onedims", aggr=3)
## summary(m1)
```

microData

microData

Description

Small artificial data set for demonstration.

Usage

```
data(microData)
```

Format

The format is: num [1:13, 1:5] 5 7 2 1 7 8 12 3 15 4 ... - attr(*, "dimnames")=List of 2 ..: chr[1 : 13]"10000""11000""12000""12100"..... : chr [1:5] "one" "two" "three" "four" ...

Examples

```
data(microData)
m1 <- microaggregation(microData, method="mdav")
x <- m1$x ## fix me
summary(m1)
```

plot.indivRisk *plot method for indivRisk objects*

Description

Plots an interactive histogram or ecdf plot with various interactive sliders.

Usage

```
## S3 method for class 'indivRisk':  
plot(x, ...)
```

Arguments

x object of class 'indivRisk'
...

Details

With the sliders one can move the individual risk threshold. By this movement the threshold will be moved on the plot and the slider with a re-identification rate and the slider of the number of unsafe records (based on your chosen threshold) are also moved based on the individual risk threshold. This plot is very similar to the individual risk plot of the software mu-Argus.

Author(s)

Matthias Templ

References

look e.g. on the mu-Argus manuals available at <http://neon.vb.cbs.nl/casc/Software/MuManual4.1.pdf>

Templ, M. *Statistical Disclosure Control for Microdata Using the R-Package sdcMicro*, Transactions on Data Privacy, vol. 1, number 2, pp. 67-85, 2008. <http://www.tdp.cat/issues/abs.a004a08.php>

See Also

[indivRisk](#)

Examples

```
## example from Capobianchi, Polettini and Lucarelli:  
data(franmdat)  
ff <- freqCalc(franmdat, keyVars=c(2,4,5,6),w=8)  
irisk <- indivRisk( ff )  
## and now apply:  
## plot(irisk)
```

```
data(free1)
ff <- freqCalc(free1, keyVars=1:3, w=30)
irisk2 <- indivRisk(ff)
## and now apply:
## plot(irisk2)
```

plot.localSupp2 *plot method for localSupp2 objects*

Description

Special barplot for objects from class localSupp2.

Usage

```
## S3 method for class 'localSupp2':
plot(x, ...)
```

Arguments

x object of class 'localSupp2'
...

Details

Just look at the resulting plot.

Author(s)

Matthias Templ

See Also

[localSupp2](#), [localSupp2Wrapper](#)

Examples

```
## example from Capobianchi, Polettini and Lucarelli:
data(franccdat)
l1 <- localSupp2(franccdat, keyVars=c(2,4,5,6), w=8)
l1
plot(l1)
```

`plotMicro`*Comparison plots*

Description

Plots for the comparison of the original data and perturbed data.

Usage

```
plotMicro(x, p, which.plot = 1:3)
```

Arguments

<code>x</code>	object from class <code>micro</code>
<code>p</code>	necessary parameter for the box cox transformation (<code>lambda</code>)
<code>which.plot</code>	which plot should be created? 1: density traces, 2: parallel boxplots, 3: differences in totals

Details

Univariate and multivariate comparison plots are implemented to detect differences between the perturbed and the original data, but also to compare perturbed data which are produced by different methods.

Author(s)

Matthias Templ

References

Templ, M. and Meindl, B., *Software Development for SDC in R*, Lecture Notes in Computer Science, Privacy in Statistical Databases, vol. 4302, pp. 347-359, 2006.

See Also

[microaggregation](#)

Examples

```
data(free1)
m1 <- microaggregation(free1[, 31:34], method="onedims", aggr=3)
m2 <- microaggregation(free1[, 31:34], method="pca", aggr=3)
plotMicro(m1, 0.1, which.plot=1)
```

pram

Post RAndomisation Method (PRAM)

Description

PRAM is a probabilistic, perturbative method which can be applied on categorical variables.

Usage

```
pram(x, pd=0.8, alpha=0.5)
```

Arguments

x	a numeric vector or factor
pd	minimum diagonal entries for the generated transition matrix P. Either a vector of length 1 or a vector of length (number of categories).
alpha	amount of perturbation for the invariant Pram method

Details

The method is implemented exactly as described in the citation in the references. First a transition matrix is created in that way, that the diagonal entries of a matrix P are random numbers between 'pd' and 1. The remaining enties of the matrix are generated such that the rowSums of the matrix is 1. Then a invariant transition matrix is generated.

Value

x	original vector
xpramed	the perturbed vector
pd	randomly generated diagonal entry of the P (between original pd and 1)
Rs	invariant transition matrix
alpha	amount of perturbation for the invariant Pram method

Author(s)

Matthias Templ

References

Shlomo, Natalie and de Waal, Ton (2006) Protection of Micro-data Subject to Edit Constraints Against Statistical Disclosure. Southampton, UK, Southampton Statistical Sciences Research Institute, 36pp. (S3RI Methodology Working Papers, M06/16)

Examples

```
set.seed(123)
x <- sample(1:4, 250, replace=TRUE)
pr1 <- pram(x)
length(which(pr1$x == x))
x2 <- sample(1:4, 250, replace=TRUE)
length(which(pram(x2)$x == x2))

data(free1)
marstatPramed <- pram(free1[, "MARSTAT"])
```

print.freqCalc *Print method for objects from class freqCalc*

Description

Print method for objects from class freqCalc.

Usage

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

Arguments

x object from class freqCalc
...

Value

information about the frequency counts for key variables for object of class 'freqCalc'.

Author(s)

Matthias Templ

See Also

[freqCalc](#)

Examples

```
## example from Capobianchi, Polettini and Lucarelli:
data(franccdat)
f <- freqCalc(franccdat, keyVars=c(2,4,5,6),w=8)
f
```

```
print.indivRisk
```

Print method for objects from class indivRisk

Description

Print method for objects from class indivRisk

Usage

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

Arguments

x object from class indivRisk
...

Value

few information about the method and the final correction factor for objects of class 'indivRisk'.

Author(s)

Matthias Templ

See Also

[indivRisk](#)

Examples

```
## example from Capobianchi, Polettini and Lucarelli:  
data(franccdat)  
f <- freqCalc(franccdat, keyVars=c(2,4,5,6),w=8)  
f  
f$fk  
f$Fk  
## individual risk calculation:  
indivRisk(f)
```

```
print.localSupp2 Print method for objects from class localSupp2
```

Description

Print method for objects from class localSupp2.

Usage

```
## S3 method for class 'localSupp2':  
print(x, ...)
```

Arguments

x object from class localSupp2
...

Value

information about the frequency counts for key variables for object of class 'localSupp2'.

Author(s)

Matthias Templ

See Also

[localSupp2](#)

Examples

```
## example from Capobianchi, Polettini and Lucarelli:  
data(franmdat)  
l1 <- localSupp2(franmdat, keyVars=c(2,4,5,6), w=8)  
l1
```

```
print.micro Print method for objects from class micro
```

Description

Print method for objects from class micro.

Usage

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

Arguments

x object from class micro
...

Value

information about method and aggregation level from objects of class micro.

Author(s)

Matthias Templ

See Also

[microaggregation](#)

Examples

```
data(free1)
m1 <- microaggregation(free1[, 31:34], method="onedims", aggr=3)
m1
```

`print.pram`

Print method for objects from class pram

Description

Print method for objects from class 'pram'.

Usage

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

Arguments

x object from class 'pram'
...

Value

Short information about the method and the parameters used.

Author(s)

Matthias Templ

References

Templ, M. *Statistical Disclosure Control for Microdata Using the R-Package sdcMicro*, Transactions on Data Privacy, vol. 1, number 2, pp. 67-85, 2008. <http://www.tdp.cat/issues/abs.a004a08.php>

See Also

[pram](#)

Examples

```
data(freel)
x <- freel[, "MARSTAT"]
x2 <- pram(x)
x2
```

sdcGUI

GUI for the sdcMicro package

Description

This graphical user interface supports the main functions of sdcMicro.

Usage

```
sdcGUI ()
```

Details

This GUI provides an extension to the package sdcMicro. The developed GUI makes sdcMicro accessible to a wider range of users including ones not used to the R command line interface. The user can access all basic functions for microdata protection by using this GUI.

The graphical user interface of sdcMicro allows an interactive interaction between objects. Flexibility is provided by automatic displaying of the main results (e.g., the summary of the frequency counts and the estimated disclosure risk) which are updated after a user interaction automatically. Additional flexibility is provided by storing all the users operations with all parameters in a script which can then be saved, modified and/or reloaded. Thus, full reproducibility is provided also when using this GUI instead of the CLI version.

It is programmed based on the gWidgetsRGtk2 and RGtk2 package.

Author(s)

Thomas Petelin, Matthias Templ

Examples

```
##sdcGUI ()
```

summary.freqCalc *Summary method for objects from class freqCalc*

Description

Summary method for objects of class 'freqCalc' to provide information about local suppressions.

Usage

```
## S3 method for class 'freqCalc':  
summary(object, ...)
```

Arguments

object object from class freqCalc
...

Details

Shows the amount of local suppressions on each variable in which local suppression was applied.

Value

Information about local suppression in each variable (only if a local suppression is already done).

Author(s)

Matthias Templ

See Also

[freqCalc](#)

Examples

```
## example from Capobianchi, Polettini and Lucarelli:  
data(franccat)  
f <- freqCalc(franccat, keyVars=c(2,4,5,6),w=8)  
f  
f$fk  
f$Fk  
## individual risk calculation:  
indivf <- indivRisk(f)  
indivf$rk  
## Local Suppression  
localS <- localSupp(f, keyVar=2, indivRisk=indivf$rk, threshold=0.25)  
f2 <- freqCalc(localS$freqCalc, keyVars=c(4,5,6), w=8)  
summary(f2)
```

summary.micro *Summary method for objects from class micro*

Description

Summary method for objects from class 'micro'.

Usage

```
## S3 method for class 'micro':
summary(object, ...)
```

Arguments

object objects from class micro
 ...

Details

This function computes several measures of information loss, such as

Value

meanx	A conventional summary of the original data
meanxm	A conventional summary of the microaggregated data
amean	average relative absolute deviation of means
amedian	average relative absolute deviation of medians
aonestep	average relative absolute deviation of onestep from median
devvar	average relative absolute deviation of variances
amad	average relative absolute deviation of the mad
acov	average relative absolute deviation of covariances
arcov	average relative absolute deviation of robust (with mcd) covariances
acor	average relative absolute deviation of correlations
arcor	average relative absolute deviation of robust (with mcd) correlations
acors	average relative absolute deviation of rank-correlations
adlm	average absolute deviation of lm regression coefficients (without intercept)
adlts	average absolute deviation of lts regression coefficients (without intercept)
apcaload	average absolute deviation of pca loadings
appacaload	average absolute deviation of robust (with projection pursuit approach) pca loadings
atotals	average relative absolute deviation of totals
pmtotals	average relative deviation of totals

Author(s)

Matthias Templ

References

Templ, M. *Statistical Disclosure Control for Microdata Using the R-Package sdcMicro*, Transactions on Data Privacy, vol. 1, number 2, pp. 67-85, 2008. <http://www.tdp.cat/issues/abs.a004a08.php>

See Also[microaggregation](#), [valTable](#)**Examples**

```
data(Tarragona)
m1 <- microaggregation(Tarragona, method="onedims", aggr=3)
## summary(m1)
```

`summary.pram`*Summary method for objects from class pram*

Description

Summary method for objects from class ‘pram’ to provide information about transitions.

Usage

```
## S3 method for class 'pram':
summary(object, ...)
```

Arguments

```
object      object from class ‘pram’
...         ...
```

Details

Shows various information about the transitions.

Value

The summary of object from class ‘pram’.

Author(s)

Matthias Templ

References

Templ, M. *Statistical Disclosure Control for Microdata Using the R-Package sdcMicro*, Transactions on Data Privacy, vol. 1, number 2, pp. 67-85, 2008. <http://www.tdp.cat/issues/abs.a004a08.php>

See Also

[pram](#)

Examples

```
data(free1)
x <- free1[, "MARSTAT"]
x2 <- pram(x)
x2
summary(x2)
```

swappNum

Rank Swapping

Description

Rank Swapping.

Usage

```
swappNum(x, w = 1:(dim(x)[2]), p)
```

Arguments

x	matrix or data frame
w	variables, on which rank swapping should be applied
p	Percentage. Swapping range.

Details

The values of a variable are ranked, then each ranked value is swapped with another ranked value randomly chosen within a restricted range, i.e. the rank of two swapped values cannot differ by more than p percente of the total number of records. The function apply the rank swapping on each variable independently.

Value

x	original data
xm	the rank swapped data
method	info about the method name

Author(s)

Matthias Templ

ReferencesLook, e.g. on <http://www.niss.org/dgii/TR/dataswap-finalrevision.pdf>**See Also**[microaggregation](#)**Examples**

```
## Numerical Rank Swapping:
data(free1)
free1[, 31:34] <- swappNum(free1[, 31:34], p=10)
```

Tarragona

Tarragona data set

Description

A real data set comprising figures of 834 companies in the Tarragona area. Data correspond to year 1995.

Usage

```
data(Tarragona)
```

Format

A data frame with 834 observations on the following 13 variables.

FIXED.ASSETS a numeric vector
CURRENT.ASSETS a numeric vector
TREASURY a numeric vector
UNCOMMITTED.FUNDS a numeric vector
PAID.UP.CAPITAL a numeric vector
SHORT.TERM.DEBT a numeric vector
SALES a numeric vector
LABOR.COSTS a numeric vector
DEPRECIATION a numeric vector
OPERATING.PROFIT a numeric vector
FINANCIAL.OUTCOME a numeric vector
GROSS.PROFIT a numeric vector
NET.PROFIT a numeric vector

Source

<http://neon.vb.cbs.nl/casc/testsets.html>

References

Brand, R. and Domingo-Ferrer, J. and Mateo-Sanz, J.M., Reference data sets to test and compare SDC methods for protection of numerical microdata. Unpublished. <http://neon.vb.cbs.nl/casc/CASCrefmicrodata.pdf>

Examples

```
data(Tarragona)
head(Tarragona)
dim(Tarragona)
```

topBotCoding

Top and Bottom Coding

Description

Function for Top and Bottom Coding.

Usage

```
topBotCoding(x, value, replacement, kind = "top")
```

Arguments

x	vector or one-dimensional matrix or data.frame
value	limit, from where it should be top- or bottom-coded
replacement	replacement value.
kind	top or bottom

Details

Extreme values are replaced by one value to reduce the disclosure risk.

Value

Top or bottom coded data.

Author(s)

Matthias Templ

See Also

[indivRisk](#)

Examples

```
data(free1)
topBotCoding(free1[, "DEBTS"], value=9000, replacement=9100, kind="top")
```

valTable *Comparison of different microaggregation methods*

Description

A Function for the comparison of different perturbation methods.

Usage

```
valTable(x, method = c("simple", "onedims", "clustpppca", "addNoise: additive", "sw"))
```

Arguments

x	data frame or matrix
method	microaggregation methods or adding noise methods or rank swapping.
measure	FUN for aggregation. Possible values are mean (default), median, trim, onestep.
clustermethod	clustermethod, if a method will need a clustering procedure
aggr	aggregation level (default=3)
nc	number of clusters. Necessary, if a method will need a clustering procedure
transf	Transformation of variables before clustering.
p	Swapping range, if method swappNum has been chosen
noise	noise addition, if an addNoise method has been chosen
w	variables for swapping, if method swappNum has been chosen
delta	parameter for adding noise method 'correlated2'

Details

Tabularise the output from summary.micro. Will be enhanced to all perturbation methods in future versions.

Value

Measures of information loss splitted for the comparison of different methods.

Methods for adding noise should be named via "addNoise: method", e.g. "addNoise: correlated", i.e. the term 'at first' then followed by a ':' and a blank and then followed by the name of the method as described in function 'addNoise'.

Author(s)

Matthias Templ

References

Templ, M. and Meindl, B., *Software Development for SDC in R*, Lecture Notes in Computer Science, Privacy in Statistical Databases, vol. 4302, pp. 347-359, 2006.

See Also

[microaggregation](#), [summary.micro](#)

Examples

```
data(Tarragona)
## valTable(Tarragona[100:200,], method=c("simple", "onedims", "pca", "addNoise: additive"))
## valTable(Tarragona, method=c("simple", "onedims", "pca", "clustpppca", "mdav", "addNoise: add
## clustpppca in combination with Mclust outperforms the other algorithms for this data set.
```

Index

*Topic **IO**

sdCGUI, 37

*Topic **aplot**

plot.indivRisk, 29
plot.localSupp2, 30
plotMicro, 31

*Topic **datasets**

cascl, 6
CASCrefermicrodata, 7
EIA, 13
francdat, 16
freel, 17
microData, 28
Tarragona, 42

*Topic **manip**

addNoise, 5
dataGen, 8
dRisk, 9
dRiskRMD, 10
dUtility, 11
freqCalc, 17
globalRecode, 19
indivRisk, 20
localSupp, 21
localSupp2, 22
localSupp2Wrapper, 24
microaggregation, 26
pram, 32
swappNum, 41
topBotCoding, 43

*Topic **package**

sdCMicro-package, 1

*Topic **print**

print.freqCalc, 33
print.indivRisk, 34
print.localSupp2, 35
print.micro, 35
print.pram, 36
summary.freqCalc, 38

summary.micro, 39
summary.pram, 40
valTable, 44

addNoise, 5

cascl, 6
CASCrefermicrodata, 7
cut, 20

dataGen, 8
dRisk, 9, 11, 12
dRiskRMD, 10, 12
dUtility, 9, 11

EIA, 13

francdat, 16
freel, 17
freqCalc, 17, 21, 22, 24, 25, 33, 38

globalRecode, 19

indivRisk, 18, 20, 22, 29, 34, 43

localSupp, 21, 24, 25
localSupp2, 22, 30, 35
localSupp2Wrapper, 24, 30

microaggregation, 26, 31, 36, 40, 42, 45
microData, 28

plot.indivRisk, 29
plot.localSupp2, 30
plotMicro, 28, 31
pram, 32, 37, 41
print.freqCalc, 33
print.indivRisk, 34
print.localSupp2, 35
print.micro, 35
print.pram, 36

sdCGUI, [37](#)
sdcMicro (*sdcMicro-package*), [1](#)
sdcMicro-package, [1](#)
summary.freqCalc, [38](#)
summary.micro, [6](#), [28](#), [39](#), [45](#)
summary.pram, [40](#)
swappNum, [41](#)

Tarragona, [42](#)
topBotCoding, [43](#)

valTable, [28](#), [40](#), [44](#)