Package ‘mvoutlier’

October 13, 2022

Version 2.1.1
Date 2021-07-29
Title Multivariate Outlier Detection Based on Robust Methods
Author Peter Filzmoser <P.Filzmoser@tuwien.ac.at> and
    Moritz Gschwandtner <e0125439@student.tuwien.ac.at>
Maintainer P. Filzmoser <P.Filzmoser@tuwien.ac.at>
Depends sgeostat, R (>= 3.1)
Imports robustbase
Description Various methods for multivariate outlier detection: arw, a Mahalanobis-type method with an adaptive outlier cutoff value; locout, a method incorporating local neighborhood; pcout, a method for high-dimensional data; mvoutlier.CoDa, a method for compositional data. References are provided in the corresponding help files.
LazyData TRUE
License GPL (>= 3)
URL http://cstat.tuwien.ac.at/filz/
NeedsCompilation no
Repository CRAN
Date/Publication 2021-07-30 08:10:05 UTC

R topics documented:

aq.plot ........................................ 2
arw ........................................ 3
bhorizon .................................... 5
bss.background ............................... 7
bssbot .................................... 8
bssstop .................................... 10
chisq.plot .................................. 12
chorizon .................................... 13
color.plot .................................. 17
corr.plot .................................. 18
Description

The function `aq.plot` plots the ordered squared robust Mahalanobis distances of the observations against the empirical distribution function of the $MD^2_i$. In addition the distribution function of $\text{chisq}_p$ is plotted as well as two vertical lines corresponding to the chisq-quantile specified in the argument list (default is 0.975) and the so-called adjusted quantile. Three additional graphics are created (the first showing the data, the second showing the outliers detected by the specified quantile of the $\text{chisq}_p$ distribution and the third showing these detected outliers by the adjusted quantile).

Usage

```r
aq.plot(x, delta=qchisq(0.975, df=ncol(x)), quan=1/2, alpha=0.05)
```

Arguments

- **x**: matrix or data.frame containing the data; has to be at least two-dimensional
- **delta**: quantile of the chi-squared distribution with ncol(x) degrees of freedom. This quantile appears as cyan-colored vertical line in the plot.
- **quan**: proportion of observations which are used for mcd estimations; has to be between 0.5 and 1, default is 0.5
- **alpha**: Maximum thresholding proportion (optional scalar, default: alpha = 0.05)
Details

The function `aq.plot` plots the ordered squared robust Mahalanobis distances of the observations against the empirical distribution function of the $MD^2_i$. The distance calculations are based on the MCD estimator.

For outlier detection two different methods are used. The first one marks observations as outliers if they exceed a certain quantile of the chi-squared distribution. The second is an adaptive procedure searching for outliers specifically in the tails of the distribution, beginning at a certain chisq-quantile (see Filzmoser et al., 2005).

The function behaves differently depending on the dimension of the data. If the data is more than two-dimensional the data are projected on the first two robust principal components.

Value

- outliers: boolean vector of outliers

Author(s)

Moritz Gschwandtner <<e@125439@student.tuwien.ac.at>>
Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> [http://cstat.tuwien.ac.at/filz/]

References


Examples

```r
# create data:
set.seed(134)
x <- cbind(rnorm(80), rnorm(80), rnorm(80))
y <- cbind(rnorm(10, 5, 1), rnorm(10, 5, 1), rnorm(10, 5, 1))
z <- rbind(x, y)
# execute:
aq.plot(z, alpha=0.1)
```

Description

Adaptive reweighted estimator for multivariate location and scatter with hard-rejection weights. The multivariate outliers are defined according to the supremum of the difference between the empirical distribution function of the robust Mahalanobis distance and the theoretical distribution function.

Usage

`arw(x, m0, c0, alpha, pcrit)`
Arguments

- **x**  
  Dataset (n x p)
- **m0**  
  Initial location estimator (1 x p)
- **c0**  
  Initial scatter estimator (p x p)
- **alpha**  
  Maximum thresholding proportion (optional scalar, default: alpha = 0.025)
- **pcrit**  
  Critical value obtained by simulations (optional scalar, default value obtained from simulations)

Details

At the basis of initial estimators of location and scatter, the function `arw` performs a reweighting step to adjust the threshold for outlier rejection. The critical value `pcrit` was obtained by simulations using the MCD estimator as initial robust covariance estimator. If a different estimator is used, `pcrit` should be changed and computed by simulations for the specific dimensions of the data `x`.

Value

- **m**  
  Adaptive location estimator (p x 1)
- **c**  
  Adaptive scatter estimator (p x p)
- **cn**  
  Adaptive threshold ("adjusted quantile")
- **w**  
  Weight vector (n x 1)

Author(s)

Moritz Gschwandtner <e0125439@student.tuwien.ac.at>
Peter Filzmoser <P.Filzmoser@tuwien.ac.at> [http://cstat.tuwien.ac.at/filz/](http://cstat.tuwien.ac.at/filz/)

References


Examples

```r
x <- cbind(rnorm(100), rnorm(100))
arw(x, apply(x,2,mean), cov(x))
```
Description

The Kola data were collected in the Kola Project (1993-1998, Geological Surveys of Finland (GTK) and Norway (NGU) and Central Kola Expedition (CKE), Russia). More than 600 samples in five different layers were analysed, this dataset contains the B-horizon.

Usage

data(bhorizon)

Format

A data frame with 609 observations on the following 48 variables.

ID  a numeric vector
XCOO a numeric vector
YCOO a numeric vector
Ag  a numeric vector
Al  a numeric vector
Al_XRF a numeric vector
As  a numeric vector
Ba  a numeric vector
Be  a numeric vector
Bi  a numeric vector
Ca  a numeric vector
Ca_XRF a numeric vector
Cd  a numeric vector
Co  a numeric vector
Cr  a numeric vector
Cu  a numeric vector
EC  a numeric vector
Fe  a numeric vector
Fe_XRF a numeric vector
K   a numeric vector
K_XRF a numeric vector
LOI a numeric vector
La  a numeric vector
Li a numeric vector
Mg a numeric vector
Mg_XRF a numeric vector
Mn a numeric vector
Mn_XRF a numeric vector
Mo a numeric vector
Na a numeric vector
Na_XRF a numeric vector
Ni a numeric vector
P a numeric vector
P_XRF a numeric vector
Pb a numeric vector
S a numeric vector
Sc a numeric vector
Se a numeric vector
Si a numeric vector
Si_XRF a numeric vector
Sr a numeric vector
Te a numeric vector
Th a numeric vector
Ti a numeric vector
Ti_XRF a numeric vector
V a numeric vector
Y a numeric vector
Zn a numeric vector

Source

References

Examples
data(bhorizon)
# classical versus robust correlation
corr.plot(log(bhorizon[,"Al"]), log(bhorizon[,"Na"]))
bss.background  

Background map for the BSS project

Description

Coordinates of the BSS data background map

Usage

data(bss.background)

Format

A data frame with 6093 observations on the following 2 variables.

V1  a numeric vector with the x-coordinates
V2  a numeric vector with the y-coordinates

Details

Is used by pbb()

Source

BSS project

References


Examples

data(bss.background)
pbb()
bssbot

Bottom Layer of the BSS Data

Description

The BSS data were collected in agricultural soils from Northern Europe from an area of about 1,800,000 km². 769 samples on an irregular grid were taken in two different layers, the top layer (0-20cm) and the bottom layer. This dataset contains the bottom layer of the BSS data. It has 46 variables, including x and y coordinates.

Usage

data(bssbot)

Format

A data frame with 768 observations on the following 46 variables.

ID  a numeric vector
CNo  a numeric vector
XCOO  x coordinates: a numeric vector
YCOO  y coordinates: a numeric vector
SiO2_B  a numeric vector
TiO2_B  a numeric vector
Al2O3_B  a numeric vector
Fe2O3_B  a numeric vector
MnO_B  a numeric vector
MgO_B  a numeric vector
CaO_B  a numeric vector
Na2O_B  a numeric vector
K2O_B  a numeric vector
P2O5_B  a numeric vector
SO3_B  a numeric vector
Cl_B  a numeric vector
F_B  a numeric vector
LOI_B  a numeric vector
As_B  a numeric vector
Ba_B  a numeric vector
Bi_B  a numeric vector
Ce_B  a numeric vector
Co_B a numeric vector
Cr_B a numeric vector
Cs_B a numeric vector
Cu_B a numeric vector
Ga_B a numeric vector
Hf_B a numeric vector
La_B a numeric vector
Mo_B a numeric vector
Nb_B a numeric vector
Ni_B a numeric vector
Pb_B a numeric vector
Rb_B a numeric vector
Sb_B a numeric vector
Sc_B a numeric vector
Sn_B a numeric vector
Sr_B a numeric vector
Ta_B a numeric vector
Th_B a numeric vector
U_B a numeric vector
V_B a numeric vector
W_B a numeric vector
Y_B a numeric vector
Zn_B a numeric vector
Zr_B a numeric vector

Source

BSS Project in Northern Europe

References


Examples

data(bssbot)
# classical versus robust correlation
corr.plot(log(bssbot[, "Al2O3_B"]), log(bssbot[, "Na2O_B"]))
bsstop  

Top Layer of the BSS Data

Description

The BSS data were collected in agricultural soils from Northern Europe. From an area of about 1,800,000 km², 769 samples on an irregular grid were taken in two different layers, the top layer (0-20cm) and the bottom layer. This dataset contains the top layer of the BSS data. It has 46 variables, including x and y coordinates.

Usage

data(bsstop)

Format

A data frame with 768 observations on the following 46 variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>CNo</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>XCOO</td>
<td>x coordinates: a numeric vector</td>
</tr>
<tr>
<td>YCOO</td>
<td>y coordinates: a numeric vector</td>
</tr>
<tr>
<td>SiO2_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>TiO2_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Al2O3_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Fe2O3_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>MnO_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>MgO_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>CaO_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Na2O_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>K2O_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>P2O5_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>SO3_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Cl_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>F_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>LOI_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>As_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Ba_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Bi_T</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Ce_T</td>
<td>a numeric vector</td>
</tr>
</tbody>
</table>


bsstop

Co_T  a numeric vector
Cr_T  a numeric vector
Cs_T  a numeric vector
Cu_T  a numeric vector
Ga_T  a numeric vector
Hf_T  a numeric vector
La_T  a numeric vector
Mo_T  a numeric vector
Nb_T  a numeric vector
Ni_T  a numeric vector
Pb_T  a numeric vector
Rb_T  a numeric vector
Sb_T  a numeric vector
Sc_T  a numeric vector
Sn_T  a numeric vector
Sr_T  a numeric vector
Ta_T  a numeric vector
Th_T  a numeric vector
U_T  a numeric vector
V_T  a numeric vector
W_T  a numeric vector
Y_T  a numeric vector
Zn_T  a numeric vector
Zr_T  a numeric vector

Source
BSS Project in Northern Europe

References

Examples

data(bsstop)
# classical versus robust correlation
corr.plot(log(bsstop[, "Al2O3_T"]), log(bsstop[, "Na2O_T"]))
chisq.plot

Chi-Square Plot

Description

The function chisq.plot plots the ordered robust mahalanobis distances of the data against the quantiles of the Chi-squared distribution. By user interaction this plotting is iterated each time leaving out the observation with the greatest distance.

Usage

chisq.plot(x, quan=1/2, ask=TRUE, ...)

Arguments

x
matrix or data.frame containing the data

quan
amount of observations which are used for mcd estimations. has to be between 0.5 and 1, default ist 0.5

ask
logical. specifies whether user interacton is allowed or not. default is TRUE

... additional graphical parameters

Details

The function chisq.plot plots the ordered robust mahalanobis distances of the data against the quantiles of the Chi-squared distribution. If the data is normal distributed these values should approximately correspond to each other, so outliers can be detected visually. By user interaction this procedure is repeated, each time leaving out the observation with the greatest distance (the number of the observation is printed on the console). This method can be seen as an iterative deletion of outliers until a straight line appears.

Value

outliers
indices of the outliers that are removed by left-click on the plotting device.

Author(s)

Moritz Gschwandtner <<e0125439@student.tuwien.ac.at>>
Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> http://cstat.tuwien.ac.at/filz/

References

Examples

```r
data(humus)
res <- chisq.plot(log(humus[,c("Co","Cu","Ni")]))
res$outliers # these are the potential outliers
```

---

**chorizon**

*C-horizon of the Kola Data*

---

**Description**

The Kola Data were collected in the Kola Project (1993-1998, Geological Surveys of Finland (GTK) and Norway (NGU) and Central Kola Expedition (CKE), Russia). More than 600 samples in five different layers were analysed, this dataset contains the C-horizon.

**Usage**

```r
data(chorizon)
```

**Format**

A data frame with 606 observations on the following 110 variables.

- **ID** a numeric vector
- **XCOO** a numeric vector
- **YCOO** a numeric vector
- **Ag** a numeric vector
- **Ag_INAA** a numeric vector
- **Al** a numeric vector
- **Al2O3** a numeric vector
- **As** a numeric vector
- **As_INAA** a numeric vector
- **Au_INAA** a numeric vector
- **B** a numeric vector
- **Ba** a numeric vector
- **Ba_INAA** a numeric vector
- **Be** a numeric vector
- **Bi** a numeric vector
- **Br_IC** a numeric vector
- **Br_INAA** a numeric vector
- **Ca** a numeric vector
- **Ca_INAA** a numeric vector
<table>
<thead>
<tr>
<th></th>
<th>a numeric vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>CaO</td>
<td></td>
</tr>
<tr>
<td>Cd</td>
<td></td>
</tr>
<tr>
<td>Ce_INAA</td>
<td></td>
</tr>
<tr>
<td>Cl_IC</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Co</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Co_INAA</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>EC</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Cr</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Cr_INAA</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Cs_INAA</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Cu</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Cu_INAA</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Eu_INAA</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>F_IC</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Fe</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Fe_INAA</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Fe2O3</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Hf_INAA</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Hg</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Hg_INAA</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Ir_INAA</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>K</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>K2O</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>La</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>La_INAA</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Li</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>LOI</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Lu_INAA</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>wt_INAA</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Mg</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>MgO</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Mn</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>MnO</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Mo</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Mo_INAA</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Na</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Na_INAA</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Na20</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Nd_INAA</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Ni</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Ni_INAA</td>
<td>numeric vector</td>
</tr>
<tr>
<td>NO3_IC</td>
<td>numeric vector</td>
</tr>
<tr>
<td>P</td>
<td>numeric vector</td>
</tr>
<tr>
<td>P2O5</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Pb</td>
<td>numeric vector</td>
</tr>
<tr>
<td>pH</td>
<td>numeric vector</td>
</tr>
<tr>
<td>PO4_IC</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Rb</td>
<td>numeric vector</td>
</tr>
<tr>
<td>S</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Sb</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Sb_INAA</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Sc</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Sc_INAA</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Se</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Se_INAA</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Si</td>
<td>numeric vector</td>
</tr>
<tr>
<td>SiO2</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Sm_INAA</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Sn_INAA</td>
<td>numeric vector</td>
</tr>
<tr>
<td>SO4_IC</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Sr</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Sr_INAA</td>
<td>numeric vector</td>
</tr>
<tr>
<td>SUM_XRF</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Ta_INAA</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Tb_INAA</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Te</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Th</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Th_INAA</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Ti</td>
<td>numeric vector</td>
</tr>
<tr>
<td>TiO2</td>
<td>numeric vector</td>
</tr>
<tr>
<td>U_INAA</td>
<td>numeric vector</td>
</tr>
<tr>
<td>V</td>
<td>numeric vector</td>
</tr>
<tr>
<td>W_INAA</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Y</td>
<td>numeric vector</td>
</tr>
<tr>
<td>Yb_INAA</td>
<td>numeric vector</td>
</tr>
</tbody>
</table>
Zn  a numeric vector
Zn_INAA  a numeric vector
ELEV  a numeric vector
COUN  a numeric vector
ASP  a numeric vector
TOPC  a numeric vector
LIT0  a numeric vector
AL_XRF  a numeric vector
Ca_XRF  a numeric vector
Fe_XRF  a numeric vector
K_XRF  a numeric vector
Mg_XRF  a numeric vector
Mn_XRF  a numeric vector
Na_XRF  a numeric vector
P_XRF  a numeric vector
Si_XRF  a numeric vector
Ti_XRF  a numeric vector

Source

References

Examples

data(chorizon)
# classical versus robust correlation
corr.plot(log(chorizon[,"Al"]), log(chorizon[,"Na"]))
**Description**

The function `color.plot` plots the (two-dimensional) data using different symbols according to the robust mahalanobis distance based on the mcd estimator with adjustment and using different colors according to the euclidean distances of the observations.

**Usage**

```
color.plot(x, quan=1/2, alpha=0.025, ...)
```

**Arguments**

- `x` : two dimensional matrix or data.frame containing the data.
- `quan` : amount of observations which are used for mcd estimations. has to be between 0.5 and 1, default ist 0.5
- `alpha` : amount of observations used for calculating the adjusted quantile (see function arw).
- `...` : additional graphical parameters

**Details**

The function `color.plot` plots the (two-dimensional) data using different symbols (see function `symbol.plot`) according to the robust mahalanobis distance based on the mcd estimator with adjustment and using different colors according to the euclidean distances of the observations. Blue is typical for a little distance, whereas red is the opposite. In addition four ellipsoids are drawn, on which mahalanobis distances are constant. These constant values correspond to the 25%, 50%, 75% and adjusted quantiles (see function arw) of the chi-square distribution (see Filzmoser et al., 2005).

**Value**

- `outliers` : boolean vector of outliers
- `md` : robust mahalanobis distances of the data
- `euclidean` : euclidean distances of the observations according to the minimum of the data.

**Author(s)**

Moritz Gschwandtner &lt;e0125439@student.tuwien.ac.at&gt;  
Peter Filzmoser &lt;P.Filzmoser@tuwien.ac.at&gt; http://cstat.tuwien.ac.at/filz/

**References**

See Also

- symbol.plot
- dd.plot
- arw

Examples

```r
# create data:
x <- cbind(rnorm(100), rnorm(100))
y <- cbind(rnorm(10, 5, 1), rnorm(10, 5, 1))
z <- rbind(x, y)
# execute:
color.plot(z, quan=0.75)
```

corr.plot

**Correlation Plot: robust versus classical bivariate correlation**

Description

The function `corr.plot` plots the (two-dimensional) data and adds two correlation ellipsoids, based on classical and robust estimation of location and scatter. Robust estimation can be thought of as estimating the mean and covariance of the 'good' part of the data.

Usage

```r
corr.plot(x, y, quan=1/2, alpha=0.025, ...)
```

Arguments

- `x`: vector to be plotted against `y` and of which the correlation with `y` is calculated.
- `y`: vector to be plotted against `x` and of which the correlation with `x` is calculated.
- `quan`: amount of observations which are used for `mcd` estimations. has to be between 0.5 and 1, default is 0.5
- `alpha`: Determines the size of the ellipsoids. An observation will be outside of the ellipsoid if its mahalanobis distance exceeds the 1-alpha quantile of the chi-squared distribution.
- `...`: additional graphical parameters

Value

- `cor.cla`: correlation between `x` and `y` based on classical estimation of location and scatter
- `cor.rob`: correlation between `x` and `y` based on robust estimation of location and scatter

Author(s)

Moritz Gschwandtner <<e0125439@student.tuwien.ac.at>>
Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> [http://cstat.tuwien.ac.at/filz/](http://cstat.tuwien.ac.at/filz/)
**dat**

**See Also**

`covMcd`

**Examples**

```r
# create data:
x <- cbind(rnorm(100), rnorm(100))
y <- cbind(rnorm(10, 3, 1), rnorm(10, 3, 1))
z <- rbind(x,y)
# execute:
corr.plot(z[,1], z[,2])
```

---

**Description**

Illustrative data example with 100 observations in two dimensions.

**Usage**

```r
data(dat)
```

**Format**

The format is: num [1:100, 1:2] 3.39 4.08 4.35 4.89 4.55 ...

**Details**

Data are constructed to contain global as well as local outliers.

**Source**


**References**


**Examples**

```r
data(dat)
plot(dat)
```
dd.plot

**Distance-Distance Plot**

**Description**

The function `dd.plot` plots the classical mahalanobis distance of the data against the robust mahalanobis distance based on the mcd estimator. Different symbols (see function `symbol.plot`) and colours (see function `color.plot`) are used depending on the mahalanobis and euclidean distance of the observations (see Filzmoser et al., 2005).

**Usage**

```
dd.plot(x, quan=1/2, alpha=0.025, ...)  
```

**Arguments**

- `x`: matrix or data frame containing the data
- `quan`: amount of observations which are used for mcd estimations. has to be between 0.5 and 1, default ist 0.5
- `alpha`: amount of observations used for calculating the adjusted quantile (see function `arw`).
- `...`: additional graphical parameters

**Value**

- `outliers`: boolean vector of outliers
- `md.cla`: mahalanobis distances of the observations based on classical estimators of location and scatter.
- `md.rob`: mahalanobis distances of the observations based on robust estimators of location and scatter (mcd).

**Author(s)**

Moritz Gschwandtner <<e@125439@student.tuwien.ac.at>>
Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> [http://cstat.tuwien.ac.at/filz/](http://cstat.tuwien.ac.at/filz/)

**References**


**See Also**

`symbol.plot`, `color.plot`, `arw`, `covPlot`
Examples

```r
# create data:
x <- cbind(rnorm(100), rnorm(100))
y <- cbind(rnorm(10, 3, 1), rnorm(10, 3, 1))
z <- rbind(x, y)
# execute:
dd.plot(z)
#
# Identify multivariate outliers for Co-Cu-Ni in humus layer of Kola data:
data(humus)
dd.plot(log(humus[,c("Co","Cu","Ni")]))
```

---

**humus**

*Humus Layer (O-horizon) of the Kola Data*

**Description**

The Kola Data were collected in the Kola Project (1993-1998, Geological Surveys of Finland (GTK) and Norway (NGU) and Central Kola Expedition (CKE), Russia). More than 600 samples in five different layers were analysed, this dataset contains the humus layer.

**Usage**

data(humus)

**Format**

A data frame with 617 observations on the following 44 variables.

- **ID**: a numeric vector
- **XCOO**: a numeric vector
- **YCOO**: a numeric vector
- **Ag**: a numeric vector
- **Al**: a numeric vector
- **As**: a numeric vector
- **B**: a numeric vector
- **Ba**: a numeric vector
- **Be**: a numeric vector
- **Bi**: a numeric vector
- **Ca**: a numeric vector
- **Cd**: a numeric vector
- **Co**: a numeric vector
- **Cr**: a numeric vector
Cu  a numeric vector
Fe  a numeric vector
Hg  a numeric vector
K  a numeric vector
La  a numeric vector
Mg  a numeric vector
Mn  a numeric vector
Mo  a numeric vector
Na  a numeric vector
Ni  a numeric vector
P  a numeric vector
Pb  a numeric vector
Rb  a numeric vector
S  a numeric vector
Sb  a numeric vector
Sc  a numeric vector
Si  a numeric vector
Sr  a numeric vector
Th  a numeric vector
Tl  a numeric vector
U  a numeric vector
V  a numeric vector
Y  a numeric vector
Zn  a numeric vector
C  a numeric vector
H  a numeric vector
N  a numeric vector
LOI  a numeric vector
pH  a numeric vector
Cond  a numeric vector

**Source**


**References**

Examples

```r
data(humus)
# classical versus robust correlation:
corr.plot(log(humus[, "Al"]), log(humus[, "Na"]))
```

**Description**

Coordinates of the Kola background map

**Usage**

```r
data(kola.background)
```

**Format**


**Details**

Is used by map.plot()

**Source**


**References**

locoutNeighbor  

Diagnostic plot for identifying local outliers with varying size of neighborhood

Description
Computes global and pairwise Mahalanobis distances for visualizing global and local multivariate outliers. The size of the neighborhood (number of neighbors) is varying, but the fraction of neighbors is fixed.

Usage
locoutNeighbor(dat, X, Y, propneighb = 0.1, variant = c("dist", "knn"), usemax = 1/3, npoints = 50, chisqqu = 0.975, indices = NULL, xlab = NULL, ylab = NULL, colall = gray(0.7), colsel = 1, ...)

Arguments
- dat: multivariate data set (without coordinates)
- X: X coordinates of the data points
- Y: Y coordinates of the data points
- propneighb: proportion of neighbors to be included in tolerance ellipse
- variant: either search for neighbors according to the Eucl.Distance, or according to kNN
- usemax: for either variant: give fraction of points (max Dist) that is used for the plot
- npoints: computation is done at most at npoints points
- chisqqu: quantile of the chisquare distribution for splitting the plot
- indices: if this is not NULL, these should be indices of observations to be highlighted
- xlab: x-axis label for plot
- ylab: y-axis label for plot
- colall: color for lines if indices is NULL
- colsel: color for lines if indices are selected
- ... additional parameters for plotting

Details
For this diagnostic tool, the number of neighbors is varied up to a fraction of usemax observations. Then propneighb (called beta) is fixed, and for each observation we compute the degree of isolation from a fraction of 1-beta of its neighbors. Neighborhood can be defined either via the Euclidean distance or by k-Nearest-Neighbors. For computational reasons, all computations are done at most for npoints points. The critical value for outliers is the quantile chisqqu of the chisquare distribution. One can also provide indices of observations (for indices). Then the corresponding lines in the plots will be highlighted.
**locoutPercent**

### Value

- `indices.reg` indices of the (selected) observations being regular observations
- `indices.out` indices of the (selected) observations being global outliers

### Author(s)

Peter Filzmoser &lt;P.Filzmoser@tuwien.ac.at&gt; [http://cstat.tuwien.ac.at/filz/](http://cstat.tuwien.ac.at/filz/)

### References


### See Also

`locoutPercent, locoutSort`

### Examples

```r
# use data from illustrative example in paper:
data(X)
data(Y)
data(dat)
res <- locoutNeighbor(dat,X,Y,variant="knn",usemax=1,chisqqu=0.975,indices=c(1,11,24,36),
propneighb=0.1,npoints=100)
```

---

**locoutPercent**

*Diagnostic plot for identifying local outliers with fixed size of neighborhood*

### Description

Computes global and pairwise Mahalanobis distances for visualizing global and local multivariate outliers. The size of the neighborhood (number of neighbors) is fixed, but the fraction of neighbors is varying.

### Usage

```
locoutPercent(dat, X, Y, dist = NULL, k = 10, chisqqu = 0.975, sortup = 10, sortlow = 90,
nlinesup = 10, nlineslow = 10, indices = NULL, xlab = "(Sorted) Index",
 ylab = "Distance to neighbor", col = gray(0.7), ...)
```
locoutPercent

Arguments

dat   multivariate data set (without coordinates)
X     X coordinates of the data points
Y     Y coordinates of the data points
dist  maximum distance to search for neighbors; if nothing is provided, k for kNN is used
k     number of nearest neighbors to search - not taken if a value for dist is provided
chisqqu quantile of the chisquare distribution for splitting the plot
sortup sort local outliers according to given percentage
sortlow sort local inliers according to given percentage
nlinesup number of lines to be plotted for upper part
nlineslow number of lines to be plotted for lower part
indices if this is not NULL, these should be indices of observations to be highlighted
xlab   x-axis label for plot
ylab   y-axis label for plot
col    color for lines
...    additional parameters for plotting

Details

For this diagnostic tool, the number of neighbors is fixed, but propneighb (called beta) is varied. For each observation we compute the degree of isolation from a fraction of 1-beta of its neighbors. Neighborhood can be defined either via the Euclidean distance or by k-Nearest-Neighbors. The critical value for outliers is the quantile chisqqu of the chisquare distribution. One can also provide indices of observations (for indices). Then the corresponding lines in the plots will be highlighted.

Value

ret    list containing indices of regular and outlying observations

Author(s)

Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> http://cstat.tuwien.ac.at/filz/

References


See Also

locoutNeighbor, locoutSort
Examples

```r
# use data from illustrative example in paper:
data(X)
data(Y)
data(dat)
res <- locoutPercent(dat, X, Y, k=10, chisqqu=0.975, indices=c(1,11,24,36))
```

**locoutSort**

*Interactive diagnostic plot for identifying local outliers*

**Description**

Computes global and pairwise Mahalanobis distances for visualizing global and local multivariate outliers. The plot is split into regular (left) and global (right) outliers, and points can be selected interactively. In a second plot, these points are shown by spatial coordinates.

**Usage**

```r
locoutSort(dat, X, Y, distc = NULL, k = 10, propneighb = 0.1, chisqqu = 0.975, sel = NULL, ...)
```

**Arguments**

- `dat` multivariate data set (without coordinates)
- `X` X coordinates of the data points
- `Y` Y coordinates of the data points
- `distc` maximum distance to search for neighbors; if nothing is provided, k for kNN is used
- `k` number of nearest neighbors to search - not taken if a value for dist is provided
- `propneighb` proportion of neighbors to be included in tolerance ellipse
- `chisqqu` quantile of the chi-square distribution for splitting the plot
- `sel` optional list with x and y, i.e. coordinates with selected polygon
- `...` additional parameters for plotting

**Details**

For this diagnostic tool, the number of neighbors is fixed, and propneighb (called beta) is also fixed. For each observation we compute the degree of isolation from a fraction of 1-beta of its neighbors. The observations are sorted according to this degree of isolation, and this sorted index forms the x-axis of the left plot. This plot is also split into regular (left) and global (right) outliers. Then one can select with the mouse a region in this plot, meaning an observation and (some of) its neighbors. Alternatively, this region can be supplied by sel. The selected observations are then shown in the right plot. Links to the neighbors are also shown.
Value

list(sel=sel,index.regular=res$indices.regular,index.outliers=res$indices.outliers)

sel       plot coordinates of the selected region
indices.reg indices of the observations being regular observations
indices.out indices of the observations being global outliers

Author(s)

Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> [http://cstat.tuwien.ac.at/filz/]

References

Submitted for publication, 2012.

See Also

locoutPercent, locoutNeighbor

Examples

# use data from illustrative example in paper:
data(X)
data(Y)
data(dat)

 sel <- locoutSort(dat,X,Y,k=10,propneighb=0.1,chisqu=0.975,
                     sel=list(x=c(87.5,87.5,89.3,89.3),y=c(4.3,0.7,0.7,4.3)))

Description

The function map.plot creates a map using geographical (x,y)-coordinates. This is thought for
spatially dependent data of which coordinates are available. Multivariate outliers are marked.

Usage

map.plot(coord, data, quan=1/2, alpha=0.025, symb=FALSE, plotmap=TRUE,
         map="kola.background",which.map=c(1,2,3,4),map.col=c(5,1,3,4),
         map.lwd=c(2,1,2,1),... )
**Arguments**

- `coord` (x,y)-coordinates of the data
- `data` matrix or data.frame containing the data.
- `quan` amount of observations which are used for mcd estimations. has to be between 0.5 and 1, default ist 0.5
- `alpha` amount of observations used for calculating the adjusted quantile (see function `arw`).
- `symb` logical for plotting special symbols (see details).
- `plotmap` logical for plotting the background map.
- `map` see `plot.kola.background()`
- `which.map` see `plot.kola.background()`
- `map.col` see `plot.kola.background()`
- `map.lwd` see `plot.kola.background()`
- `...` additional graphical parameters

**Details**

The function `map.plot` shows multivariate outliers in a map. If `symb=FALSE` (default), only two colors and no special symbols are used to mark multivariate outliers (the outliers are marked red). If `symb=TRUE` different symbols and colors are used. The symbols (cross means big value, circle means little value) are selected according to the robust mahalanobis distance based on the adjusted mcd estimator (see function `symbol.plot`) Different colors (red means big value, blue means little value) according to the euclidean distances of the observations (see function `color.plot`) are used. For details see Filzmoser et al. (2005).

**Value**

- `outliers` boolean vector of outliers
- `md` robust mahalanobis distances of the data
- `euclidean` (only if symb=TRUE) euclidean distances of the observations according to the minimum of the data.

**Author(s)**

Moritz Gschwandtner <<e0125439@student.tuwien.ac.at>>
Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> [http://cstat.tuwien.ac.at/filz/](http://cstat.tuwien.ac.at/filz/)

**References**


**See Also**

`symbol.plot`, `color.plot`, `arw`
**Examples**

```r
data(humus) # Load humus data
xy <- humus[,c("XCOO","YCOO")]] # X and Y Coordinates
myhumus <- log(humus[, c("As", "Cd", "Co", "Cu", "Mg", "Pb", "Zn")])
map.plot(xy, myhumus, symb=TRUE)
```

---

**moss**

*Moss Layer of the Kola Data*

---

**Description**

The Kola Data were collected in the Kola Project (1993-1998, Geological Surveys of Finland (GTK) and Norway (NGU) and Central Kola Expedition (CKE), Russia). More than 600 samples in five different layers were analysed, this dataset contains the moss layer.

**Usage**

```r
data(moss)
```

**Format**

A data frame with 598 observations on the following 34 variables.

- **ID**  a numeric vector
- **XCOO** a numeric vector
- **YCOO** a numeric vector
- **Ag**  a numeric vector
- **Al**  a numeric vector
- **As**  a numeric vector
- **B**  a numeric vector
- **Ba**  a numeric vector
- **Bi**  a numeric vector
- **Ca**  a numeric vector
- **Cd**  a numeric vector
- **Co**  a numeric vector
- **Cr**  a numeric vector
- **Cu**  a numeric vector
- **Fe**  a numeric vector
- **Hg**  a numeric vector
- **K**  a numeric vector
- **Mg**  a numeric vector
- **Mn**  a numeric vector
mvoutlier.CoDa

Mo a numeric vector
Na a numeric vector
Ni a numeric vector
P a numeric vector
Pb a numeric vector
Rb a numeric vector
S a numeric vector
Sb a numeric vector
Si a numeric vector
Sr a numeric vector
Th a numeric vector
Tl a numeric vector
U a numeric vector
V a numeric vector
Zn a numeric vector

Source


References


Examples

data(moss)
# classical versus robust correlation:
corr.plot(log(moss[,"Al"])), log(moss[,"Na"]))

mvoutlier.CoDa

Interpreting multivariate outliers of CoDa

Description

Computes the basis information for plot functions supporting the interpretation of multivariate outliers in case of compositional data.
Usage

mvoutlier.CoDa(x, quan = 0.75, alpha = 0.025,
col.quantile = c(0, 0.05, 0.1, 0.5, 0.9, 0.95, 1),
symb.pch = c(3, 3, 16, 1, 1), symb.cex = c(1.5, 1, 0.5, 1, 1.5),
adaptive = TRUE)

Arguments

x data set (matrix or data frame) containing the raw untransformed compositional data
quan quantity of data used for robust estimation; between 0.5 and 1
alpha maximum threshold for adaptive outlier detection
col.quantile quantiles of an average concentration defining the colors
symb.pch plotting character for symbols
symb.cex plotting size for symbols
adaptive if TRUE then the adaptive method for the outlier threshold is used

Details

In a first step, the raw compositional data set in transformed by the isometric logratio (ilr) transformation to the usual Euclidean space. Then adaptive outlier detection is performed: Starting from a quantile 1-alpha of the chisquare distribution, one looks for the supremum of the differences between the chisquare distribution and the empirical distribution of the squared Mahalanobis distances. The latter are derived from the MCD estimator using the proportion quan of the data. The supremum is the outlier cutoff, and certain colors and symbols for the outliers are computed: The colors should reflect the magnitude of the median element concentration of the observations, which is done by computing for each observation along the single ilr variables the distances to the medians. The mediab of all distances determines the color (or grey scale): a high value, resulting in a red (or dark) symbol, means that most univariate parts have higher values than the average, and a low value (blue or light symbol) refers to an observation with mainly low values. The symbols are according to the cut-points from the quantiles 0.25, 0.5, 0.75, and the outlier cutoff of the squared Mahalanobis distances.

Value

ilrvariables the ilr transformed data matrix
outliers TRUE/FALSE vector; TRUE refers to outlier
pcaobj object from PCA
colcol vector with the colors
colbw vector with the grey scales
pchvec vector with plotting symbols
cexvec vector with sizes of plot symbols

Author(s)

Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> http://cstat.tuwien.ac.at/filz/
References


See Also

plot.mvoutlierCoDa, arw, map.plot, uni.plot

Examples

data(humus)
d <- humus[,c("As","Cd","Co","Cu","Mg","Pb","Zn")]
res <- mvoutlier.CoDa(d)
str(res)

pbb

BSS background Plot

Description

Plots the BSS background map

Usage

pbb(map = "bss.background", add.plot = FALSE, ...)

Arguments

map
add.plot
...

List of coordinates. For the correct format see also help(kola.background)
logical. If true background is added to an existing plot
additional plot parameters, see help(par)

Details

The list of coordinates is plotted as a polygon line.

Value

The plot is produced on the graphical device.

Author(s)

Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> http://cstat.tuwien.ac.at/filz/
References


See Also

See also pkb

Examples

data(bss.background)
data(bsstop)
plot(bsstop$XCOO, bsstop$YCOO, col="red", pch=3)
pbb(add=TRUE)

pcout

PCOut Method for Outlier Identification in High Dimensions

Description

Fast algorithm for identifying multivariate outliers in high-dimensional and/or large datasets, using the algorithm of Filzmoser, Maronna, and Werner (CSDA, 2007).

Usage

pcout(x, makeplot = FALSE, explvar = 0.99, crit.M1 = 1/3, crit.c1 = 2.5, crit.M2 = 1/4, crit.c2 = 0.99, cs = 0.25, outbound = 0.25, ...)

Arguments

x a numeric matrix or data frame which provides the data for outlier detection
makeplot a logical value indicating whether a diagnostic plot should be generated (default to FALSE)
explvar a numeric value between 0 and 1 indicating how much variance should be covered by the robust PCs (default to 0.99)
crit.M1 a numeric value between 0 and 1 indicating the quantile to be used as lower boundary for location outlier detection (default to 1/3)
crit.c1 a positive numeric value used for determining the upper boundary for location outlier detection (default to 2.5)
crit.M2 a numeric value between 0 and 1 indicating the quantile to be used as lower boundary for scatter outlier detection (default to 1/4)
crit.c2 a numeric value between 0 and 1 indicating the quantile to be used as upper boundary for scatter outlier detection (default to 0.99)
pcout

\texttt{cs} \quad \text{a numeric value indicating the scaling constant for combined location and scatter weights (default to 0.25)}

\texttt{outbound} \quad \text{a numeric value between 0 and 1 indicating the outlier boundary for defining values as final outliers (default to 0.25)}

\ldots \quad \text{additional plot parameters, see help(par)}

\textmd{Details}

Based on the robustly sphered data, semi-robust principal components are computed which are needed for determining distances for each observation. Separate weights for location and scatter outliers are computed based on these distances. The combined weights are used for outlier identification.

\textmd{Value}

\texttt{wfinal01} \quad 0/1 vector with final weights for each observation; weight 0 indicates potential multivariate outliers.

\texttt{wfinal} \quad \text{numeric vector with final weights for each observation; small values indicate potential multivariate outliers.}

\texttt{wloc} \quad \text{numeric vector with weights for each observation; small values indicate potential location outliers.}

\texttt{wscat} \quad \text{numeric vector with weights for each observation; small values indicate potential scatter outliers.}

\texttt{x.dist1} \quad \text{numeric vector with distances for location outlier detection.}

\texttt{x.dist2} \quad \text{numeric vector with distances for scatter outlier detection.}

\texttt{M1} \quad \text{upper boundary for assigning weight 1 in location outlier detection.}

\texttt{const1} \quad \text{lower boundary for assigning weight 0 in location outlier detection.}

\texttt{M2} \quad \text{upper boundary for assigning weight 1 in scatter outlier detection.}

\texttt{const2} \quad \text{lower boundary for assigning weight 0 in scatter outlier detection.}

\textmd{Author(s)}

Peter Filzmoser \texttt{<<P.Filzmoser@tuwien.ac.at>>} \url{http://cstat.tuwien.ac.at/filz/}

\textmd{References}


\textmd{See Also}

\texttt{sign1, sign2}
Examples

```r
# geochemical data from northern Europe
data(bsstop)
x=bsstop[,5:14]
# identify multivariate outliers
x.out=pcout(x,makeplot=FALSE)
# visualize multivariate outliers in the map
op <- par(mfrow=c(1,2))
data(bss.background)
pbb(asp=1)
points(bsstop$XCOO,bsstop$YCOO,pch=16,col=x.out$wfinal01+2)
title("Outlier detection based on pcout")
legend("topleft",legend=c("potential outliers","regular observations"),pch=16,col=c(2,3))

# compare with outlier detection based on MCD:
x.mcd <- robustbase::covMcd(x)
pbb(asp=1)
points(bsstop$XCOO,bsstop$YCOO,pch=16,col=x.mcd$mcd.wt+2)
title("Outlier detection based on MCD")
legend("topleft",legend=c("potential outliers","regular observations"),pch=16,col=c(2,3))
par(op)
```

---

**pkb**

**Kola background Plot**

**Description**

Plots the Kola background map

**Usage**

```r
pkb(map = "kola.background", which.map = c(1, 2, 3, 4), map.col = c(5, 1, 3, 4),
    map.lwd = c(2, 1, 2, 1), add.plot = FALSE, ...)
```

**Arguments**

- `map` List of coordinates. For the correct format see also help(kola.background)
- `which.map` which==1 ... plot project boundary \# which==2 ... plot coast line \# which==3 ... plot country borders \# which==4 ... plot lakes and rivers
- `map.col` Map colors to be used
- `map.lwd` Defines linestyle of the background
- `add.plot` logical. if true background is added to an existing plot
- `...` additional plot parameters, see help(par)

**Details**

Is used by map.plot()
**plot.mvoutlierCoDa**

**Author(s)**
Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> [http://cstat.tuwien.ac.at/filz/](http://cstat.tuwien.ac.at/filz/)

**References**

**Examples**
```r
example(map.plot)
```

---

**plot.mvoutlierCoDa**  
*Plots for interpreting multivariate outliers of CoDa*

**Description**
Plots the computed information by `mvoutlier.CoDa` for supporting the interpretation of multivariate outliers in case of compositional data.

**Usage**
```r
## S3 method for class 'mvoutlierCoDa'
plot(x, ..., which = c("biplot", "map", "uni", "parallel"), 
     choice = 1:2, coord = NULL, map = NULL, onlyout = TRUE, bw = FALSE, symb = TRUE, 
     symbtxt = FALSE, col = NULL, pch = NULL, obj.cex = NULL, transp = 1)
```

**Arguments**
- `x` : resulting object from function `mvoutlier.CoDa`
- `...` : further plotting arguments
- `which` : type of plot that should be made
- `choice` : select the pair of PCs used for the biplot
- `coord` : coordinates for the presentation in a map
- `map` : coordinates for the background map; see details below
- `onlyout` : if TRUE only the outliers are shown in the plot
- `bw` : if TRUE symbold will be in grey scale rather than in color
- `symb` : if TRUE special symbols are used according to outlyingness
- `symbtxt` : if TRUE text labels are used for plotting
- `col` : define colors to be used for outliers and non-outliers
- `pch` : define plotting symbols to be used for outliers and non-outliers
- `obj.cex` : define symbol size for outliers and non-outliers
- `transp` : define transparancy for parallel coordinate plot
Details

The function `mvoutlier.CoDa` prepares the information needed for this plot function: In a first step, the raw compositional data set is transformed by the isometric logratio (ilr) transformation to the usual Euclidean space. Then adaptive outlier detection is performed: Starting from a quantile 1-alpha of the chi-square distribution, one looks for the supremum of the differences between the chi-square distribution and the empirical distribution of the squared Mahalanobis distances. The latter are derived from the MCD estimator using the proportion quan of the data. The supremum is the outlier cutoff, and certain colors and symbols for the outliers are computed: The colors should reflect the magnitude of the median element concentration of the observations, which is done by computing for each observation along the single ilr variables the distances to the medians. The median of all distances determines the color (or grey scale): a high value, resulting in a red (or dark) symbol, means that most univariate parts have higher values than the average, and a low value (blue or light symbol) refers to an observation with mainly low values. The symbols are according to the cut-points from the quantiles 0.25, 0.5, 0.75, and the outlier cutoff of the squared Mahalanobis distances. This plot function then allows to visualize the information.

The optional background map for the representation of the outliers in a map can be included using the argument `map`. This should consist of one or more polygons representing the geographical x- and y- coordinates of the background map. Of course, this map should be represented in the same coordinate system as the coordinates for the sample locations provided by `coord`. The structure of `map` is as follows: It should consist of 2 columns, one for the x-, one for the y-coordinates. If a polygon ends, a row with 2 entries NA should follow. At the end two NA rows are needed. See also examples below.

Value

A plot is drawn.

Author(s)

Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> http://cstat.tuwien.ac.at/filz/

References


See Also

`mvoutlier.CoDa`, `arw`, `map.plot`, `uni.plot`

Examples

data(humus)
el=c("As","Cd","Co","Cu","Mg","Pb","Zn")
dsel <- humus[,el]
data(kola.background) # contains different information (coast, borders, etc.)
coo <- rbind(kola.background$coast,kola.background$boundary,kola.background$boundaries)
XY <- humus[,c("XCOO","YCOO")]
set.seed(123)
```r
res <- mvoutlier.CoDa(dsel)

par(ask=TRUE)
### Parallel coordinate plot:
## show for all observations (transp is only useful when generating e.g. a pdf):
# plot(res,onlyout=FALSE,bw=TRUE,which="parallel",symb=FALSE,symbtxt=FALSE,transp=0.3)
## show only outliers with special colors and labels in the margins:
plot(res,onlyout=TRUE,bw=FALSE,which="parallel",symb=TRUE,symbtxt=TRUE,transp=0.3)

### Biplot:
## show all data points, outliers are in different color and have different symbol:
# plot(res,onlyout=FALSE,which="biplot",bw=FALSE,symb=FALSE,symbtxt=FALSE)
## show only the outliers with special symbols and colors:
plot(res,onlyout=TRUE,which="biplot",bw=FALSE,symb=TRUE,symbtxt=TRUE)

### Map:
## show all data points, outliers are in different color and have different symbol:
# plot(res,coord=XY,map=coo,onlyout=FALSE,which="map",bw=FALSE,symb=FALSE,symbtxt=FALSE)
## show only the outliers with special symbols and colors:
plot(res,coord=XY,map=coo,onlyout=TRUE,which="map",bw=FALSE,symb=TRUE,symbtxt=TRUE)

### Univariate scatterplot:
## show all data points, outliers are in different color and have different symbol:
# plot(res,onlyout=FALSE,which="uni",symb=FALSE,symbtxt=FALSE)
## show only the outliers with special symbols and colors:
plot(res,onlyout=TRUE,which="uni",symb=TRUE,symbtxt=TRUE)
```

---

**sign1**

*Sign Method for Outlier Identification in High Dimensions - Simple Version*

**Description**

Fast algorithm for identifying multivariate outliers in high-dimensional and/or large datasets, using spatial signs, see Filzmoser, Maronna, and Werner (CSDA, 2007). The computation of the distances is based on Mahalanobis distances.

**Usage**

`sign1(x, makeplot = FALSE, qcrit = 0.975, ...)`

**Arguments**

- `x` : a numeric matrix or data frame which provides the data for outlier detection
- `makeplot` : a logical value indicating whether a diagnostic plot should be generated (default to FALSE)
- `qcrit` : a numeric value between 0 and 1 indicating the quantile to be used as critical value for outlier detection (default to 0.975)
- `...` : additional plot parameters, see help(par)
Details

Based on the robustly sphered and normed data, robust principal components are computed. These are used for computing the covariance matrix which is the basis for Mahalanobis distances. A critical value from the chi-square distribution is then used as outlier cutoff.

Value

- `wfinal01` 0/1 vector with final weights for each observation; weight 0 indicates potential multivariate outliers.
- `x.dist` numeric vector with distances used for outlier detection.
- `const` outlier cutoff value.

Author(s)

Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> [http://cstat.tuwien.ac.at/filz/]

References


See Also

`pcout`, `sign2`

Examples

```r
# geochemical data from northern Europe
data(bsstop)
xt=bsstop[,5:14]
# identify multivariate outliers
x.out=sign1(xt,makeplot=FALSE)
# visualize multivariate outliers in the map
op <- par(mfrow=c(1,2))
data(bss.background)
pbb(asp=1)
points(bsstop$XCOO,bsstop$YCOO,pch=16,col=x.out$wfinal01+2)
title("Outlier detection based on signout")
legend("topleft",legend=c("potential outliers","regular observations"),pch=16,col=c(2,3))

# compare with outlier detection based on MCD:
x.mcd <- robustbase::covMcd(xt)
pbb(asp=1)
points(bsstop$XCOO,bsstop$YCOO,pch=16,col=x.mcd$mcd.wt+2)
title("Outlier detection based on MCD")
legend("topleft",legend=c("potential outliers","regular observations"),pch=16,col=c(2,3))
par(op)
```
Description

Fast algorithm for identifying multivariate outliers in high-dimensional and/or large datasets, using spatial signs, see Filzmoser, Maronna, and Werner (CSDA, 2007). The computation of the distances is based on principal components.

Usage

```r
sign2(x, makeplot = FALSE, explvar = 0.99, qcrit = 0.975, ...)
```

Arguments

- `x`: a numeric matrix or data frame which provides the data for outlier detection
- `makeplot`: a logical value indicating whether a diagnostic plot should be generated (default to FALSE)
- `explvar`: a numeric value between 0 and 1 indicating how much variance should be covered by the robust PCs (default to 0.99)
- `qcrit`: a numeric value between 0 and 1 indicating the quantile to be used as critical value for outlier detection (default to 0.975)
- `...`: additional plot parameters, see help(par)

Details

Based on the robustly sphered and normed data, robust principal components are computed which are needed for determining distances for each observation. The distances are transformed to approach chi-square distribution, and a critical value is then used as outlier cutoff.

Value

- `wfinal01`: 0/1 vector with final weights for each observation; weight 0 indicates potential multivariate outliers.
- `x.dist`: numeric vector with distances used for outlier detection.
- `const`: outlier cutoff value.

Author(s)

Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> [http://cstat.tuwien.ac.at/filz/](http://cstat.tuwien.ac.at/filz/)

References


See Also

pcout, sign1

Examples

```r
# geochemical data from northern Europe
data(bsstop)
x <- bsstop[,5:14]
# identify multivariate outliers
x.out <- sign2(x, makeplot = FALSE)
# visualize multivariate outliers in the map
op <- par(mfrow = c(1, 2))
data(bsstop$background)
pbb(asp = 1)
points(bsstop$XCOO, bsstop$YCOO, pch = 16, col = x.out$wfinal01 + 2)
title("Outlier detection based on signout")
legend("topleft", legend = c("potential outliers", "regular observations"), pch = 16, col = c(2, 3))

# compare with outlier detection based on MCD:
x.mcd <- robustbase::covMcd(x)
pbb(asp = 1)
points(bsstop$XCOO, bsstop$YCOO, pch = 16, col = x.mcd$mcd.wt + 2)
title("Outlier detection based on MCD")
legend("topleft", legend = c("potential outliers", "regular observations"), pch = 16, col = c(2, 3))
par(op)
```

symbol.plot

Symbol Plot

Description

The function symbol.plot plots the (two-dimensional) data using different symbols according to the robust mahalanobis distance based on the mcd estimator with adjustment.

Usage

```r
symbol.plot(x, quan = 1/2, alpha = 0.025, ...)
```

Arguments

- **x**: two dimensional matrix or data.frame containing the data.
- **quan**: amount of observations which are used for mcd estimations. has to be between 0.5 and 1, default is 0.5
- **alpha**: amount of observations used for calculating the adjusted quantile (see function arw).
- **...**: additional graphical parameters
uni.plot

Details

The function symbol.plot plots the (two-dimensional) data using different symbols. In addition a legend and four ellipsoids are drawn, on which mahalanobis distances are constant. As the legend shows, these constant values correspond to the 25%, 50%, 75% and adjusted (see function arw) quantiles of the chi-square distribution.

Value

- outliers: boolean vector of outliers
- md: robust mahalanobis distances of the data

Author(s)

Moritz Gschwandtner <e0125439@student.tuwien.ac.at>
Peter Filzmoser <P.Filzmoser@tuwien.ac.at> http://cstat.tuwien.ac.at/filz/

References


See Also

dd.plot, color.plot, arw

Examples

```r
# create data:
x <- cbind(rnorm(100), rnorm(100))
y <- cbind(rnorm(10, 5, 1), rnorm(10, 5, 1))
z <- rbind(x, y)
# execute:
symbol.plot(z, quan=0.75)
```

---

uni.plot

Univariate Presentation of Multivariate Outliers

Description

The function uni.plot plots each variable of x parallel in a one-dimensional scatter plot and in addition marks multivariate outliers.

Usage

uni.plot(x, symb=FALSE, quan=1/2, alpha=0.025, ...)

Arguments

x  
    matrix or data.frame containing the data.

symb  
    logical. if FALSE, only two colors and no special symbols are used. outliers are marked red. if TRUE different symbols (cross means big value, circle means little value) according to the robust mahalanobis distance based on the mcd estimator and different colors (red means big value, blue means little value) according to the euclidean distances of the observations are used.

quan  
    amount of observations which are used for mcd estimations. has to be between 0.5 and 1, default ist 0.5

alpha  
    amount of observations used for calculating the adjusted quantile (see function arw).

...  
    additional graphical parameters

Details

The function uni.plot shows the multivariate outliers in the single variables by one-dimensional scatter plots. If symb=FALSE (default), only two colors and no special symbols are used to mark multivariate outliers (the outliers are marked red). If symb=TRUE different symbols and colors are used. The symbols (cross means big value, circle means little value) are selected according to the robust mahalanobis distance based on the adjusted mcd estimator (see function symbol.plot) Different colors (red means big value, blue means little value) according to the euclidean distances of the observations (see function color.plot) are used. For details see Filzmoser et al. (2005).

Value

outliers  
    boolean vector of outliers

md  
    robust multivariate mahalanobis distances of the data

euclidean  
    (only if symb=TRUE) multivariate euclidean distances of the observations according to the minimum of the data.

Author(s)

Moritz Gschwandtner <<e0125439@student.tuwien.ac.at>>
Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> http://cstat.tuwien.ac.at/filz/

References


See Also

map.plot, symbol.plot, color.plot, arw
Examples

data(swiss)
uni.plot(swiss)

# Geostatistical data:
data(humus) # Load humus data
uni.plot(log(humus[, c("As", "Cd", "Co", "Cu", "Mg", "Pb", "Zn")]), symb=TRUE)

Data (X coordinate) of illustrative example in paper (see below)

Description

Illustrative data example with 100 values for the X coordinate.

Usage

data(X)

Format

The format is: num [1:100] 3.72 5.1 3.33 2.13 4.42 ...

Details

Data are constructed to contain global as well as local outliers.

Source


References


Examples

data(X)
data(Y)
plot(X,Y)
Description

Illustrative data example with 100 values for the Y coordinate.

Usage

data(Y)

Format

The format is: num [1:100] 1.25 1.4 0.372 0.791 2.74 ...

Details

Data are constructed to contain global as well as local outliers.

Source


References


Examples

data(X)
data(Y)
plot(X,Y)
Index

* datasets
  bhorizon, 5
  bss.background, 7
  bsstop, 10
  chorizon, 13
  dat, 19
  humus, 21
  kola.background, 23
  moss, 30
  pbb, 33
  pkb, 36
  X, 45
  Y, 46

* dplot
  aq.plot, 2
  arw, 3
  chisq.plot, 12
  color.plot, 17
  corr.plot, 18
  dd.plot, 20
  map.plot, 28
  symbol.plot, 42
  uni.plot, 43

* multivariate
  locoutNeighbor, 24
  locoutPercent, 25
  locoutSort, 27
  mvoutlier.CoDa, 31
  pcout, 34
  plot.mvoutlierCoDa, 37
  sign1, 39
  sign2, 41

  aq.plot, 2
  arw, 3, 18, 20, 29, 33, 38, 43, 44
  bhorizon, 5
  bss.background, 7
  bsstop, 10
  chisq.plot, 12
  chorizon, 13
  color.plot, 17, 20, 29, 43, 44
  corr.plot, 18
  covMcd, 19
  covPlot, 20
  dat, 19
  dd.plot, 18, 20, 43
  humus, 21
  kola.background, 23
  locoutNeighbor, 24, 26, 28
  locoutPercent, 25, 25, 28
  locoutSort, 25, 26, 27
  map.plot, 28, 33, 38, 44
  moss, 30
  mvoutlier.CoDa, 31, 38
  pbb, 33
  pcout, 34, 40, 42
  pkb, 34, 36
  plot.mvoutlierCoDa, 33, 37
  sign1, 35, 39, 42
  sign2, 35, 40, 41

  pbb, 33
  pcout, 34, 40, 42
  pkb, 34, 36
  plot.mvoutlierCoDa, 33, 37
  sign1, 35, 39, 42
  sign2, 35, 40, 41
symbol.plot, 18, 20, 29, 42, 44
uni.plot, 33, 38, 43
X, 45
Y, 46