Package ‘modi’

November 20, 2018

Type Package

Title Multivariate Outlier Detection and Imputation for Incomplete Survey Data

Version 0.1.0

Description Algorithms for multivariate outlier detection when missing values occur. Algorithms are based on Mahalanobis distance or data depth. Imputation is based on the multivariate normal model or uses nearest neighbour donors. The algorithms take sample designs, in particular weighting, into account. The methods are described in Bill and Hulliger (2016) <doi:10.17713/ajs.v45i1.86>.

License MIT + file LICENSE

URL https://github.com/martinSter/modi

BugReports https://github.com/martinSter/modi/issues

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports MASS (>= 7.3-50), norm (>= 1.0-9.5), stats, graphics, utils

RoxygenNote 6.0.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Beat Hulliger [aut],
Martin Sterchi [cre]

Maintainer Martin Sterchi <martin.sterchi@fhnw.ch>

Repository CRAN

Date/Publication 2018-11-20 13:10:07 UTC
BEM 

R topics documented:

- BEM .......................................................... 2
- bushfire ..................................................... 4
- bushfire.weights ........................................... 5
- bushfirem .................................................... 5
- EAdet ......................................................... 6
- EApr .......................................................... 9
- ER .............................................................. 11
- GIMCD ......................................................... 12
- MDmiss ....................................................... 13
- modi .......................................................... 14
- PlotMD ........................................................ 15
- POEM .......................................................... 16
- sepe .......................................................... 18
- TRC ............................................................ 19
- weighted.quantile ........................................ 21
- weighted.var ................................................ 22
- Winsimp ...................................................... 23

Index 25

BEM  BACON-EEM Algorithm for multivariate outlier detection in incomplete multivariate survey data

Description

BEM starts from a set of uncontaminated data with possible missing values, applies a version of the EM-algorithm to estimate the center and scatter of the good data, then adds (or deletes) observations to the good data which have a Mahalanobis distance below a threshold. This process iterates until the good data remain stable. Observations not among the good data are outliers.

Usage

BEM(data, weights, v = 2, c0 = 3, alpha = 0.01, md.type = "m",
    em.steps.start = 10, em.steps.loop = 5, better.estimation = FALSE,
    monitor = FALSE)

Arguments

data     a matrix or data frame. As usual, rows are observations and columns are variables.
weights  a non-negative and non-zero vector of weights for each observation. Its length must equal the number of rows of the data. Default is rep(1, nrow(data)).

v        an integer indicating the distance for the definition of the starting good subset: v = 1 uses the Mahalanobis distance based on the weighted mean and covariance, v = 2 uses the Euclidean distance from the componentwise median.
c0 the size of initial subset is $c0 \times ncol(data)$.
alpha a small probability indicating the level $(1 - \alpha)$ of the cutoff quantile for good observations.
md.type type of Mahalanobis distance: "m" marginal, "c" conditional.
em.steps.start number of iterations of EM-algorithm for starting good subset.
em.steps.loop number of iterations of EM-algorithm for good subset.
better.estimation if better.estimation = TRUE, then the EM-algorithm for the final good subset iterates em.steps.start more.
monitor if TRUE, verbose output.

Details

The BACON algorithm with $v = 1$ is not robust but affine equivariant while $v = 1$ is robust but not affine equivariant. The threshold for the (squared) Mahalanobis distances, beyond which an observation is an outlier, is a standardised chisquare quantile at $(1 - \alpha)$. For large data sets it may be better to choose alpha / n instead. The internal function EM.normal is usually called from BEM. EM.normal is implementing the EM-algorithm in such a way that part of the calculations can be saved to be reused in the BEM algorithm. EM.normal does not contain the computation of the observed sufficient statistics, they will be computed in the main program of BEM and passed as parameters as well as the statistics on the missingness patterns.

Value

BEM returns a list whose first component output is a sublist with the following components:

- sample.size Number of observations
- discarded.observations Number of discarded observations
- number.of.variables Number of variables
- significance.level The probability used for the cutpoint, i.e. alpha
- initial.basic.subset.size Size of initial good subset
- final.basic.subset.size Size of final good subset
- number.of.iterations Number of iterations of the BACON step
- computation.time Elapsed computation time
- center Final estimate of the center
- scatter Final estimate of the covariance matrix
- cutpoint The threshold MD-value for the cut-off of outliers

The further components returned by BEM are:

- outind Indicator of outliers
- dist Final Mahalanobis distances

Note

BEM uses an adapted version of the EM-algorithm in function .EM.normal.
**Author(s)**

Beat Hulliger

**References**


**Examples**

```r
# Bushfire data set with 20% MCAR
data(bushfire, bushfire.weights)

bem.res <- BEM(bushfire, bushfire.weights, alpha = (1 - 0.01 / nrow(bushfire)))
print(bem.res$output)
```

---

**bushfire**

_Bushfire scars._

**Description**

The bushfire data set was used by Campbell (1984, 1989) to locate bushfire scars. The dataset contains satellite measurements on five frequency bands, corresponding to each of 38 pixels.

**Usage**

bushfire

**Format**

A data frame with 38 rows and 5 variables.

**Details**

The data contains an outlying cluster of observations 33 to 38 a second outlier cluster of observations 7 to 11 and a few more isolated outliers, namely observations 12, 13, 31 and 32.

For testing purposes weights are provided: `bushfire.weights <- rep(c(1,2,5), length = nrow(bushfire))`

**References**

**bushfire.weights**

Examples

```r
data(bushfire)
```

---

**bushfire.weights**  
*Weights for Bushfire scars.*

**Description**

The bushfire data set was used by Campbell (1984, 1989) to locate bushfire scars. The dataset contains satellite measurements on five frequency bands, corresponding to each of 38 pixels.

**Usage**

```r
bushfire.weights
```

**Format**

A vector of length 38.

**Details**

For testing purposes, `bushfire.weights` provides artificial weights created according to:

```r
bushfire.weights <- rep(c(QLRLUIL length nrow(bushfireII
```

**References**


**Examples**

```r
data(bushfire.weights)
```

---

**bushfirem**  
*Bushfire scars with missing data.*

**Description**

The bushfire data set was used by Campbell (1984, 1989) to locate bushfire scars. The dataset contains satellite measurements on five frequency bands, corresponding to each of 38 pixels. However, this dataset contains missing values.

**Usage**

```r
bushfirem
```
Format

A data frame with 38 rows and 5 variables.

Details

The data contains an outlying cluster of observations 33 to 38 a second outlier cluster of observations 7 to 11 and a few more isolated outliers, namely observations 12, 13, 31 and 32.
bushfirem is created from bushfire by setting a proportion of 0.2 of the values to missing.
For testing purposes weights are provided: bushfire.weights <- rep(c(1, 2, 5), length = nrow(bushfire))

References


Examples

data(bushfirem)

\begin{verbatim}
EAdet

Description

In EAdet an epidemic is started at a center of the data. The epidemic spreads out and infects neighbouring points (probabilistically or deterministically). The last points infected are outliers. After running EAdet an imputation with EAimp may be run.

Usage

EAdet(data, weights, reach = "max", transmission.function = "root", power = ncol(data), distance.type = "euclidean", maxl = 5, plotting = TRUE, monitor = FALSE, prob.quantile = 0.9, random.start = FALSE, fix.start, threshold = FALSE, deterministic = TRUE, rm.missobs = FALSE, verbose = FALSE)

Arguments

data a data frame or matrix with data.
weights a vector of positive sampling weights.
reach if reach = "max" the maximal nearest neighbor distance is used as the basis for the transmission function, otherwise the weighted $(1 - (p + 1)/n)$ quantile of the nearest neighbor distances is used.

\end{verbatim}
transmission.function

form of the transmission function of distance d: "step" is a heaviside function which jumps to 1 at d0. "linear" is linear between 0 and d0, "power" is \((\beta d + p)^{-}\) for \(p = \text{ncol(data)}\) as default, "root" is the function \(1-(1-d/d0)^{1/(1/mx1)}\).

power

sets \(p = \text{power}\).

distance.type

distance type in function \(\text{dist()}\).

max1

maximum number of steps without infection.

plotting

if \(\text{TRUE}\), the cdf of infection times is plotted.

monitor

if \(\text{TRUE}\), verbose output on epidemic.

prob.quantile

if mads fail, take this quantile absolute deviation.

random.start

if \(\text{TRUE}\), take a starting point at random instead of the spatial median.

fix.start

force epidemic to start at a specific observation.

threshold

infect all remaining points with infection probability above the threshold \(1-0.5^*(1/mx1)\).

deterministic

if \(\text{TRUE}\), the number of infections is the expected number and the infected observations are the ones with largest infection probabilities.

rm.missobs

set \(\text{rm.missobs=TRUE}\) if completely missing observations should be discarded. This has to be done actively as a safeguard to avoid mismatches when imputing.

verbose

more output with \(\text{verbose=TRUE}\).

Details

The form and parameters of the transmission function should be chosen such that the infection times have at least a range of 10. The default cutting point to decide on outliers is the median infection time plus three times the mad of infection times. A better cutpoint may be chosen by visual inspection of the cdf of infection times. EAdet calls the function EA.dist, which passes the counterprobabilities of infection (a \(n \times (n - 1)/2\) size vector!) and three parameters (sample spatial median index, maximal distance to nearest neighbor and transmission distance = reach) as arguments to EAdet. The distances vector may be too large to be passed as arguments. Then either the memory size must be increased. Former versions of the code used a global variable to store the distances in order to save memory.

Value

EAdet returns a list whose first component output is a sub-list with the following components:

sample.size

Number of observations

discarded.observations

Indices of discarded observations

missing.observations

Indices of completely missing observations

number.of.variables

Number of variables

n.complete.records

Number of records without missing values

n.usable.records

Number of records with less than half of values missing (unusable observations are discarded)

medians

Component wise medians
mads  Component wise mads
prob.quantile  Use this quantile if mads fail, i.e. if one of the mads is 0
quantile.deviations  Quantile of absolute deviations
start  Starting observation
transmission.function  Input parameter
power  Input parameter
maxl  Maximum number of steps without infection
min.nn.dist  Maximal nearest neighbor distance
transmission.distance  d0
threshold  Input parameter
distance.type  Input parameter
deterministic  Input parameter
number.infected  Number of infected observations
cutpoint  Cutpoint of infection times for outlier definition
number.outliers  Number of outliers
outliers  Indices of outliers
duration  Duration of epidemic
computation.time  Elapsed computation time
initialisation.computation.time  Elapsed computation time for standardisation and calculation of distance matrix

The further components returned by EAdet are:

infected  Indicator of infection
infection.time  Time of infection
outind  Indicator of outliers

Author(s)

Beat Hulliger

References


See Also

EAdet for imputation with the Epidemic Algorithm.

Examples

data(bushfirem, bushfire.weights)
det.res <- EAdet(bushfirem, bushfire.weights)
print(det.res$output)
EAIMp

Epidemic Algorithm for imputation of multivariate outliers in incomplete survey data.

Description

After running EAdet an imputation of the detected outliers with EAIMp may be run.

Usage

EAIMp(data, weights, outind, reach = "max", transmission.function = "root", power = ncol(data), distance.type = "euclidean", duration = 5, maxl = 5, kdon = 1, monitor = FALSE, threshold = FALSE, deterministic = TRUE, fixedprop = 0)

Arguments

data a data frame or matrix with the data.
weights a vector of positive sampling weights.
outind a logical vector with component TRUE for outliers.
reach reach of the threshold function (usually set to the maximum distance to a nearest neighbour, see internal function EA.dist).
transmission.function form of the transmission function of distance d: "step" is a heaviside function which jumps to 1 at d=0, "linear" is linear between 0 and d0, "power" is beta*d+1^(-p) for p=ncol(data) as default, "root" is the function 1-(1-d/d0)^(1/maxl).
power sets p=power, where p is the parameter in the above transmission function.
distance.type distance type in function dist().
duration the duration of the detection epidemic.
maxl maximum number of steps without infection.
kdon the number of donors that should be infected before imputation.
monitor if TRUE verbose output on epidemic.
threshold Infect all remaining points with infection probability above the threshold 1-0.5^(1/maxl).
deterministic if TRUE the number of infections is the expected number and the infected observations are the ones with largest infection probabilities.
fixedprop if TRUE a fixed proportion of observations is infected at each step.

Details

EAIMp uses the distances calculated in EAdet (actually the counterprobabilities, which are stored in a global data set) and starts an epidemic at each observation to be imputed until donors for the missing values are infected. Then a donor is selected randomly.
Value

Eaimp returns a list with two components: parameters and imputed.data. parameters contains the following elements:

- sample.size Number of observations
- number.of.variables Number of variables
- n.complete.records Number of records without missing values
- n.usable.records Number of records with less than half of values missing (unusable observations are discarded)
- duration Duration of epidemic
- reach Transmission distance (d0)
- threshold Input parameter
- deterministic Input parameter
- computation.time Elapsed computation time

imputed.data contains the imputed data.

Author(s)

Beat Hulliger

References


See Also

EAdet for outlier detection with the Epidemic Algorithm.

Examples

data(bushfiremL, bushfireNweights)
det.res <- EAdet(bushfiremL, bushfireNweights)
imp.res <- Eaimp(bushfiremL, bushfireNweights, outind = det.resNoutputNind, reach = det.resNoutputNmaxNminNdi, kdon = 3)
print(imp.resNoutput)
Robust EM-algorithm ER

Description

The ER function is an implementation of the ER-algorithm of Little and Smith (1987).

Usage

ER(data, weights, alpha = 0.01, psi.par = c(2, 1.25), em.steps = 100,
    steps.output = FALSE, Estep.output = FALSE, tolerance = 1e-06)

Arguments

data a data frame or matrix with the data.
weights sampling weights.
alpha probability for the quantile of the cut-off.
psi.par further parameters passed to the psi-function.
em.steps number of iteration steps of the EM-algorithm.
steps.output if TRUE, verbose output.
Estep.output if TRUE, estimators are output at each iteration.
tolerance convergence criterion (relative change).

Details

The M-step of the EM-algorithm uses a one-step M-estimator.

Value

sample.size Number of observations
number.of.variables Number of variables
significance.level alpha
computation.time Elapsed computation time
good.data Indices of the data in the final good subset
outliers Indices of the outliers
center Final estimate of the center
scatter Final estimate of the covariance matrix
dist Final Mahalanobis distances
rob.weights Robustness weights in the final EM step

Author(s)

Beat Hulliger
References


See Also

BEM

Examples

data(bushfirem, bushfire.weights)
det.res <- ER(bushfirem, weights = bushfire.weights, alpha = 0.05, steps.output = TRUE, em.steps = 100, tol = 2e-6)
PlotMD(det.res$dist, ncol(bushfirem))

GIMCD

Gaussian imputation followed by MCD

Description

Gaussian imputation uses the classical non-robust mean and covariance estimator and then imputes predictions under the multivariate normal model. Outliers may be created by this procedure. Then a high-breakdown robust estimate of the location and scatter with the Minimum Covariance Determinant algorithm is obtained and finally outliers are determined based on Mahalanobis distances based on the robust location and scatter.

Usage

GIMCD(data, alpha = 0.05, seedem = 23456789, seedmcd)

Arguments

data a data frame or matrix with the data.
alpha a threshold value for the cut-off for the outlier Mahalanobis distances.
seedem random number generator seed for EM algorithm
seedmcd random number generator seed for MCD algorithm, if seedmcd is missing, an internal seed will be used.

Details

Normal imputation from package norm and MCD from package MASS. Note that currently MCD does not accept weights.
**MDmiss**

**Value**

Result is stored in a global list GIMCD.r:

- `center` robust center
- `scatter` robust covariance
- `alpha` quantile for cut-off value
- `computation.time` elapsed computation time
- `outind` logical vector of outlier indicators
- `dist` Mahalanobis distances

**Author(s)**

Beat Hulliger

**References**


**See Also**

cov.rob

**Examples**

```r
data(bushfirem)
det.res <- GIMCD(bushfirem, alpha = 0.1)
print(det.res$center)
PlotMD(det.res$dist, ncol(bushfirem))
```

---

**MDmiss**

*Mahalanobis distance (MD) for data with missing values*

**Description**

For each observation the missing dimensions are omitted before calculating the MD. The MD contains a correction factor \( \frac{p}{q} \) to account for the number of observed values, where \( p \) is the number of variables and \( q \) is the number of observed dimensions for the particular observation.

**Usage**

```r
MDmiss(data, center, cov)
```
Arguments

data the data as a dataframe or matrix.
center the center to be used (may not contain missing values).
cov the covariance to be used (may not contain missing values).

Details

The function loops over the observations. This is not optimal if only a few missingness patterns occur. If no missing values occur the function returns the Mahalanobis distance.

Value

The function returns a vector of the (squared) Mahalanobis distances.

Author(s)

Beat Hulliger

References


See Also

mahalanobis

Examples

data(bushfiremL, bushfire)
MDmiss(bushfiremL, apply(bushfire, 2, mean), var(bushfire))

modi: Multivariate outlier detection for incomplete survey data.

Description

The package modi is a collection of functions for multivariate outlier detection and imputation. The aim is to provide a set of functions which cope with missing values and take sampling weights into account. The original functions were developed in the EUREDIT project. This work was partially supported by the EU FP5 ICT programme, the Swiss Federal Office of Education and Science and the Swiss Federal Statistical Office. Subsequent development was in the AMELI project of the EU FP7 SSH Programme and also supported by the University of Applied Sciences and Arts Northwestern Switzerland (FHNW).
modi functions

BACON-EEM algorithm in BEM(), Epidemic algorithm in EAdet() and EAimp(), Transformed Rank Correlations in TRC(), Gaussian imputation with MCD in GIMCD().

References


plotMD

QQ-Plot of Mahalanobis distances

Description

QQ-plot of (squared) Mahalanobis distances vs. scaled F-distribution (or a scaled chisquare distribution). In addition, two default cutpoints are proposed.

Usage

plotMD(dist, p, alpha = 0.95, chisquare = FALSE)

Arguments

dist a vector of Mahalanobis distances.
p the number of variables involved in the Mahalanobis distances.
alpha a probability for cut-off, usually close to 1.
chisquare a logical indicating the the chisquare distribution should be used instead of the F-distribution.

Details

Scaling of the F-distribution as median(dist)*qf((1/n)/(n+1), p, n-p)/qf(0.5, p, n-p). First default cutpoint is median(dist)*qf(alpha, p, n-p)/qf(0.5, p, n-p) and the second default cutpoint is the alpha quantile of the Mahalanobis distances.

Value

hmed first proposed cutpoint based on F-distribution
halpha second proposed cutpoint (alpha-quantile)

QQ-plot
Author(s)

Beat Hulliger

References


Examples

data(bushfirem, bushfire.weights)
det.res <- TRC(bushfirem, weights = bushfire.weights)
PlotMD(det.res$dist, ncol(bushfirem))

POEM

Nearest Neighbour Imputation with Mahalanobis distance

Description

POEM takes into account missing values, outlier indicators, error indicators and sampling weights.

Usage

POEM(data, weights, outind, errors, missing.matrix, alpha = 0.5, beta = 0.5,
      reweight.out = FALSE, c = 5, preliminary.mean.imputation = FALSE,
      monitor = FALSE)

Arguments

data a data frame or matrix with the data.
weights sampling weights.
outind an indicator vector for the outliers with 1 indicating an outlier.
errors matrix of indicators for items which failed edits.
missing.matrix the missingness matrix can be given as input. Otherwise, it will be recalculated.
alpha scalar giving the weight attributed to an item that is failing.
beta minimal overlap to accept a donor.
reweight.out if TRUE, the outliers are redefined.
c tuning constant when redefining the outliers (cutoff for Mahalanobis distance).
preliminary.mean.imputation assume the problematic observation is at the mean of good observations.
monitor if TRUE verbose output.
Details

**POEM** assumes that an multivariate outlier detection has been carried out beforehand and assumes the result is summarized in the vector `outind`. In addition, further observations may have been flagged as failing edit-rules and this information is given in the vector `errors`. The mean and covariance estimate is calculated with the good observations (no outliers and downweighted errors). Preliminary mean imputation is sometimes needed to avoid a non-positive definite covariance estimate at this stage. Preliminary mean imputation assumes that the problematic values of an observation (with errors, outliers or missing) can be replaced by the mean of the rest of the non-problematic observations. Note that the algorithm imputes these problematic observations afterwards and therefore the final covariance matrix with imputed data is not the same as the working covariance matrix (which may be based on preliminary mean imputation).

Value

**POEM** returns a list whose first component `output` is a sub-list with the following components:

- `preliminary.mean.imputation` Logical. TRUE if preliminary mean imputation should be used
- `completely.missing` Number of observations with no observed values
- `good.values` Weighted number of of good values (not missing, not outlying, not erroneous)
- `nonoutliers.before` Number of nonoutliers before reweighting
- `weighted.nonoutliers.before` Weighted number of nonoutliers before reweighting
- `nonoutliers.after` Number of nonoutliers after reweighting
- `weighted.nonoutliers.after` Weighted number of nonoutliers after reweighting
- `old.center` Coordinate means after weighting, before imputation
- `old.variances` Coordinate variances after weighting, before imputation
- `new.center` Coordinate means after weighting, after imputation
- `new.variances` Coordinate variances after weighting, after imputation
- `covariance` Covariance (of standardised observations) before imputation
- `imputed.observations` Indices of observations with imputed values
- `donors` Indices of donors for imputed observations
- `new.outind` Indices of new outliers

The further component returned by **POEM** is:

- `imputed.data` Imputed data set

Author(s)

Beat Hulliger

References

Examples

```r
data(bushfirem, bushfire.weights)
outliers <- rep(0, nrow(bushfirem))
outliers[31:38] <- 1
imp.res <- POEM(bushfirem, bushfire.weights, outliers,
preliminary.mean.imputation = TRUE)
print(imp.res$output)
var(imp.res$imputed.data)
```

Description

The sepe data set is a sample of the pilot survey in 1993 of the Swiss Federal Statistical Office on environment protection expenditures of Swiss private economy in the previous accounting year. The units are enterprises, the monetary variables are in thousand Swiss Francs (CHF). From the original sample a random subsample was chosen of which certain enterprises were excluded for confidentiality reasons. In addition, noise has been added to certain variables, and certain categories have been collapsed. The data set has missing values. The data set has first been prepared for the EU FP5 project EUREDIT and later been data protected for educational purposes.

Usage

```r
sepe
```

Format

A data frame with 675 rows and 23 variables:

- `idnr` identifier (anonymous)
- `exp` categorical variable where 1 = 'non-zero total expenditure' and 2 = 'zero total expenditure, and 3 = 'no answer'
- `totinvwp` total investment for water protection
- `totinvwm` total investment for waste management
- `totinvap` total investment for air protection
- `totinvnp` total investment for noise protection
- `totinvot` total investment for other environmental protection
- `totinvto` overall total investment in all environmental protection areas
- `totexpwp` total current expenditure in environmental protection area water protection
- `totexpwm` total current expenditure in environmental protection area waste management
- `totexpap` total current expenditure in environmental protection area air protection
- `totexpnp` total current expenditure in environmental protection area noise protection
- `totexpot` total current expenditure in other environmental protection


Sample Environment Protection Expenditure Survey.
\texttt{totexpto} overall total current expenditure in all environmental protection
\texttt{subtot} total subsidies for environmental protection received
\texttt{rectot} total receipts from environmental protection
\texttt{employ} number of employees
\texttt{sizeclass} size class (according to number of employees)
\texttt{stratum} stratum number of sample design
\texttt{activity} code of economic activity (aggregated)
\texttt{popsiz}e number of enterprises in the population-stratum
\texttt{popempl} number of employees in population activity group
\texttt{weight} sampling weight (for extrapolation to the population)

Details

The sample design is stratified random sampling with different sampling rates. Use package survey or sampling to obtain correct point and variance estimates. In addition a ratio estimator may be built using the variable \texttt{popempl} which gives the total employment per activity.

There are two balance rules: the subtotals of the investment variables should sum to \texttt{totinvto} and the expenditure subtotals should sum to \texttt{totexpto}.

The missing values stem from the survey itself. In the actual survey the missing values were declared as 'guessed' rather than copied from records.

The sampling weight weight is adjusted for non-response in the stratum, i.e. weight=\texttt{popsiz}/\texttt{sampsize}.

References


Charlton, J. (ed.), Towards Effective Statistical Editing and Imputation Strategies - Findings of the Euredit project, unpublished manuscript available from Eurostat and \url{http://www.cs.york.ac.uk/euredit/}.

Examples

\begin{verbatim}
data(sepe)
\end{verbatim}

---

\texttt{TRC} \hspace{1in} \textit{Transformed rank correlations for multivariate outlier detection}

Description

\texttt{TRC} starts from bivariate Spearman correlations and obtains a positive definite covariance matrix by back-transforming robust univariate medians and mads of the eigenspace. \texttt{TRC} can cope with missing values by a regression imputation using the a robust regression on the best predictor and it takes sampling weights into account.
Usage

TRC(data, weights, overlap = 3, mincor = 0, robust.regression = "rank",
gamma = 0.5, prob.quantile = 0.75, alpha = 0.05, md.type = "m",
monitor = FALSE)

Arguments

data       a data frame or matrix with the data.
weights    sampling weights.
overlap    minimum number of jointly observed values for calculating the rank correlation.
mincor     minimal absolute correlation to impute.
robust.regression
            type of regression: "irls" is iteratively reweighted least squares M-estimator,
                             "rank" is based on the rank correlations.
gamma      minimal number of jointly observed values to impute.
prob.quantile if mads are 0, try this quantile of absolute deviations.
alpha       (1 - alpha) Quantile of F-distribution is used for cut-off.
md.type     type of Mahalanobis distance when missing values occur: "m" marginal (default), "c" conditional.
monitor     if TRUE, verbose output.

Details

TRC is similar to a one-step OGK estimator where the starting covariances are obtained from rank correlations and an ad hoc missing value imputation plus weighting is provided.

Value

TRC returns a list whose first component output is a sublist with the following components:

sample.size Number of observations
number.of.variables Number of variables
number.of.missing.items Number of missing values
significance.level 1 - alpha
computation.time Elapsed computation time
medians Componentwise medians
mads Componentwise mads
center Location estimate
scatter Covariance estimate
robust.regression Input parameter
md.type Input parameter
cutpoint The default threshold MD-value for the cut-off of outliers

The further components returned by TRC are:

outind Indicator of outliers
dist Mahalanobis distances (with missing values)
weighted.quantile

Author(s)
Beat Hulliger

References

Examples
```
data(bushfiremL, bushfire.weights)
det.res <- TRC(bushfiremL, weights = bushfire.weights)
PlotMD(det.res$dist, ncol(bushfiremL))
print(det.res)
```

weighted.quantile  Quantiles of a weighted cdf

Description
A weighted cdf is calculated and quantiles are evaluated. Missing values are discarded.

Usage
weighted.quantile(x, w, prob = 0.5, plot = FALSE)

Arguments
- x  a vector of data.
- w  a vector of (sampling) weights.
- prob  the probability for the quantile.
- plot  if TRUE, the weighted cdf is plotted.

Details
Weighted linear interpolation in case of non-unique inverse. Gives a warning when the contribution of the weight of the smallest observation to the total weight is larger than prob.

Value
The quantile according to prob (by default it returns the weighted median).

Note
No variance calculation.
Weighted univariate variance coping with missing values

Description
This function is analogous to weighted.mean.

Usage
weighted.var(x, w, na.rm = FALSE)

Arguments
- x: a vector of data.
- w: a vector of positive weights (may not have missings where x is observed).
- na.rm: if TRUE remove missing values.

Details
The weights are standardised such that \( \sum_{observed} w_i \) equals the number of observed values in \( x \).
The function calculates
\[
\sum_{observed} w_i (x_i - \text{weighted.mean}(x, w, na.rm = TRUE))^2 / ((\sum_{observed} w_i) - 1)
\]

Value
The weighted variance of \( x \) with weights \( w \) (with missing values removed when \( na.rm = \) TRUE).

Author(s)
Beat Hulliger
### Description

Winsorization of outliers according to the Mahalanobis distance followed by an imputation under the multivariate normal model. Only the outliers are winsorized. The Mahalanobis distance MDmiss allows for missing values.

### Usage

```r
winsimp(data, center, scatter, outind, seed = 1000003)
```

### Arguments

- **data**: a data frame with the data.
- **center**: (robust) estimate of the center (location) of the observations.
- **scatter**: (robust) estimate of the scatter (covariance-matrix) of the observations.
- **outind**: logical vector indicating outliers with 1 or TRUE for outliers.
- **seed**: seed for random number generator.

### Details

It is assumed that center, scatter and outind stem from a multivariate outlier detection algorithm which produces robust estimates and which declares outliers observations with a large Mahalanobis distance. The cutpoint is calculated as the least (unsquared) Mahalanobis distance among the outliers. The winsorization reduces the weight of the outliers:

\[
\tilde{y}_i = \mu_R + (y_i - \mu_R) \cdot c/d_i
\]

where \(\mu_R\) is the robust center and \(d_i\) is the (unsquared) Mahalanobis distance of observation \(i\).
Value

`winsimp` returns a list whose first component `output` is a sublist with the following components:

- `cutpoint` Cutpoint for outliers
- `proc.time` Processing time
- `n.missing.before` Number of missing values before imputation
- `n.missing.after` Number of missing values after imputation

The further component returned by `winsimp` is:

- `imputed.data` Imputed data set

Author(s)

Beat Hulliger

References


See Also

`MDmiss`. Uses `imp.norm`.

Examples

```r
data(bushfirem, bushfire.weights)
det.res <- TRC(bushfirem, weight = bushfire.weights)
imp.res <- winsimp(bushfirem, det.res$output$center, det.res$output$scatter, det.res$outind)
print(imp.res$output)
```
Index

*Topic datasets
  bushfire, 4
  bushfire.weights, 5
  bushfirem, 5
  sepe, 18

BEM, 2, 12
bushfire, 4
bushfire.weights, 5
bushfirem, 5
cov.rob, 13
EAdet, 6, 10
EAimp, 8, 9
ER, 11
GIMCD, 12
imp.norm, 24
mahalanobis, 14
MDmiss, 13, 24
modi, 14
modi-package (modi), 14
PlotMD, 15
POEM, 16
sepe, 18
TRC, 19
weighted.quantile, 21
weighted.var, 22
Winsimp, 23