Package ‘robCompositions’

May 15, 2012

Type Package

Title Robust Estimation for Compositional Data.

Version 1.6.0

Date 2012-05-15

Depends R (>= 2.10), utils, robustbase, rrcov, car (>= 2.0-0), MASS

Author Matthias Templ, Karel Hron, Peter Filzmoser

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

Description The package includes methods for imputation of compositional data including robust methods, methods to impute rounded zeros, (robust) outlier detection for compositional data, (robust) principal component analysis for compositional data, (robust) factor analysis for compositional data, (robust) discriminant analysis for compositional data (Fisher rule),robust regression with compositional predictors and (robust) Anderson-Darling normality tests for compositional data as well as popular log-ratio transformations (alr, clr, ilr, and their inverse transformations). In addition, visualisation and diagnostic tools are implemented as well as high and low-level plot functions for the ternary diagram.

License GPL-2

LazyLoad yes

Repository CRAN

Date/Publication 2012-05-15 15:35:41

R topics documented:

robCompositions-package

Robust Estimation for Compositional Data.

Description

The package contains methods for imputation of compositional data including robust methods, (robust) outlier detection for compositional data, (robust) principal component analysis for compositional data, (robust) factor analysis for compositional data, (robust) discriminant analysis (Fisher rule) and (robust) Anderson-Darling normality tests for compositional data as well as popular log-ratio transformations (alr, clr, ilr, and their inverse transformations).

Details

Package:	robCompositions
Type:	Package
Version:	1.3.3
Date:	2009-11-28
License:	GPL 2
LazyLoad:	yes

Author(s)

Matthias Templ, Peter Filzmoser, Karel Hron,

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

References

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data* Monographs on Statistics and Applied Probability. Chapman \& Hall Ltd., London (UK). 416p. \

Filzmoser, P., and Hron, K. (2008) Outlier detection for compositional data using robust methods. *Math. Geosciences*, **40** 233-248.

Filzmoser, P., Hron, K., Reimann, C. (2009) Principal Component Analysis for Compositional Data with Outliers. *Environmetrics*, **20** (6), 621–632.

P. Filzmoser, K. Hron, C. Reimann, R. Garrett (2009): Robust Factor Analysis for Compositional Data. *Computers and Geosciences*, **35** (9), 1854–1861.

Hron, K. and Templ, M. and Filzmoser, P. (2010) Imputation of missing values for compositional data using classical and robust methods *Computational Statistics and Data Analysis*, **54** (12), 3095–3107.

C. Reimann, P. Filzmoser, R.G. Garrett, and R. Dutter (2008): Statistical Data Analysis Explained. *Applied Environmental Statistics with R*. John Wiley and Sons, Chichester, 2008.

Examples

```
## k nearest neighbor imputation
data(expenditures)
expenditures[1,3]
expenditures[1,3] <- NA
impKNNa(expenditures)$xImp[1,3]

## iterative model based imputation
data(expenditures)
x <- expenditures
x[1,3]
x[1,3] <- NA
xi <- impCoda(x)$xImp
xi[1,3]
s1 <- sum(x[1,-3])
impS <- sum(xi[1,-3])
xi[,3] * s1/impS

xi <- impKNNa(expenditures)
xi
summary(xi)
plot(xi, which=1)
plot(xi, which=2)
plot(xi, which=3)

## pca
data(expenditures)
p1 <- pcaCoDa(expenditures)
p1
plot(p1)

## outlier detection
data(expenditures)
oD <- outCoDa(expenditures)
oD
plot(oD)

## transformations
data(arcticLake)
x <- arcticLake
x.alr <- alr(x, 2)
y <- invalr(x.alr)
invalr(alr(x, 3))
data(expenditures)
x <- expenditures
y <- invalr(alr(x, 5))
head(x)
head(y)
invalr(x.alr, ivar=2, useClassInfo=FALSE)

data(expenditures)
eclr <- clr(expenditures)
```

```
inveclr <- invclr(eclr)
head(expenditures)
head(inveclr)
head(invclr(eclr$x.clr))

require(MASS)
Sigma <- matrix(c(5.05,4.95,4.95,5.05), ncol=2, byrow=TRUE)
z <- invilr(mvrnorm(100, mu=c(0,2), Sigma=Sigma))
```

`aDist` *Aitchison distance*

Description

Computes the Aitchison distance between two observations or between two data sets.

Usage

```
aDist(x, y)
```

Arguments

`x`	a vector, matrix or data.frame
`y`	a vector, matrix or data.frame with equal dimension as `x`

Details

This distance measure accounts for the relative scale property of the Aitchison distance. It measures the distance between two compositions if `x` and `y` are vectors and evaluate sum of the distances between `x` and `y` for each row of `x` and `y` if `x` and `y` are matrices or data frames.

It is not designed to apply it on one matrix, such as function 'acomp()' in package 'compositions', but it is designed to compare different matrices.

The underlying code is written in C and allows a fast computation also for large data sets.

Value

The Aitchison distance between two compositions or between two data sets.

Author(s)

Matthias Templ

References

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data* Monographs on Statistics and Applied Probability. Chapman and Hall Ltd., London (UK). 416p.

Aitchison, J. and Barcelo-Vidal, C. and Martin-Fernandez, J.A. and Pawlowsky-Glahn, V. (2000) Logratio analysis and compositional distance. *Mathematical Geology*, **32**, 271-275.

Hron, K. and Templ, M. and Filzmoser, P. (2010) Imputation of missing values for compositional data using classical and robust methods *Computational Statistics and Data Analysis*, vol 54 (12), pages 3095-3107.

See Also

ilr

Examples

```
data(expenditures)
x <- xOrig <- expenditures
## Aitchison distance between the first 2 observations:
aDist(x[,1], x[,2])

## set some missing values:
x[1,3] <- x[3,5] <- x[2,4] <- x[5,3] <- x[8,3] <- NA

## impute them:
xImp <- impCoda(x, method="ltsReg")$xImp

## calculate the relative Aitchsion distance between xOrig and xImp:
aDist(xOrig, xImp)
```

---

adjust *Adjusting for original scale*

Description

Results from the model based iterative methods provides the results in another scale (but the ratios are still the same). This function rescale the output to the original scale.

Usage

```
adjust(x)
```

Arguments

x	object from class 'imp'

Details

It is self-explaining if you try the examples.

Value

The object of class 'imp' but with the adjusted imputed data.

Author(s)

Matthias Templ

References

Hron, K. and Templ, M. and Filzmoser, P. (2010) Imputation of missing values for compositional data using classical and robust methods *Computational Statistics and Data Analysis*, In Press, Corrected Proof, ISSN: 0167-9473, DOI:10.1016/j.csda.2009.11.023

See Also

impCoda

Examples

```
data(expenditures)
x <- expenditures
x[1,3] <- x[2,4] <- x[3,3] <- x[3,4] <- NA
xi <- impCoda(x)
x
xi$xImp
adjust(xi)$xImp
```

---

adtest *Anderson-Darling Normality Tests*

Description

This function provides three kinds of Anderson-Darling Normality Tests (Anderson and Darling, 1952).

Usage

```
adtest(x, R = 1000, locscatt = "standard")
```

Arguments

x	either a numeric vector, or a data.frame, or a matrix
R	Number of Monte Carlo simulations to obtain p-values
locscatt	standard for classical estimates of mean and (co)variance. robust for robust estimates using 'covMcd()' from package robustbase

Details

Three version of the test are implemented (univariate, angle and radius test) and it depends on the data which test is chosen.

If the data is univariate the univariate Anderson-Darling test for normality is applied.

If the data is bivariate the angle Anderson-Darling test for normality is performed out.

If the data is multivariate the radius Anderson-Darling test for normality is used.

If 'locscatt' is equal to "robust" then within the procedure, robust estimates of mean and covariance are provided using 'covMcd()' from package robustbase.

To provide estimates for the corresponding p-values, i.e. to compute the probability of obtaining a result at least as extreme as the one that was actually observed under the null hypothesis, we use Monte Carlo techniques where we check how often the statistic of the underlying data is more extreme than statistics obtained from simulated normal distributed data with the same (column-wise-) mean(s) and (co)variance.

Value

`statistic`	The result of the corresponding test statistic
`method`	The chosen method (univariate, angle or radius)
`p.value`	p-value

Note

These functions are use by `adtestWrapper`.

Author(s)

Karel Hron, Matthias Templ

References

Anderson, T.W. and Darling, D.A. (1952) Asymptotic theory of certain goodness-of-fit criteria based on stochastic processes. *Annals of Mathematical Statistics*, **23** 193-212.

See Also

`adtestWrapper`

Examples

```
adtest(rnorm(100))
data(machineOperators)
x <- machineOperators
adtest(ilr(x[,1:2]))
adtest(ilr(x[,1:3]))
adtest(ilr(x))
adtest(ilr(x[,1:2]), locscatt="robust")
```

`adtestWrapper` *Wrapper for Anderson-Darling tests*

Description

A set of Anderson-Darling tests (Anderson and Darling, 1952) are applied as proposed by Aitchison (Aichison, 1986).

Usage

```
adtestWrapper(x, alpha = 0.05, R = 1000, robustEst = FALSE)
```

Arguments

`x`	compositional data of class data.frame or matrix
`alpha`	significance level
`R`	Number of Monte Carlo simulations in order to provide p-values.
`robustEst`	logical

Details

First, the data is transformed using the 'ilr'-transformation. After applying this transformation

- all (D-1)-dimensional marginal, univariate distributions are tested using the univariate Anderson-Darling test for normality.

- all 0.5 (D-1)(D-2)-dimensional bivariate angle distributions are tested using the Anderson-Darling angle test for normality.

- the (D-1)-dimensional radius distribution is tested using the Anderson-Darling radius test for normality.

Value

`res`	a list including each test result
`check`	information about the rejection of the null hypothesis
`alpha`	the underlying significance level
`info`	further information which is used by the print and summary method.
`est`	"standard" for standard estimation and "robust" for robust estimation

Author(s)

Matthias Templ and Karel Hron

References

Anderson, T.W. and Darling, D.A. (1952) *Asymptotic theory of certain goodness-of-fit criteria based on stochastic processes* Annals of Mathematical Statistics, **23** 193-212.

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data* Monographs on Statistics and Applied Probability. Chapman \& Hall Ltd., London (UK). 416p.

See Also

adtest, ilr

Examples

```
data(machineOperators)
a <- adtestWrapper(machineOperators, R=50) # choose higher value of R
a
summary(a)
```

---

alr *Additive log-ratio transformation*

Description

The alr transformation moves D-part compositional data from the simplex into a (D-1)-dimensional real space.

Usage

```
alr(x, ivar=ncol(x))
```

Arguments

x	D-part compositional data
ivar	Rationing part

Details

The compositional parts are divided by the rationing part before the logarithm is taken.

Value

A list of class “alr” which includes the following content:

x.alr	the transformed data
varx	the rationing variable
ivar	the index of the rationing variable, indicating the column number of the rationing variable in the data matrix *x*
cnames	the column names of *x*

The additional information such as *cnames* or *ivar* is usefull when a back-transformation is applied on the 'same' data set.

Author(s)

Matthias Templ

References

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data* Monographs on Statistics and Applied Probability. Chapman \& Hall Ltd., London (UK). 416p.

See Also

invalr, ilr, alr

Examples

```
data(arcticLake)
x <- arcticLake
x.alr <- alr(x, 2)
y <- invalr(x.alr)
## This exactly fulfills:
invalr(alr(x, 3))
data(expenditures)
x <- expenditures
y <- invalr(alr(x, 5))
head(x)
head(y)
## --> absolute values are preserved as well.

## preserve only the ratios:
invalr(x.alr, ivar=2, useClassInfo=FALSE)
```

`arcticLake` *Artic lake sediment data*

Description

Sand, silt, clay compositions of 39 sediment samples at different water depths in an Arctic lake. This data set can be found on page 359 of the Aitchison book (see reference).

Usage

```
data(arcticLake)
```

Format

A data frame with 39 observations on the following 3 variables.

`sand` numeric vector of percentages of sand

`silt` numeric vector of percentages of silt

`clay` numeric vector of percentages of clay

Details

The rows sum up to 100, except for rounding errors. The full data set including the water depth can be found in package compositions, for example.

Source

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data* Monographs on Statistics and Applied Probability. Chapman \& Hall Ltd., London (UK). 416p.

Examples

```
data(arcticLake)
```

`clr` *Centred log-ratio transformation*

Description

The clr transformation moves D-part compositional data from the simplex into a D-dimensional real space.

Usage

```
clr(x)
```

Arguments

`x` multivariate data ideally of class data.frame or matrix

Details

Each composition is divided by the geometric mean of its parts before the logarithm is taken.

Value

The transformed data, including

`x.clr` clr transformed data

`gm` the geometric means of the original composition.

Note

The resulting transformed data set is singular by definition.

Author(s)

Matthias Templ

References

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data* Monographs on Statistics and Applied Probability. Chapman \& Hall Ltd., London (UK). 416p.

See Also

invclr, alr, ilr, invalr, invilr

Examples

```
data(expenditures)
eclr <- clr(expenditures)
inveclr <- invclr(eclr)
head(expenditures)
head(inveclr)
head(invclr(eclr$x.clr))
```

`coffee` *Coffee data*

Description

27 commercially available coffee samples of different origins.

Usage

```
data(coffee)
```

Format

A data frame with 27 observations on the following 4 variables.

`Metpyr` Hydroxy-2-propanone

`5-Met` methylpyrazine

`furfu` methylfurfural

`sort` a character vector

Details

In the original data set, 15 volatile compounds (descriptors of coffee aroma) were selected for a statistical analysis. We selected only three compounds (compositional parts) Hydroxy-2-propanone, methylpyrazine and methylfurfural to allow for a visualization in a ternary diagram.

References

M.~Korhonov\'a, K.~Hron, D.~Klimc\'ikov\'a, L.~Muller, P.~Bedn\'ar, and P.~Bart\'ak (2009) Coffee aroma - statistical analysis of compositional data. *Talanta*, 80(2): 710–715.

Examples

```
data(coffee)
```

---

constSum *Constant sum*

Description

Closes compositions to sum up to a given constant (default 1), by dividing each part of a composition by its row sum.

Usage

```
constSum(x, const=1)
```

Arguments

x	multivariate data ideally of class data.frame or matrix
const	constant, the default equals 1.

Value

The data for which the row sums are equal to `const`.

Author(s)

Matthias Templ

See Also

`clo`

Examples

```
data(expenditures)
constSum(expenditures)
constSum(expenditures, 100)
```

`daFisher` *Discriminant analysis by Fisher Rule.*

Description

Discriminant analysis by Fishers rule.

Usage

```
daFisher(x, grp, coda = TRUE, method = "classical", plotScore=FALSE)
```

Arguments

`x`	a matrix or data frame containing the explanatory variables (training set)
`grp`	grouping variable: a factor specifying the class for each observation.
`coda`	TRUE, when the underlying data are compositions.
`method`	"classical" or "robust"
`plotScore`	TRUE, if the scores should be plotted automatically.

Details

The Fisher rule leads only to linear boundaries. However, this method allows for dimension reduction and thus for a better visualization of the separation boundaries. For the Fisher discriminant rule (Fisher, 1938; Rao, 1948) the assumption of normal distribution of the groups is not explicitly required, although the method looses its optimality in case of deviations from normality.

The classical Fisher discriminant rule is invariant to ilr and clr transformations. The robust rule is invariant to ilr transformations if affine equivariant robust estimators of location and covariance are taken.

Robustification is done (method "robust") by estimating the columnwise means and the covariance by the Minimum Covariance Estimator.

Value

an object of class "daFisher" including the following elements

`B`	Between variance of the groups
`W`	Within variance of the groups
`loadings`	loadings
`coda`	coda

Author(s)

The code is was written by Peter Filzmoser. Minor modifications by Matthias Templ.

References

Filzmoser, P. and Hron, K. and Templ, M. (2009) Discriminant analysis for compositional data and robust parameter estimation. *Research Report SM-2009-3*, Vienna University of Technology, 27 pages.

Fisher, R. A. (1938) The statistical utiliziation of multiple measurements. *Annals of Eugenics*, 8:376-386.

Rao, C.R. (1948) The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society*, Series B, 10:159-203.

See Also

Linda

Examples

```
require(MASS)
x1 <- mvrnorm(20,c(0,0,0),diag(3))
x2 <- mvrnorm(30,c(3,0,0),diag(3))
x3 <- mvrnorm(40,c(0,3,0),diag(3))
X <- rbind(x1,x2,x3)
grp=c(rep(1,20),rep(2,30),rep(3,40))

#par(mfrow=c(1,2))
d1 <- daFisher(X,grp=grp,method="classical",coda=FALSE)
d2 <- daFisher(X,grp=grp,method="robust",coda=FALSE)
d2
```

---

expenditures *Household expenditures data*

Description

This data set from Aitchison (1986), p. 395, describes household expenditures (in former Hong Kong dollars) on five commundity groups.

Usage

```
data(expenditures)
```

Format

A data frame with 20 observations with the following 5 variables.

`housing` housing (including fuel and light)

`foodstuffs` foodstuffs

`alcohol` alcohol and tobacco

`other` other goods (including clothing, footwear and durable goods)

`services` services (including transport and vehicles)

Details

This data set contains household expenditures on five commodity groups of 20 single men. The variables represent housing (including fuel and light), foodstuff, alcohol and tobacco, other goods (including clothing, footwear and durable goods) and services (including transport and vehicles). Thus they represent the ratios of the men's income spent on the mentioned expenditures.

Source

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data* Monographs on Statistics and Applied Probability. Chapman \& Hall Ltd., London (UK). 416p.

Examples

```
data(expenditures)
## imputing a missing value in the data set using k-nearest neighbor imputation:
expenditures[1,3]
expenditures[1,3] <- NA
impKNNa(expenditures)$xImp[1,3]
```

`expendituresEU` *Mean consumption expenditures data.*

Description

Mean consumption expenditure of households at EU-level. The final consumption expenditure of households encompasses all domestic costs (by residents and non-residents) for individual needs.

Usage

```
data(expendituresEU)
```

Format

A data frame with 27 observations on the following 12 variables.

`Food` a numeric vector

`Alcohol` a numeric vector

`Clothing` a numeric vector

`Housing` a numeric vector

`Furnishings` a numeric vector

`Health` a numeric vector

`Transport` a numeric vector

`Communications` a numeric vector

`Recreation` a numeric vector

`Education` a numeric vector

`Restaurants` a numeric vector

`Other` a numeric vector

Source

Eurostat: http://epp.eurostat.ec.europa.eu/statistics_explained/images/c/c2/Mean_consumption_expenditure_of_households,_2005(PPS).PNG

References

Eurostat provides a website with the data:

http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Household_consumption_expenditure

Examples

```
data(expendituresEU)
```

---

haplogroups *Haplogroups data.*

Description

Distribution of European Y-chromosome DNA (Y-DNA) haplogroups by region in percentage.

Usage

```
data(haplogroups)
```

Format

A data frame with 38 observations on the following 12 variables.

`I1` pre-Germanic (Nordic)

`I2b` pre-Celto-Germanic

`I2a1` Sardinian, Basque

`I2a2` Dinaric, Danubian

`N1c1` Uralo-Finnic, Baltic, Siberian

`R1a` Balto-Slavic, Mycenaean Greek, Macedonia

`R1b` Italic, Celtic, Germanic; Hitite, Armenian

`G2a` Caucasian, Greco-Anatolien

`E1b1b` North and Eastern Afrika, Near Eastern, Balkanic

`J2` Mesopotamian, Minoan Greek, Phoenician

`J1` Semitic (Arabic, Jewish)

`T` Near-Eastern, Egyptian, Ethiopian, Arabic

Details

Human Y-chromosome DNA can be divided in genealogical groups sharing a common ancestor, called haplogroups.

Source

Eupedia: http://www.eupedia.com/europe/european_y-dna_haplogroups.shtml

Examples

```
data(haplogroups)
```

---

ilr *Isometric log-ratio transformation*

Description

An isometric log-ratio transformation with a special choice of the balances according to Hron et al. (2010).

Usage

```
ilr(x)
```

Arguments

`x`	object of class data.frame or matrix with positive entries

Details

The ilr transformation moves D-part compositional data from the simplex into a (D-1)-dimensional real space isometrically. From this choice of the balances, all the relative information of the part x_1 from the remaining parts is separated. It is useful for estimating missing values in x_1 by regression of the remaining variables.

Value

The ilr transformed data.

Author(s)

Karel Hron, Matthias Templ

References

Egozcue J.J., V. Pawlowsky-Glahn, G. Mateu-Figueras and C. Barcel'o-Vidal (2003) Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, **35**(3) 279-300. \

Hron, K. and Templ, M. and Filzmoser, P. (2010) Imputation of missing values for compositional data using classical and robust methods *Computational Statistics and Data Analysis*, vol 54 (12), pages 3095-3107.

See Also

`invilr`, `ilr`

Examples

```
require(MASS)
Sigma <- matrix(c(5.05,4.95,4.95,5.05), ncol=2, byrow=TRUE)
z <- invilr(mvrnorm(100, mu=c(0,2), Sigma=Sigma))
```

`impCoda` *Imputation of missing values in compositional data*

Description

This function offers different methods for the imputation of missing values in compositional data. Missing values are initialized with proper values. Then iterative algorithms try to find better estimations for the former missing values.

Usage

```
impCoda(x, maxit = 10, eps = 0.5, method = "ltsReg", closed = FALSE, init = "KNN", k = 5, dl = rep(0.05, nc
```

Arguments

`x`	data frame or matrix
`maxit`	maximum number of iterations
`eps`	convergence criteria
`method`	imputation method
`closed`	imputation of transformed data (using ilr transformation) or in the original space (`closed` equals TRUE)
`init`	method for initializing missing values
`k`	number of nearest neighbors (if init $==$ "KNN")
`dl`	detection limit(s), only important for the imputation of rounded zeros
`noise`	amount of adding random noise to predictors after convergency
`bruteforce`	if TRUE, imputations over dl are set to dl. If FALSE, truncated (Tobit) regression is applied.

Details

eps: The algorithm is finished as soon as the imputed values stabilize, i.e. until the sum of Aitchison distances from the present and previous iteration changes only marginally (eps).\

method: Several different methods can be chosen, such as 'ltsReg': least trimmed squares regression is used within the iterative procedure. 'lm': least squares regression is used within the iterative procedure. 'classical': principal component analysis is used within the iterative procedure. 'ltsReg2': least trimmed squares regression is used within the iterative procedure. The imputated values are perturbed in the direction of the predictor by values drawn form a normal distribution with mean and standard deviation related to the corresponding residuals and multiplied by `noise`.

method 'roundedZero' is experimental. It imputes rounded zeros within our iterative framework.

Value

`xOrig`	Original data frame or matrix
`xImp`	Imputed data
`criteria`	Sum of the Aitchison distances from the present and previous iteration
`iter`	Number of iterations
`maxit`	Maximum number of iterations
`w`	Amount of imputed values
`wind`	Index of the missing values in the data

Author(s)

Matthias Templ, Karel Hron

References

Hron, K. and Templ, M. and Filzmoser, P. (2010) Imputation of missing values for compositional data using classical and robust methods *Computational Statistics and Data Analysis*, vol 54 (12), pages 3095-3107.

See Also

`impKNNa`, `ilr`

Examples

```
data(expenditures)
x <- expenditures
x[1,3]
x[1,3] <- NA
xi <- impCoda(x)$xImp
xi[1,3]
s1 <- sum(x[1,-3])
impS <- sum(xi[1,-3])
xi[,3] * s1/impS
```

impKNNa — *Imputation of missing values in compositional data using knn methods*

Description

This function offers several k-nearest neighbor methods for the imputation of missing values in compositional data.

Usage

```
impKNNa(x, method = "knn", k = 3, metric = "Aitchison", agg = "median",
        primitive = FALSE, normknn = TRUE, das = FALSE, adj="median")
```

Arguments

`x`	data frame or matrix
`method`	method (at the moment, only "knn" can be used)
`k`	number of nearest neighbors chosen for imputation
`metric`	"Aichison" or "Euclidean"
`agg`	"median" or "mean", for the aggregation of the nearest neighbors
`primitive`	if TRUE, a more enhanced search for the k-nearest neighbors is obtained (see details)
`normknn`	An adjustment of the imputed values is performed if TRUE
`das`	depricated. if TRUE, the definition of the Aitchison distance, based on simple logratios of the compositional part, is used (Aitchison, 2000) to calculate distances between observations. if FALSE, a version using the clr transformation is used.
`adj`	either 'median' (default) or 'sum' can be chosen for the adjustment of the nearest neighbors, see Hron et al., 2010.

Details

The Aitchison `metric` should be chosen when dealing with compositional data, the Euclidean `metric` otherwise.

If `primitive` $==$ FALSE, a sequential search for the k-nearest neighbors is applied for every missing value where all information corresponding to the non-missing cells plus the information in the variable to be imputed plus some additional information is available. If `primitive` $==$ TRUE, a search of the k-nearest neighbors among observations is applied where in addition to the variable to be imputed any further cells are non-missing.

If `normknn` is TRUE (prefered option) the imputed cells from a nearest neighbor method are adjusted with special adjustment factors (more details can be found online (see the references)).

Value

xOrig	Original data frame or matrix
xImp	Imputed data
w	Amount of imputed values
wind	Index of the missing values in the data
metric	Metric used

Author(s)

Matthias Templ

References

Aitchison, J. and Barcelo-Vidal, C. and Martin-Fernandez, J.A. and Pawlowsky-Glahn, V. (2000) Logratio analysis and compositional distance, Mathematical Geology 32(3):271-275.

Hron, K. and Templ, M. and Filzmoser, P. (2010) Imputation of missing values for compositional data using classical and robust methods *Computational Statistics and Data Analysis*, vol 54 (12), pages 3095-3107.

See Also

impCoda

Examples

```
data(expenditures)
x <- expenditures
x[1,3]
x[1,3] <- NA
xi <- impKNNa(x)$xImp
xi[1,3]
```

---

impRZalr *alr EM-based Imputation for Rounded Zeros*

Description

A modified EM alr-algorithm for replacing rounded zeros in compositional data sets.

Usage

```
impRZalr(x, pos = ncol(x), dl = rep(0.05, ncol(x) - 1), eps = 1e-04, maxit = 50, bruteforce=FALSE, method
```

Arguments

`x`	Compositional data
`pos`	Position of the rationing variable for alr transformation
`dl`	Detection limit for each part
`eps`	convergence criteria
`maxit`	maximum number of iterations
`bruteforce`	if TRUE, imputations over dl are set to dl. If FALSE, truncated (Tobit) regression is applied.
`method`	either "lm" (default) or "MM"
`step`	if TRUE, a stepwise (AIC) procedure is applied when fitting models

Details

Statistical analysis of compositional data including zeros runs into problems, because log-ratios cannot be applied. Usually, rounded zeros are considerer as missing not at random missing values. The algorithm first applies an additive log-ratio transformation to the compositions. Then the rounded zeros are imputed using a modified EM algorithm.

Value

`xOrig`	Original data frame or matrix
`xImp`	Imputed data
`wind`	Index of the missing values in the data
`iter`	Number of iterations
`eps`	eps

Author(s)

Matthias Templ and Karel Hron

See Also

`impRZilr`

Examples

```
data(arcticLake)
x <- arcticLake
## generate rounded zeros artificially:
x[x[,1] < 5, 1] <- 0
x[x[,2] < 47, 2] <- 0
xia <- impRZalr(x, pos=3, dl=c(5,47), eps=0.05)
xia$xImp
```

impRZilr — *EM-based replacement of rounded zeros in compositional data*

Description

Parametric replacement of rounded zeros for compositional data using classical and robust methods based on ilr-transformations with special choice of balances.

Usage

```
impRZilr(x, maxit = 10, eps = 0.1, method = "roundedZero", dl = rep(0.05, ncol(x)), bruteforce = FALSE)
```

Arguments

`x`	data.frame or matrix
`maxit`	maximum number of iterations
`eps`	convergency criteria
`method`	either “roundedZeor” or “roundedZeroRobust”
`dl`	Detection limit for each variable
`bruteforce`	sets imputed values above the detection limit to the detection limit. Replacement above the detection limit are only exeptionally occur due to numerical instabilities. The default is FALSE!

Details

Statistical analysis of compositional data including zeros runs into problems, because log-ratios cannot be applied. Usually, rounded zeros are considerer as missing not at random missing values.

The algorithm iteratively imputes parts with rounded zeros whereas in each step (1) an specific ilr transformation is applied (2) tobit regression is applied (3) the rounded zeros are replaced by the expected values (4) the corresponding inverse ilr transformation is applied. After all parts are imputed, the algorithm starts again until the imputations do not change.

Value

`xOrig`	Original data frame or matrix
`xImp`	Imputed data
`wind`	Index of the missing values in the data
`iter`	Number of iterations
`eps`	eps

Author(s)

Matthias Templ and Peter Filzmoser

See Also

impRZalr

Examples

```
data(arcticLake)
x <- arcticLake
## generate rounded zeros artificially:
x[x[,1] < 5, 1] <- 0
x[x[,2] < 47, 2] <- 0
xia <- impRZilr(x, dl=c(5,47,9999), eps=0.05)
xia$xImp
```

---

invalr *Additive logistic transformaton*

Description

Inverse additive log-ratio transformation, often called additive logistic transformation.

Usage

```
invalr(x, cnames = NULL, ivar = NULL, useClassInfo = TRUE)
```

Arguments

x	data set, object of class “alr”, “matrix” or “data.frame”
cnames	column names. If the object is of class “alr” the column names are chosen from therein.
ivar	index of the rationing part. If the object is of class “alr” the column names are chosen from therein. If not and ivar is not provided by the user, it is assumed that the rationing part was the last column of the data in the simplex.
useClassInfo	if FALSE, the class information of object x is not used.

Details

The function allows also to preserve absolute values when class info is provided. Otherwise only the relative information is preserved.

Value

the transformed data matrix

Author(s)

Matthias Templ

References

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data* Monographs on Statistics and Applied Probability. Chapman \& Hall Ltd., London (UK). 416p.

See Also

invilr, invclr, clr, alr, ilrInv

Examples

```
data(arcticLake)
x <- arcticLake
x.alr <- alr(x, 2)
y <- invalr(x.alr)
## This exactly fulfills:
invalr(alr(x, 3))
data(expenditures)
x <- expenditures
y <- invalr(alr(x, 5))
head(x)
head(y)
## --> absolute values are preserved as well.

## preserve only the ratios:
invalr(x.alr, ivar=2, useClassInfo=FALSE)
```

---

invclr *Inverse centred log-ratio transformation*

Description

Applies the inverse centred log-ratio transformation.

Usage

```
invclr(x, useClassInfo = TRUE)
```

Arguments

x	an object of class “clr”, “data.frame” or “matrix”
useClassInfo	if the object is of class clr, the useClassInfo is used to determine if the class information should be used. If yes, also absolute values may be preserved.

Value

the transformed data set.

Author(s)

Matthias Templ

References

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data* Monographs on Statistics and Applied Probability. Chapman \& Hall Ltd., London (UK). 416p.

See Also

`clr`, `alr`, `ilr`, `invalr`, `invilr`

Examples

```
data(expenditures)
eclr <- clr(expenditures)
inveclr <- invclr(eclr)
head(expenditures)
head(inveclr)
head(invclr(eclr$x.clr))
```

---

invilr — *Inverse isometric log-ratio transformation*

Description

The inverse transformation of 'ilr()'.

Usage

```
invilr(x.ilr)
```

Arguments

`x.ilr`	data frame or matrix

Details

For details on the choice of the balances, please, see at the research report for which the link is given below.

Value

The transformed data.

Author(s)

Karel Hron

References

Egozcue J.J., V. Pawlowsky-Glahn, G. Mateu-Figueras and C. Barcel'o-Vidal (2003) Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, **35**(3) 279-300 \

Hron, K. and Templ, M. and Filzmoser, P. (2010) Imputation of missing values for compositional data using classical and robust methods *Computational Statistics and Data Analysis*, vol 54 (12), pages 3095-3107.

See Also

ilr

Examples

```
require(MASS)
Sigma <- matrix(c(5.05,4.95,4.95,5.05), ncol=2, byrow=TRUE)
set.seed(123)
z <- mvrnorm(100, mu=c(0,2), Sigma=Sigma)
x <- invilr(z)
head(x)
```

---

lmCoDaX *Robust regression of non-compositional response on compositional predictors*

Description

Delivers appropriate inference for regression of y on a compositional matrix X.

Usage

```
lmCoDaX(y, X, method = "robust")
```

Arguments

`y`	The response which should be non-compositional
`X`	The compositional predictors as a matrix, data.frame or numeric vector
`method`	if robust, lts-regression is applied while with method equals “classical” the conventional least squares estimates are applied.

Details

Compositional explanatory variables should not be directly used in a linear regression model because any inference statistic can become misleading. While various approaches for this problem were proposed, here an approach based on the isometric logratio (ilr) transformation is used.

Value

An object of class ‘lts’ or ‘lm’ and two summary objects.

Author(s)

Peter Filzmoser

References

Filzmoser, P., Hron, K., Thompsonc, K. (2012) Linear regression with compositional explanatory variables. *Journal of Applied Statistics*, accepted for publication.

See Also

lm

Examples

```
## coming soon
```

---

machineOperators — *Machine operators data set*

Description

The data set from Aitchison (1986), p. 382, contains compositions of eight-hour shifts of 27 machine operators. The parts represent proportions of shifts in each activity: high-quality production, low-quality production, machine setting and machine repair.

Usage

```
data(machineOperators)
```

Format

A data frame with 27 observations on the following 4 variables.

`hqproduction` high-quality production

`lqproduction` low-quality production

`setting` machine settings

`repair` machine repair

References

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data* Monographs on Statistics and Applied Probability. Chapman \& Hall Ltd., London (UK). 416p.

Examples

```
## maybe str(machineOperators) ; plot(machineOperators) ...
```

outCoDa *Outlier detection for compositional data*

Description

Outlier detection for compositional data using standard and robust statistical methods.

Usage

```
outCoDa(x, quantile = 0.975, method = "robust", h = 1/2)
```

Arguments

`x`	compositional data
`quantile`	quantile, corresponding to a significance level, is used as a cut-off value for outlier identification: observations with larger (squared) robust Mahalanobis distance are considered as potential outliers.
`method`	either "robust" (default) or "standard"
`h`	the size of the subsets for the robust covariance estimation according the MCD-estimator for which the determinant is minimized (the default is (n+p+1)/2).

Details

The outlier detection procedure is based on (robust) Mahalanobis distances after a isometric logratio transformation of the data. Observations with squared Mahalanobis distance greater equal a certain quantile of the Chi-squared distribution are marked as outliers.

If method "robust" is chosen, the outlier detection is based on the homogeneous majority of the compositional data set. If method "standard" is used, standard measures of location and scatter are applied during the outlier detection procedure.

Value

`mahalDist`	resulting Mahalanobis distance
`limit`	quantile of the Chi-squared distribution
`outlierIndex`	logical vector indicating outliers and non-outliers
`method`	method used

Note

It is highly recommended to use the robust version of the procedure.

Author(s)

Matthias Templ, Karel Hron

References

Egozcue J.J., V. Pawlowsky-Glahn, G. Mateu-Figueras and C. Barcel'o-Vidal (2003) Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, **35**(3) 279-300. \

Filzmoser, P., and Hron, K. (2008) Outlier detection for compositional data using robust methods. *Math. Geosciences*, **40** 233-248.\

Rousseeuw, P.J., Van Driessen, K. (1999) A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, **41** 212-223.

See Also

`ilr`

Examples

```
data(expenditures)
oD <- outCoDa(expenditures)
oD
```

`pcaCoDa` *Robust principal component analysis for compositional data*

Description

This function applies robust principal component analysis for compositional data.

Usage

```
pcaCoDa(x, method = "robust")
```

Arguments

`x`	compositional data
`method`	either "robust" (default) or "standard"

Details

The compositional data set is transformed using the ilr tranformation. Afterwards, robust principal component analysis is performed. Resulting loadings and scores are back-transformed to the clr space where the compositional biplot can be shown.

Value

`scores`	scores in clr space
`loadings`	loadings in clr space
`eigenvalues`	eigenvalues of the clr covariance matrix
`method`	method
`princompOutputClr`	output of `princomp` needed in `plot.pcaCoDa`

Author(s)

K. Hron, P. Filzmoser, M. Templ

References

Filzmoser, P., Hron, K., Reimann, C. (2009) Principal Component Analysis for Compositional Data with Outliers. *Environmetrics*, **20**, 621-632.

See Also

print.pcaCoDa, plot.pcaCoDa

Examples

```
data(expenditures)
p1 <- pcaCoDa(expenditures)
p1
plot(p1)
```

---

pfa *Factor analysis for compositional data*

Description

Computes the principal factor analysis of the input data which are transformed and centered first.

Usage

```
pfa(x, factors, data = NULL, covmat = NULL, n.obs = NA,
subset, na.action, start = NULL,
scores = c("none", "regression", "Bartlett"),
rotation = "varimax", maxiter = 5, control = NULL, ...)
```

Arguments

x	(robustly) scaled input data
factors	number of factors
data	default value is NULL
covmat	(robustly) computed covariance or correlation matrix
n.obs	number of observations
subset	if a subset is used
na.action	what to do with NA values
start	starting values
scores	which method should be used to calculate the scores
rotation	if a rotation should be made

`maxiter`	maximum number of iterations
`control`	default value is NULL
`...`	arguments for creating a list

Details

The main difference to usual implementations is that uniquenesses are nor longer of diagonal form. This kind of factor analysis is designed for centered log-ratio transformed compositional data. However, if the covariance is not specified, the covariance is estimated from isometric log-ratio transformed data internally, but the data used for factor analysis are backtransformed to the clr space (see Filzmoser et al., 2009).

Value

`loadings`	A matrix of loadings, one column for each factor. The factors are ordered in decreasing order of sums of squares of loadings.
`uniquness`	uniquness
`correlation`	correlation matrix
`criteria`	The results of the optimization: the value of the negativ log-likelihood and information of the iterations used.
`factors`	the factors
`dof`	degrees of freedom
`method`	"principal"
`n.obs`	number of observations if available, or NA
`call`	The matched call.
`STATISTIC, PVAL`	The significance-test statistic and p-value, if they can be computed

Author(s)

Peter Filzmoser, Karel Hron, Matthias Templ

References

C. Reimann, P. Filzmoser, R.G. Garrett, and R. Dutter (2008): Statistical Data Analysis Explained. *Applied Environmental Statistics with R*. John Wiley and Sons, Chichester, 2008.

P. Filzmoser, K. Hron, C. Reimann, R. Garrett (2009): Robust Factor Analysis for Compositional Data. *Computers and Geosciences*, **35** (9), 1854–1861.

Examples

```
data(expenditures)
x <- expenditures
res0 <- pfa(x, factors=1, covmat="cov")

## the following produce always the same result:
res1 <- pfa(x, factors=1, covmat="covMcd")
```

```
res2 <- pfa(x, factors=1, covmat=covMcd(ilr(x))$cov)
res3 <- pfa(x, factors=1, covmat=covMcd(ilr(x)))
```

phd *PhD Students in the EU*

Description

PhD students in Europe based on the standard klassification system splitted by different kind of studies (given as percentages).

Usage

```
data(phd)
```

Format

The format is: num [1:33, 1:8] 516.5 7.5 5.2 22.6 4.8 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:33] "EU" "Belgien" "Bulgarien" "Tschech.Rep." ... ..$: chr [1:8] "Gesamtzahl der Doktoranden (in 1 000)" "maennlich" "weiblich" "Naturwissen-schaften, Mathematik, Informatik u. Ingenieurwesen" ...

Details

Due to unknown reasons the rowSums of the percentages is not always 100.

Source

http://epp.eurostat.ec.europa.eu/cache/ITY_PUBLIC/1-18092009-AP/DE/1-18092009-AP-DE.PDF

Examples

```
data(phd)
phdImputed <- impCoda(phd)$xOrig
```

plot.imp *Plot method for objects of class imp*

Description

This function provides several diagnostic plots for the imputed data set in order to see how the imputated values are distributed in comparison with the original data values.

Usage

```
## S3 method for class 'imp'
plot(x, ..., which = 1, ord = 1:ncol(x), colcomb = "missnonmiss", plotvars = NULL,
  col = c("skyblue", "red"), alpha = NULL, lty = par("lty"), xaxt = "s", xaxlabels = NULL,
   las = 3, interactive = TRUE, pch = c(1, 3), smooth = FALSE, reg.line = FALSE,
  legend.plot = FALSE, ask = prod(par("mfcol")) < length(which) && dev.interactive(),
   center = FALSE, scale = FALSE, id = FALSE, seg.l = 0.02, seg1 = TRUE)
```

Arguments

`x`	object of class 'imp'
`...`	other parameters to be passed through to plotting functions.
`which`	if a subset of the plots is required, specify a subset of the numbers 1:3.
`ord`	determines the ordering of the variables
`colcomb`	if colcomb="missnonmiss", observations with missings in any variable are highlighted. Otherwise, observations with missings in any of the variables specified by colcomb are highlighted in the parallel coordinate plot.
`plotvars`	Parameter for the parallel coordinate plot. A vector giving the variables to be plotted. If NULL (the default), all variables are plotted.
`col`	a vector of length two giving the colors to be used in the plot. The second color will be used for highlighting.
`alpha`	a numeric value between 0 and 1 giving the level of transparency of the colors, or NULL. This can be used to prevent overplotting.
`lty`	a vector of length two giving the line types. The second line type will be used for the highlighted observations. If a single value is supplied, it will be used for both non-highlighted and highlighted observations.
`xaxt`	the x-axis type (see `par`).
`xaxlabels`	a character vector containing the labels for the x-axis. If NULL, the column names of x will be used.
`las`	the style of axis labels (see `par`).
`interactive`	a logical indicating whether the variables to be used for highlighting can be selected interactively (see 'Details').
`pch`	a vector of length two giving the symbol of the plotting points. The symbol will be used for the highlighted observations. If a single value is supplied, it will be used for both non-highlighted and highlighted observations.

`smooth`	if TRUE a lowess smooth is plotted in each off-diagonal panel of the multiple scatterplot. Further detail can be found in package car.
`reg.line`	if not FALSE a line is plotted using the function given by this argument; e.g., using rlm in package MASS plots a robust-regression line within the multiple scatterplot.
`legend.plot`	if TRUE then a legend for the groups is plotted in the bottom-right cell of the multiple scatterplot.
`ask`	logical; if TRUE, the user is asked before each plot, see `par`(ask=.).
`center`	logical, indicates if the data should be centered prior plotting the ternary plot.
`scale`	logical, indicates if the data should be centered prior plotting the ternary plot.
`id`	reads the position of the graphics pointer when the (first) mouse button is pressed and returns the corresponding index of the observation. (only used by the ternary plot)
`seg.l`	length of the plotting symbol (spikes) for the ternary plot.
`seg1`	if TRUE, the spikes of the plotting symbol are justified.

Details

The first plot (which == 1) is a multiple scatterplot where for the imputed values another plot symbol and color is used in order to highlight them.

Plot 2 is a parallel coordinate plot in which imputed values in certain variables are highlighted. In parallel coordinate plots, the variables are represented by parallel axes. Each observation of the scaled data is shown as a line. If interactive is TRUE, the variables to be used for highlighting can be selected interactively. Observations which includes imputed values in any of the selected variables will be highlighted. A variable can be added to the selection by clicking on a coordinate axis. If a variable is already selected, clicking on its coordinate axis will remove it from the selection. Clicking anywhere outside the plot region quits the interactive session.

Plot 3 shows a ternary diagram in which imputed values are highlighted, i.e. those spikes of the chosen plotting symbol are colored in red for which of the values are missing in the unimputed data set.

Value

None (invisible NULL).

Author(s)

Matthias Templ

References

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data* Monographs on Statistics and Applied Probability. Chapman \& Hall Ltd., London (UK). 416p.

Wegman, E. J. (1990) *Hyperdimensional data analysis using parallel coordinates* Journal of the American Statistical Association 85, 664–675.

See Also

impCoda, impKNNa, \ scatterplot.matrix

Examples

```
data(expenditures)
expenditures[1,3]
expenditures[1,3] <- NA
xi <- impKNNa(expenditures)
xi
summary(xi)
plot(xi, which=1)
plot(xi, which=2)
plot(xi, which=3)
plot(xi, which=3, seg1=FALSE)
```

---

plot.outCoDa *plot method for outCoDa objects*

Description

Plots the Mahalanobis distance.

Usage

```
## S3 method for class 'outCoDa'
plot(x, y, ...)
```

Arguments

x	object from class 'outCoDa'
y	...
...	...

Details

The dashed line indicates the (1 - alpha) quantile of the Chi-squared distribution. Observations with Mahalanobis distance greater than this quantile could be considered as compositional outliers.

Author(s)

Matthias Templ

References

Filzmoser, P., and Hron, K. (2008) Outlier detection for compositional data using robust methods. *Math. Geosciences*, **40** 233-248.

See Also

outCoDa

Examples

```
data(expenditures)
oD <- outCoDa(expenditures)
oD
plot(oD)
```

`plot.pcaCoDa` *Plot method*

Description

Provides robust compositional biplots.

Usage

```
## S3 method for class 'pcaCoDa'
plot(x, y, ...)
```

Arguments

x	object of class 'pcaCoDa'
y	...
...	...

Details

The robust compositional biplot according to Aitchison and Greenacre (2002), computed from resulting (robust) loadings and scores, is performed.

Value

The robust compositional biplot.

Author(s)

M. Templ, K. Hron

References

Aitchison, J. and Greenacre, M. (2002). Biplots of compositional data. *Applied Statistics*, **51**, 375-392. \

Filzmoser, P., Hron, K., Reimann, C. (2009) Principal Component Analysis for Compositional Data with Outliers. *Environmetrics*, **20** (6), 621–632.

See Also

pcaCoDa

Examples

```
data(expenditures)
p1 <- pcaCoDa(expenditures)
p1
plot(p1)
```

---

print.adtestWrapper *print method for objects of class adtestWrapper*

Description

Provides a short print output as shown in the examples.

Usage

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

Arguments

x	object of class 'adtestWrapper'
...	...

Details

Have a look at the example, it's self-explaining.

Value

nothing

Author(s)

Matthias Templ and Karel Hron

See Also

adtestWrapper, summary.adtestWrapper

Examples

```
data(machineOperators)
a <- adtestWrapper(machineOperators, R=50) # choose higher value of R
a
summary(a)
```

print.daFisher *print method for objects of class daFisher*

Description

Provides a print output of objects from class "daFisher" as shown in the examples.

Usage

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

Arguments

x	object of class 'daFisher'
...	...

Details

Have a look at the example, it's self-explaining.

Value

nothing

Author(s)

Matthias Templ

See Also

daFisher

Examples

```
require(MASS)
x1 <- mvrnorm(20,c(0,0,0),diag(3))
x2 <- mvrnorm(30,c(3,0,0),diag(3))
x3 <- mvrnorm(40,c(0,3,0),diag(3))
X <- rbind(x1,x2,x3)
grp=c(rep(1,20),rep(2,30),rep(3,40))

d1 <- daFisher(X,grp=grp,method="classical",coda=FALSE)
d2 <- daFisher(X,grp=grp,method="robust",coda=FALSE)
d2
```

`print.imp` *Print method for objects of class imp*

Description

The function returns a few information about how many missing values are imputed and possible other information about the amount of iterations, for example.

Usage

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

Arguments

`x`	an object of class 'imp'
`...`	additional arguments passed trough

Value

None (invisible NULL).

Author(s)

Matthias Templ

See Also

`impCoda`, `impKNNa`

Examples

```
data(expenditures)
expenditures[1,3]
expenditures[1,3] <- NA
xi <- impCoda(expenditures)
xi
summary(xi)
plot(xi, which=1:2)
```

print.outCoDa *print method for outCoDa objects*

Description

Gives a short information of the amount of outliers in objects of class 'outCoDa'.

Usage

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

Arguments

x	object of class 'ourCoDa'
...	...

Author(s)

Matthias Templ, Karel Hron

See Also

outCoDa

Examples

```
data(expenditures)
oD <- outCoDa(expenditures)
oD
```

print.pcaCoDa *Print method for pcaCoDa objects*

Description

Print method for objects of class 'pcaCoDa'.

Usage

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

Arguments

x	object of class 'pcaCoDa'
...	...

Value

Prints the (cummulative) percentages of explained variability for clr transformed data by principal component analysis.

Author(s)

M. Templ, K. Hron

See Also

`pcaCoDa`, `plot.pcaCoDa`

Examples

```
data(expenditures)
p1 <- pcaCoDa(expenditures)
p1
plot(p1)
```

---

robGUI *A lightweighted graphical user interface*

Description

A lightweighted graphical user interface for the robCompositions package written in Gkt2. Before calling the GUI, RGtk2 have to be installed.

Usage

```
robGUI()
```

Details

Already some of the methods from the robCompositions package can be applied by this GUI. The application is straighforward and worth to mention.

Value

The GUI.

Note

The next version of the GUI may include tools for reproducibility (save and load/run script files).

Author(s)

Jiri Eichler

Examples

```
## Not run:
## to open the GUI:
## robGUI()
## End(Not run)
```

`robVariation` *Robust variation matrix*

Description

Estimates the variation matrix with robust methods.

Usage

```
robVariation(x, robust=TRUE)
```

Arguments

`x`	data frame or matrix with positive entries
`robust`	if FALSE, standard measures are used.

Details

The variation matrix is estimated for a given compositional data set. Instead of using the classical standard deviations the `mad` is used when parameter robust is set to TRUE.

Value

The (robust) variation matrix.

Author(s)

Matthias Templ

References

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data* Monographs on Statistics and Applied Probability. Chapman \& Hall Ltd., London (UK). 416p.

See Also

`variation`

Examples

```
data(expenditures)
robVariation(expenditures)
robVariation(expenditures, robust=FALSE)
```

skyeLavas *Aphyric skye lavas data*

Description

AFM compositions of 23 aphyric Skye lavas. This data set can be found on page 360 of the Aitchison book (see reference).

Usage

```
data(skyeLavas)
```

Format

A data frame with 23 observations on the following 3 variables.

`sodium-potassium` a numeric vector of percentages of Na2O+K2O

`iron` a numeric vector of percentages of Fe2O3

`magnesium` a numeric vector of percentages of MgO

Source

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data* Monographs on Statistics and Applied Probability. Chapman \& Hall Ltd., London (UK). 416p.

Examples

```
data(skyeLavas)
```

summary.adtestWrapper *summary method for objects of class adtestWrapper*

Description

Provides a summary as shown in the examples.

Usage

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

Arguments

`object`	object of class ‘adtestWrapper’
`...`	additional arguments passed through

Details

A similar output is proposed by (Pawlowsky-Glahn, et al. (2008). In addition to that, p-values are provided.

Value

a data frame including an information about the ilr-variables used (first column), the underlying test (second column), the test statistics (third column), the corresponding estimated p-values (fourth column) and an information about the rejection of the null hypothesis (last column).

Author(s)

Matthias Templ and Karel Hron

References

Pawlowsky-Glahn, V., Egozcue, J.J. and Tolosana-Delgado, R. (2008), *Lecture Notes on Compositional Data Analysis* Universitat de Girona, http://dugi-doc.udg.edu/bitstream/10256/297/1/CoDa-book.pdf

See Also

adtestWrapper

Examples

```
data(machineOperators)
a <- adtestWrapper(machineOperators, R=50) # choose higher value of R
a
summary(a)
```

---

summary.imp *Summary method for objects of class imp*

Description

A short comparison of the original data and the imputed data is given.

Usage

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

Arguments

object	an object of class 'imp'
...	additional arguments passed trough

Details

Note that this function will be enhanced with more sophisticated methods in future versions of the package. It is very rudimental in its present form.

Value

None (invisible NULL).

Author(s)

Matthias Templ

See Also

impCoda, impKNNa

Examples

```
data(expenditures)
expenditures[1,3]
expenditures[1,3] <- NA
xi <- impKNNa(expenditures)
xi
summary(xi)
# plot(xi, which=1:2)
```

---

ternaryDiag *Ternary diagram*

Description

This plot shows the relative proportions of three variables (compositional parts) in one diagramm. Before plotting, the data are scaled.

Usage

```
ternaryDiag(x, name = colnames(x), grid = TRUE, gridCol=grey(0.6), mcex = 1.2, line = "none", robust = TR
```

Arguments

x	matrix or data.frame with 3 columns
name	names of the variables
grid	if TRUE a grid is plotted additionally in the ternary diagram
gridCol	color for the grid lines
mcex	label size
line	may be set to “none”, “pca”, “regression”, “regressionconf”, “regressionpred”, “ellipse”, “lda”

`robust`	if line equals TRUE, it dedicates if a robust estimation is applied or not.
`group`	if line equals "da", it determines the grouping variable
`tol`	if line equals "ellipse", it determines the parameter for the tolerance ellipse
`...`	further parameters, see, e.g., `par()`

Details

The relative proportions of each variable are plotted.

Author(s)

Peter Filzmoser <<P.Filzmoser@tuwien.ac.at>> http://www.statistik.tuwien.ac.at/public/filz/, Matthias Templ

References

C. Reimann, P. Filzmoser, R.G. Garrett, and R. Dutter: Statistical Data Analysis Explained. Applied Environmental Statistics with R. John Wiley and Sons, Chichester, 2008.

See Also

`ternary`

Examples

```
data(arcticLake)
ternaryDiag(arcticLake)

data(coffee)
x <- coffee[,1:3]
grp <- as.integer(factor(coffee[,4]))
ternaryDiag(x, col=grp, pch=grp)
ternaryDiag(x, grid=FALSE, col=grp, pch=grp)
legend("topright", legend=unique(coffee[,4]), pch=1:2, col=1:2)

ternaryDiag(x, grid=FALSE, col=grp, pch=grp, line="ellipse", tol=c(0.975,0.9), lty=2)
ternaryDiag(x, grid=FALSE, line="pca")
ternaryDiag(x, grid=FALSE, col=grp, pch=grp, line="pca", lty=2, lwd=2)
```

`ternaryDiagAbline` *Adds a line to a ternary diagram.*

Description

A low-level plot function which adds a line to a high-level ternary diagram.

Usage

```
ternaryDiagAbline(x, ...)
```

Arguments

`x`	Two-dimensional data set in isometric log-ratio transformed space.
`...`	Additional graphical parameters passed through.

Details

This is a small utility function which helps to add a line in a ternary plot from two given points in an isometric transformed space.

Value

no values are returned.

Author(s)

Matthias Templ

See Also

`ternaryDiag`

Examples

```
data(coffee)
x <- coffee[,1:3]
ternaryDiag(x, grid=FALSE)
ternaryDiagAbline(data.frame(z1=c(0.01,0.5), z2=c(0.4,0.8)), col="red")
```

`ternaryDiagEllipse` *Adds tolerance ellipses to a ternary diagram.*

Description

Low-level plot function which add tolerance ellipses to a high-level plot of a ternary diagram.

Usage

```
ternaryDiagEllipse(x, tolerance = c(0.9, 0.95, 0.975), locscatt = "MCD", ...)
```

Arguments

`x`	Three-part composition. Object of class “matrix” or “data.frame”.
`tolerance`	Determines the amount of observations with Mahalanobis distance larger than the drawn ellipse, scaled to one.
`locscatt`	Method for estimating the mean and covariance.
`...`	Additional arguments passed trough.

Value

no values are returned.

Author(s)

Peter Filzmoser, Matthias Templ

See Also

ternaryDiag

Examples

```
data(coffee)
x <- coffee[,1:3]
ternaryDiag(x, grid=FALSE)
ternaryDiagEllipse(x)
## or directly:
ternaryDiag(x, grid=FALSE, line="ellipse")
```

---

ternaryDiagPoints *Add points or lines to a given ternary diagram.*

Description

Low-level plot function to add points or lines to a ternary high-level plot.

Usage

```
ternaryDiagPoints(x, ...)
ternaryDiagLines(x, ...)
```

Arguments

x	Three-dimensional composition given as an object of class "matrix" or "data.frame".
...	Additional graphical parameters passed through.

Value

no values are returned.

Author(s)

Matthias Templ

References

C. Reimann, P. Filzmoser, R.G. Garrett, and R. Dutter: Statistical Data Analysis Explained. Applied Environmental Statistics with R. John Wiley and Sons, Chichester, 2008.

See Also

ternaryDiag

Examples

```
data(coffee)
x <- coffee[,1:3]
ternaryDiag(x, grid=FALSE)
ternaryDiagPoints(x+1, col="red", pch=2)
```

Index