Package ‘cellWise’

April 14, 2020

Type Package

Version 2.1.1

Date 2020-04-13

Title Analyzing Data with Cellwise Outliers

Depends R (>= 3.2.0)

Suggests knitr, robustHD, MASS, ellipse

Imports reshape2, scales, ggplot2, matrixStats, gridExtra, robustbase, 
   rrcov, svd, Rcpp (>= 0.12.10.14)

LinkingTo Rcpp, RcppArmadillo (>= 0.7.600.1.0)

Description Tools for detecting cellwise outliers and robust methods to analyze data which may con-
   tain them. Contains the implementation of the algorithms de-
   bert et al. (2019) <doi:10.1080/00401706.2018.1562989>, Raymaek-

License GPL (>= 2)

LazyLoad yes

Author Jakob Raymaekers [aut, cre],
   Peter Rousseeuw [aut],
   Wannes Van den Bossche [aut],
   Mia Hubert [aut]

Maintainer Jakob Raymaekers <jakob.raymaekers@kuleuven.be>

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-04-14 10:40:03 UTC
cellMap

**cellMap**

**Description**

This function draws a cellmap, possibly of a subset of rows and columns of the data, and possibly combining cells into blocks. A cellmap shows which cells are missing and which ones are outlying, marking them in red for unusually large cell values and in blue for unusually low cell values. When cells are combined into blocks, the final color is the average of the colors in the individual cells.

**Usage**

```r
cellMap(D, R, indcells = NULL, indrows = NULL,
        standOD=NULL,showVals=NULL,rowlabels="",
        columnlabels="",mTitle="",rowtitle="",
        columntitle="",showrows=NULL, showcolumns=NULL,
        nrowsinblock=1,ncolumnsinblock=1,autolabel=TRUE,
        columnangle=90,sizetitles=1.1,adjustrowlabels=1,
        adjustcolumnlabels=1, colContrast=1,outlyingGrad=TRUE,
        darkestColor = sqrt(qchisq(0.999,1)))
```

**Arguments**

- `D` : The data matrix (required input argument).
- `R` : Matrix of standardized residuals of the cells (required input argument)
- `indcells` : Indices of outlying cells. Defaults to NULL, which indicates the cells for which $|R| > \sqrt{(qchisq(0.999,1))}$. 

cellMap

indrows  Indices of outlying rows. By default no rows are indicated.
standOD  Standardized Orthogonal Distance of each row. Defaults to NULL, then no rows are indicated.
showVals  Takes the values "D", "R" or NULL and determines whether or not to show the entries of the data matrix (D) or the residuals (R) in the cellmap. Defaults to NULL, then no values are shown.
rowlabels  Labels of the rows.
columnlabels  Labels of the columns.
mTitle  Main title of the cellMap.
rowtitle  Title for the rows.
columntitle  Title for the columns.
showrows  Indices of the rows to be shown. Defaults to NULL which means all rows are shown.
showcolumns  Indices of the columns to be shown. Defaults to NULL which means all columns are shown.
nrowsinblock  How many rows are combined in a block. Defaults to 1.
ncolumnsinblock  How many columns are combined in a block. Defaults to 1.
autolabel  Automatically combines labels of cells in blocks. If FALSE, you must provide the final columnlabels and/or rowlabels. Defaults to TRUE.
columnangle  Angle of the column labels. Defaults to 90.
sizetitles  Size of row title and column title. Defaults to 1.1.
adjustrowlabels  Adjust row labels: 0=left, 0.5=centered, 1=right. Defaults to 1.
adjustcolumnlabels  Adjust column labels: 0=left, 0.5=centered, 1=right. Defaults to 1.
colContrast  Parameter regulating the contrast of colors, should be in [1, 5]. Defaults to 1.
outlyingGrad  If TRUE, the color is gradually adjusted in function of the outlyingness. Defaults to TRUE.
darkestColor  Standardized residuals bigger than this will get the darkest color.

Author(s)

Rousseeuw P.J., Van den Bossche W.

References


See Also

DDC
checkDataSet

Examples

# For examples of the cellmap, we refer to the vignette:
vignette("DDC_examples")

Description

This function checks the dataset X, and sets aside certain columns and rows that do not satisfy the conditions. It is used by the DDC and MacroPCA functions but can be used by itself, to clean a dataset for a different type of analysis.

Usage

checkDataSet(X, fracNA = 0.5, numDiscrete = 3, precScale = 1e-12, silent = FALSE, cleanNAfirst = "automatic")

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X is the input data, and must be an n by d matrix or data frame.</td>
</tr>
<tr>
<td>fracNA</td>
<td>Only retain columns and rows with fewer NAs than this fraction. Defaults to 0.5.</td>
</tr>
<tr>
<td>numDiscrete</td>
<td>A column that takes on numDiscrete or fewer values will be considered discrete and not retained in the cleaned data. Defaults to 3.</td>
</tr>
<tr>
<td>precScale</td>
<td>Only consider columns whose scale is larger than precScale. Here scale is measured by the median absolute deviation. Defaults to (1e^{-12}).</td>
</tr>
<tr>
<td>silent</td>
<td>Whether or not the function progress messages should be printed. Defaults to FALSE.</td>
</tr>
<tr>
<td>cleanNAfirst</td>
<td>If &quot;columns&quot;, first columns then rows are checked for NAs. If &quot;rows&quot;, first rows then columns are checked for NAs. &quot;automatic” checks columns first if (d \geq 5n) and rows first otherwise. Defaults to &quot;automatic&quot;.</td>
</tr>
</tbody>
</table>

Value

A list with components:

- **colInAnalysis**
  Column indices of the columns used in the analysis.
- **rowInAnalysis**
  Row indices of the rows used in the analysis.
- **namesNotNumeric**
  Names of the variables which are not numeric.
• **namesCaseNumber**
  The name of the variable(s) which contained the case numbers and was therefore removed.

• **namesNACol**
  Names of the columns left out due to too many NA’s.

• **namesNARow**
  Names of the rows left out due to too many NA’s.

• **namesDiscrete**
  Names of the discrete variables.

• **namesZeroScale**
  Names of the variables with zero scale.

• **remX**
  Remaining (cleaned) data after checkDataSet.

**Author(s)**
Rousseeuw P.J., Van den Bossche W.

**References**

**See Also**

- DDC

**Examples**

```r
library(MASS)
set.seed(12345)
n <- 100; d = 10
A <- matrix(0.9, d, d); diag(A) = 1
x <- mvrnorm(n, rep(0, d), A)
x[sample(1:(n * d), 100, FALSE)] <- NA
x <- cbind(1:n, x)
checkedx <- checkDataSet(x)
```

# For more examples, we refer to the vignette:
vignette("DDC_examples")

---

**Description**

This function aims to detect cellwise outliers in the data. These are entries in the data matrix which are substantially higher or lower than what could be expected based on the other cells in its column as well as the other cells in its row, taking the relations between the columns into account. Note that this function first calls `checkDataSet` and analyzes the remaining cleaned data.
Usage

\[ \text{DDC}(X, \text{DDCpars} = \text{list()}) \]

Arguments

- **X**: X is the input data, and must be an \( n \) by \( d \) matrix or a data frame.

- **DDCpars**: A list of available options:
  - **fracNA**: Only consider columns and rows with fewer NAs (missing values) than this fraction (percentage). Defaults to 0.5.
  - **numDiscrete**: A column that takes on numDiscrete or fewer values will be considered discrete and not used in the analysis. Defaults to 3.
  - **precScale**: Only consider columns whose scale is larger than precScale. Here scale is measured by the median absolute deviation. Defaults to \( 1 \times 10^{-12} \).
  - **cleanNAfirst**: If "columns", first columns then rows are checked for NAs. If "rows", first rows then columns are checked for NAs. "automatic" checks columns first if \( d \geq 5n \) and rows first otherwise. Defaults to "automatic".
  - **tolProb**: Tolerance probability, with default 0.99, which determines the cutoff values for flagging outliers in several steps of the algorithm.
  - **corrlim**: When trying to estimate \( z_{ij} \) from other variables \( h \), we will only use variables \( h \) with \( |\rho_{j,h}| \geq \text{corrlim} \). Variables \( j \) without any correlated variables \( h \) satisfying this are considered standalone, and treated on their own. Defaults to 0.5.
  - **combinRule**: The operation to combine estimates of \( z_{ij} \) coming from other variables \( h \): can be "mean", "median", "wmean" (weighted mean) or "wmedian" (weighted median). Defaults to "wmean".
  - **returnBigXimp**: If TRUE, the imputed data matrix \( X_{\text{imp}} \) in the output will include the rows and columns that were not part of the analysis (and can still contain NAs). Defaults to FALSE.
  - **silent**: If TRUE, statements tracking the algorithm’s progress will not be printed. Defaults to FALSE.
  - **nLocScale**: When estimating location or scale from more than nLocScale data values, the computation is based on a random sample of size nLocScale to save time. When nLocScale = 0 all values are used. Defaults to 25000.
  - **fastDDC**: Whether to use the fastDDC option or not. The fastDDC algorithm uses approximations to allow to deal with high dimensions. Defaults to TRUE for \( d > 750 \) and FALSE otherwise.
• standType
   The location and scale estimators used for robust standardization. Should be one of "1stepM", "mcd" or "wrap". See estLocScale for more info. Only used when fastDDC = FALSE. Defaults to "1stepM".

• corrType
   The correlation estimator used to find the neighboring variables. Must be one of "wrap" (wrapping correlation), "rank" (Spearman correlation) or "gkwls" (Gnanadesikan-Kettenring correlation followed by weighting). Only used when fastDDC = FALSE. Defaults to "gkwls".

• transFun
   The transformation function used to compute the robust correlations when fastDDC = TRUE. Can be "wrap" or "rank". Defaults to "wrap".

• nbngbrs
   When fastDDC = TRUE, each column is predicted from at most nbngbrs columns correlated to it. Defaults to 100.

Value

A list with components:

• DDCpars
   The list of options used.

• colInAnalysis
   The column indices of the columns used in the analysis.

• rowInAnalysis
   The row indices of the rows used in the analysis.

• namesNotNumeric
   The names of the variables which are not numeric.

• namesCaseNumber
   The name of the variable(s) which contained the case numbers and was therefore removed.

• namesNAcol
   Names of the columns left out due to too many NA’s.

• namesNArow
   Names of the rows left out due to too many NA’s.

• namesDiscrete
   Names of the discrete variables.

• namesZeroScale
   Names of the variables with zero scale.

• remX
   Cleaned data after checkDataSet.

• locX
   Estimated location of X.

• scaleX
   Estimated scales of X.
- $Z$
  Standardized $\text{remX}$.
- $\text{nbngbrs}$
  Number of neighbors used in estimation.
- $\text{ngbrs}$
  Indicates neighbors of each column, i.e. the columns most correlated with it.
- $\text{robcors}$
  Robust correlations.
- $\text{robslopes}$
  Robust slopes.
- $\text{deshrinkage}$
  The deshrinkage factor used for every connected (i.e. non-standalone) column of $X$.
- $X_{\text{est}}$
  Predicted $X$.
- $\text{scalestres}$
  Scale estimate of the residuals $X - X_{\text{est}}$.
- $\text{stdResid}$
  Residuals of orginal $X$ minus the estimated $X_{\text{est}}$, standardized by column.
- $\text{indcells}$
  Indices of the cells which were flagged in the analysis.
- $T_i$
  Outlyingness (test) value of each row.
- $\text{medTi}$
  Median of the $T_i$ values.
- $\text{madTi}$
  Mad of the $T_i$ values.
- $\text{indrows}$
  Indices of the rows which were flagged in the analysis.
- $\text{indNAs}$
  Indices of all NA cells.
- $\text{indall}$
  Indices of all cells which were flagged in the analysis plus all cells in flagged rows plus the indices of the NA cells.
- $X_{\text{imp}}$
  Imputed $X$.

**Author(s)**

Raymaekers J., Rousseeuw P.J., Van den Bossche W.

**References**


See Also

checkDataSet, cellMap

Examples

library(MASS); set.seed(12345)
n <- 50; d <- 20
A <- matrix(0.9, d, d); diag(A) = 1
x <- mvrnorm(n, rep(0, d), A)
x[sample(1:(n * d), 50, FALSE)] <- NA
x[sample(1:(n * d), 50, FALSE)] <- 10
x[sample(1:(n * d), 50, FALSE)] <- -10
x <- cbind(1:n, x)
DDCx <- DDC(x)
cellMap(DDCx$remX, DDCx$stdResid,
columnlabels = 1:d, rowlabels = 1:n)

# For more examples, we refer to the vignette:
vignette("DDC_examples")

DDCpredict

Description

Based on a DDC fit on an initial (training) data set $X$, this function analyzes a new (test) data set $X_{\text{new}}$.

Usage

DDCpredict($X_{\text{new}}$, InitialDDC, DDCpars = NULL)

Arguments

$X_{\text{new}}$ The new data (test data), which must be a matrix or a data frame. It must always be provided.

InitialDDC The output of the DDC function on the initial (training) dataset. Must be provided.

DDCpars The input options to be used for the prediction. By default the options of InitialDDC are used.

Value

A list with components:

DDCpars the options used in the call, see DDC.
locX the locations of the columns, from InitialDDC.
scaleX the scales of the columns, from InitialDDC.
Z Xnew standardized by locX and scaleX.
nbngbrs predictions use a combination of Nbngbrs columns.
gbbrs for each column, the list of its neighbors, from InitialDDC.
robcors for each column, the correlations with its neighbors, from InitialDDC.
robslopes slopes to predict each column by its neighbors, from InitialDDC.
deshrinkage for each connected column, its deshrinkage factor used in InitialDDC.
Xest predicted values for every cell of Xnew.
scalestres scale estimate of the residuals (Xnew - Xest), from InitialDDC.
stdResid columnwise standardized residuals of Xnew.
indcells positions of cellwise outliers in Xnew.
Ti outlyingness of rows in Xnew.
medTi median of the Ti in InitialDDC.
madTi mad of the Ti in InitialDDC.
indrows row numbers of the outlying rows in Xnew.
indNAs positions of the NA's in Xnew.
indall positions of NA's and outlying cells in Xnew.
Ximp Xnew where all cells in indall are imputed by their prediction.

Author(s)
Rousseeuw P.J., Van den Bossche W.

References

See Also
checkDataSet, cellMap, DDC

Examples
library(MASS) set.seed(12345)
n <- 100; d <- 10 A <- matrix(0.9, d, d); diag(A) = 1 x <- mvrnorm(n, rep(0,d), A) x[sample(1:(n * d), 50, FALSE)] <- NA x[sample(1:(n * d), 50, FALSE)] <- 10 x <- cbind(1:n, x) DDCx <- DDC(x) xnew <- mvrnorm(50, rep(0,d), A) xnew[sample(1:(50 * d), 50, FALSE)] <- 10
predict.out <- DDCpredict(xnew, DDCx)
cellMap(xnew, predict.out$stdResid,
columnlabels = 1:d, rowlabels = 1:50)

# For more examples, we refer to the vignette:  
vignette("DDC_examples")

---

### dog_walker  
**Dog walker dataset**

**Description**

A dataset containing the image sequence of a video. The sequence consists of 54 frames of 144 by 180 pixels in Red/Geen/Blue (RGB) format.

**Usage**

data("dog_walker")

**Format**

An array of dimensions $54 \times 144 \times 180 \times 3$.

**Source**

[http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html](http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html)

**Examples**

data(dog_walker)  
# For more examples, we refer to the vignette:  
vignette("Wrap_examples")

---

### dposs  
**DPOSS dataset**

**Description**

This is a random subset of 20'000 stars from the Digitized Palomar Sky Survey (DPOSS) described by Odewahn et al. (1998).

**Usage**

data("dposs")

**Format**

A matrix of dimensions $20000 \times 21$. 
References


Examples

data(dposs)

# For more examples, we refer to the vignette:
vignette("MacroPCA_examples")

---

estLocScale | *Estimate robust location and scale*

Description

Estimate a robust location estimate and scale estimate of every column in X.

Usage

estLocScale(X, type = "wrap", precScale = 1e-12,
center = TRUE, alpha = 0.5, nLocScale = 25000, silent = FALSE)

Arguments

- **X** The input data. It must be an n by d matrix or a data frame.
- **type** The type of estimators used. One of:
  - "1stepM": The location is the 1-step M-estimator with the biweight psi function. The scale estimator is the 1-step M-estimator using a Huber rho function with $b = 2.5$.
  - "mcd": the location is the weighted univariate MCD estimator with cutoff $\sqrt{qchisq(0.975, 1)}$. The scale is the corresponding weighted univariate MCD estimator, with a correction factor to make it approximately unbiased at gaussian data.
  - "wrap": Starting from the initial estimates corresponding to option "mcd", the location is the 1-step M-estimator with the wrapping psi function with $b = 1.5$ and $c = 4$. The scale estimator is the same as in option "mcd".
- **precScale** The precision scale used throughout the algorithm. Defaults to $1e-12$.
- **center** Whether or not the data has to be centered before calculating the scale. Not in use for type = "mcd". Defaults to TRUE.
- **alpha** The value of $\alpha$ in the univariate mcd, must be between 0.5 and 1. The subsize is $h = \lceil \alpha n \rceil$. Only used for type = "mcd". Defaults to $\alpha = 0.5$. Defaults to FALSE.
If nLocScale < n, nLocScale observations are sampled to compute the location and scale. This speeds up the computation if n is very large. When nLocScale = 0 all observations are used. Defaults to nLocScale = 25000.

Whether or not a warning message should be printed when very small scales are found. Defaults to FALSE.

A list with components:

- loc
  A vector with the estimated locations.
- scale
  A vector with the estimated scales.

Raymaekers, J. and Rousseeuw P.J.


Examples

```r
library(MASS)
set.seed(12345)
n = 100; d = 10
X = mvrnorm(n, rep(0, 10), diag(10))
locScale = estLocScale(X)
```

The glass dataset

A dataset containing spectra with $d = 750$ wavelengths collected on $n = 180$ archeological glass samples.

data("glass")
Format

A data frame with 180 observations of 750 wavelengths.

Source


Examples

data(glass)

---

### ICPCA

*Iterative Classical PCA*

**Description**

This function carries out classical PCA when the data may contain missing values, by an iterative algorithm. It is based on a Matlab function from the Missing Data Imputation Toolbox v1.0 by A. Folch-Fortuny, F. Arteaga and A. Ferrer.

**Usage**

```r
ICPCA(X, k, scale = FALSE, maxiter = 20, tol = 0.005,
       tolProb = 0.99, distprob = 0.99)
```

**Arguments**

- **X**
  
  the input data, which must be a matrix or a data frame. It may contain NA’s. It must always be provided.

- **k**
  
  the desired number of principal components

- **scale**
  
  a value indicating whether and how the original variables should be scaled. If `scale=FALSE` (default) or `scale=NULL` no scaling is performed (and a vector of 1s is returned in the `$scaleX` slot). If `scale=TRUE` the variables are scaled to have a standard deviation of 1. Alternatively scale can be a function like mad, or a vector of length equal to the number of columns of x. The resulting scale estimates are returned in the `$scaleX` slot of the output.

- **maxiter**
  
  maximum number of iterations. Default is 20.

- **tol**
  
  tolerance for iterations. Default is 0.005.

- **tolProb**
  
  tolerance probability for residuals. Defaults to 0.99.

- **distprob**
  
  probability determining the cutoff values for orthogonal and score distances. Default is 0.99.
Value

A list with components:

- **scaleX**: the scales of the columns of X.
- **k**: the number of principal components.
- **loadings**: the columns are the k loading vectors.
- **eigenvalues**: the k eigenvalues.
- **center**: vector with the fitted center.
- **covmatrix**: estimated covariance matrix.
- **It**: number of iteration steps.
- **diff**: convergence criterion.
- **X.NAimp**: data with all NA's imputed.
- **scores**: scores of X.NAimp.
- **OD**: orthogonal distances of the rows of X.NAimp.
- **cutoffOD**: cutoff value for the OD.
- **SD**: score distances of the rows of X.NAimp.
- **cutoffSD**: cutoff value for the SD.
- **indrows**: row numbers of rowwise outliers.
- **residScale**: scale of the residuals.
- **stdResid**: standardized residuals. Note that these are NA for all missing values of X.
- **indcells**: indices of cellwise outliers.

Author(s)

Wannes Van Den Bossche

References


Examples

```r
library(MASS)
set.seed(12345)
n <- 100; d <- 10
A <- diag(d) * 0.1 + 0.9
x <- mvrnorm(n, rep(0, d), A)
x[sample(1:(n * d), 100, FALSE)] <- NA
ICPCA.out <- ICPCA(x, k = 2)
plot(ICPCA.out$scores)
```
MacroPCA

Description

This function performs the MacroPCA algorithm, which can deal with Missing values and Cellwise and Rowwise Outliers. Note that this function first calls checkDataSet and analyzes the remaining cleaned data.

Usage

MacroPCA(X, k = 0, MacroPCApars = NULL)

Arguments

X X is the input data, and must be an \( n \) by \( d \) matrix or a data frame.

k k is the desired number of principal components. If \( k = 0 \) or \( k = \text{NULL} \), the algorithm will compute the percentage of explained variability for \( k \) upto \( k_{\text{max}} \) and show a scree plot, and suggest to choose a value of \( k \) such that the cumulative percentage of explained variability is at least 80%.

MacroPCApars A list of available options detailed below. If MacroPCApars = NULL the defaults below are used.

- DDCpars
  A list with parameters for the first step of the MacroPCA algorithm (for the complete list see the function DDC). Default is NULL.

- kmax
  The maximal number of principal components to compute. Default is \( k_{\text{max}} = 10 \). If \( k \) is provided \( k_{\text{max}} \) does not need to be specified, unless \( k \) is larger than 10 in which case you need to set \( k_{\text{max}} \) high enough.

- alpha
  This is the coverage, i.e. the fraction of rows the algorithm should give full weight. Alpha should be between 0.50 and 1, the default is 0.50.

- scale
  A value indicating whether and how the original variables should be scaled. If scale = FALSE (default) or scale = NULL no scaling is performed (and a vector of 1s is returned in the \$scaleX slot). If scale = TRUE the data are scaled by a 1-step M-estimator of scale with the Tukey biweight weight function to have a robust scale of 1. Alternatively scale can be a vector of length equal to the number of columns of \( x \). The resulting scale estimates are returned in the \$scaleX slot of the MacroPCA output.

- maxdir
  The maximal number of random directions to use for computing the outlyingness of the data points. Default is maxdir = 250. If the number \( n \) of observations is small all \( n \times (n - 1)/2 \) pairs of observations are used.
• distprob
  The quantile determining the cutoff values for orthogonal and score dis-
  tances. Default is 0.99.
• silent
  If TRUE, statements tracking the algorithm’s progress will not be printed.
  Defaults to FALSE.
• maxiter
  Maximum number of iterations. Default is 20.
• tol
  Tolerance for iterations. Default is 0.005.
• bigOutput
  whether to compute and return NAimp, Cellimp and Fullimp. Defaults to
  TRUE.

Value

A list with components:

MacroPCApars the options used in the call.
remX     Cleaned data after checkDataSet.
DDC      results of the first step of MacroPCA. These are needed to run MacroPCApredict
         on new data.
scaleX   the scales of the columns of X.
k        the number of principal components.
loadings the columns are the k loading vectors.
eigenvalues the k eigenvalues.
center   vector with the fitted center.
alpha    alpha from the input.
h        h (computed from alpha).
It       number of iteration steps.
diff     convergence criterion.
X.NAimp  data with all NA’s imputed by MacroPCA.
scores   scores of X.NAimp.
OD       orthogonal distances of the rows of X.NAimp.
cutoffOD cutoff value for the OD.
SD       score distances of the rows of X.NAimp.
cutoffSD cutoff value for the SD.
indrows  row numbers of rowwise outliers.
residScale scale of the residuals.
stdResid standardized residuals. Note that these are NA for all missing values of X.
indcells indices of cellwise outliers.
NAimp various results for the NA-imputed data.
Cellimp various results for the cell-imputed data.
Fullimp various result for the fully imputed data.
Author(s)

Rousseeuw P.J., Van den Bossche W.

References


See Also

checkDataSet, cellMap, DDC

Examples

library(MASS)
set.seed(12345)
n <- 50; d <- 10
A <- matrix(0.9, d, d); diag(A) = 1
x <- mvrnorm(n, rep(0, d), A)
x[sample(1:(n * d), 50, FALSE)] <- NA
x[sample(1:(n * d), 50, FALSE)] <- 10
x <- cbind(1:n, x)
MacroPCA.out <- MacroPCA(x, 2)
cellMap(MacroPCA.out$remX, MacroPCA.out$stdResid,
columnlabels = 1:d, rowlabels = 1:n)

Description

Based on a MacroPCA fit of an initial (training) data set X, this function analyzes a new (test) data set Xnew.

Usage

MacroPCApredict(Xnew, InitialMacroPCA, MacroPCApars = NULL)

Arguments

Xnew The new data (test data), which must be a matrix or a data frame. It must always be provided.

InitialMacroPCA The output of the MacroPCA function on the initial (training) dataset. Must be provided.

MacroPCApars The input options to be used for the prediction. By default the options of Initial-MacroPCA are used. For the complete list of options see the function MacroPCA.
Value

A list with components:

- `MacroPCApars` the options used in the call.
- `scaleX` the scales of the columns of X.
- `k` the number of principal components.
- `loadings` the columns are the k loading vectors.
- `eigenvalues` the k eigenvalues.
- `center` vector with the fitted center.
- `It` number of iteration steps.
- `diff` convergence criterion.
- `X.NAimp` `Xnew` with all NA’s imputed by MacroPCA.
- `scores` scores of `X.NAimp`.
- `OD` orthogonal distances of the rows of `X.NAimp`.
- `cutoffOD` cutoff value for the OD.
- `SD` score distances of the rows of `X.NAimp`.
- `cutoffSD` cutoff value for the SD.
- `indrows` row numbers of rowwise outliers.
- `residScale` scale of the residuals.
- `stdResid` standardized residuals. Note that these are NA for all missing values of `Xnew`.
- `indcells` indices of cellwise outliers.
- `NAimp` various results for the NA-imputed data.
- `Cellimp` various results for the cell-imputed data.
- `Fullimp` various result for the fully imputed data.
- `DDC` result of DDCpredict which is the first step of MacroPCApredict. See the function `DDCpredict`.

Author(s)

Rousseeuw P.J., Van den Bossche W.

References


See Also

`checkDataSet, cellMap, DDC, DDCpredict, MacroPCA`
Examples

```r
library(MASS)
set.seed(12345)
n <- 50; d <- 10
A <- matrix(0.9, d, d); diag(A) = 1
x <- mvrnorm(n, rep(0,d), A)
x[sample(1:(n * d), 50, FALSE)] <- NA
x[sample(1:(n * d), 50, FALSE)] <- 10
x <- cbind(1:n, x)
MacroPCA.out <- MacroPCA(x, 2)
xnew <- mvrnorm(n, rep(0,d), A)
xnew[sample(1:(n * d), 50, FALSE)] <- 10
predict.out <- MacroPCApredict(xnew, MacroPCA.out)
cellMap(xnew, predict.out$stdResid,
columnlabels = 1:d, rowlabels = 1:n)
```

---

**mortality**

*The mortality dataset*

**Description**

This dataset contains the mortality by age for males in France, from 1816 to 2013 as obtained from the Human Mortality Database.

**Usage**

```r
data("mortality")
```

**Format**

A data frame with 198 calendar years (rows) and 91 age brackets (columns).

**Source**

Human Mortality Database. University of California, Berkeley (USA), and Max Planck Institute for Demographic Research (Germany). Available at [https://www.mortality.org](https://www.mortality.org) (data downloaded in November 2015).

**References**


**Examples**

```r
data(mortality)
```
outlierMap  

Plot the outlier map.

Description
The outlier map is a diagnostic plot for the output of MacroPCA.

Usage
outlierMap(res, title="Robust PCA", col="black", pch=16, labelOut=TRUE, id=3)

Arguments
res A list containing the orthogonal distances (OD), the score distances (SD) and their respective cut-offs (cutoffOD and cutoffSD). Can be the output of MacroPCA, robpca, rospca.
title Title of the plot, default is "Robust PCA".
col Colour of the points in the plot, this can be a single colour for all points or a vector specifying the colour for each point. The default is "black".
pch Plotting characters or symbol used in the plot, see points for more details. The default is 16 which corresponds to filled circles.
labelOut Logical indicating if outliers should be labelled on the plot, default is TRUE.
id Number of OD outliers and number of SD outliers to label on the plot, default is 3.

Details
The outlier map contains the score distances on the x-axis and the orthogonal distances on the y-axis. To detect outliers, cut-offs for both distances are shown, see Hubert et al. (2005).

Author(s)
P.J. Rousseeuw

References

See Also
MacroPCA

Examples
# empty for now
The `philips` dataset

A dataset containing measurements of $d = 9$ characteristics of $n = 677$ diaphragm parts, used in the production of TV sets.

Usage

```r
data("philips")
```

Format

A matrix with 677 rows and 9 columns.

Source

The data were provided in 1997 by Gertjan Otten and permission to analyze them was given by Herman Veraa and Frans Van Dommelen at Philips Mecoma in The Netherlands.

References


Examples

```r
data(philips)
```

**truncPC**

Classical Principal Components by truncated SVD.

Description

Similar usage to `classPC` of robustbase except for the new argument `ncomb` which is the desired number of components. Only this many PC's are computed in order to save computation time. Makes use of `propack.svd` of package `svd`.

Usage

```r
truncPC(X, ncomp = NULL, scale = FALSE, center = TRUE,
     signflip = TRUE, via.svd = NULL, scores = FALSE)
```
Arguments

X    a numeric matrix.
ncomp the desired number of components (if not specified, all components are computed).
scale logical, or numeric vector for scaling the columns.
center logical or numeric vector for centering the matrix.
signflip logical indicating if the signs of the loadings should be flipped such that the absolutely largest value is always positive.
via.svd dummy argument for compatibility with classPC calls, will be ignored.
scores logical indicating whether or not scores should be returned.

Value

A list with components:

rank    the (numerical) matrix rank of X, i.e. an integer number between 0 and min(dim(x)).
eigenvalues the k eigenvalues, proportional to the variances, where k is the rank above.
loadings the loadings, a d × k matrix.
scores if the scores argument was TRUE, the n × k matrix of scores.
center a vector of means, unless the center argument was FALSE.
scale a vector of column scales, unless the scale argument was false.

Author(s)

P.J. Rousseeuw

See Also

classPC

Examples

library(MASS)
set.seed(12345)
n <- 100; d <- 10
A <- diag(d) * 0.1 + 0.9
x <- mvrnorm(n, rep(0,d), A)
truncPCA.out <- truncPC(x, ncomp = 2, scores = TRUE)
plot(truncPCA.out$scores)
Wrap the data.

Description

Transforms multivariate data X using the wrapping function with $b = 1.5$ and $c = 4$ and the location and scale given in locX and scaleX.

Usage

```r
wrap(X, locX, scaleX, precScale = 1e-12)
```

Arguments

- **X**
  the input data. It must be an $n$ by $d$ matrix or a data frame.
- **locX**
  The location estimates of the columns of the input data X. Must be a vector of length $d$.
- **scaleX**
  The scale estimates of the columns of the input data X. Must be a vector of length $d$.
- **precScale**
  The precision scale used throughout the algorithm. Defaults to $1e - 12$

Value

A list with components:

- **Xw**
  The wrapped data.
- **colInWrap**
  The column numbers for which the scale estimate was larger than precScale. Those with scale estimate $\leq$ precScale do not occur in Xw to avoid division by (near) zero.

Author(s)

Raymaekers, J. and Rousseeuw P.J.

References

Raymaekers, J., Rousseeuw P.J. (2019). Fast robust correlation for high dimensional data. Techno-
metrics, published online.

See Also

`estLocScale`
Examples

```r
library(MASS)
set.seed(12345)
n <- 100; d <- 10
X <- mvrnorm(n, rep(0, 10), diag(10))
locScale <- estLocScale(X)
Xw <- wrap(X, locScale$loc, locScale$scale)$Xw
```
Index

cellMap, 2, 9, 10, 18, 19
checkDataSet, 4, 5, 9, 10, 16, 18, 19
classPC, 22, 23

DDC, 3-5, 9, 10, 16, 18, 19
DDCpredict, 9, 19
dog_walker, 11
dposs, 11

estLocScale, 7, 12, 24
glass, 13
ICPCA, 14

MacroPCA, 4, 16, 18, 19, 21
MacroPCApredict, 18
mortality, 20

outlierMap, 21

philips, 22
robpca, 21
rospca, 21

truncPC, 22
wrap, 13, 24