Package ‘robCompositions’

May 9, 2026

Type Package

Title Compositional Data Analysis

Version 2.4.2

Date 2025-08-06

Depends R (>= 3.5.0), ggplot2, pls, data.table

LinkingTo Rcpp, RcppEigen

Imports cvTools, fda, rrcov, cluster, dplyr, magrittr, GGally, ggfortify, kernlab, MASS, mclust, tidyr, robustbase, robustHD, sparsepca, splines, VIM, zCompositions, reshape2, Rcpp

Suggests e1071, fpc, knitr, testthat

VignetteBuilder knitr

Maintainer Matthias Templ <matthias.templ@gmail.com>

Description Methods for analysis of compositional data including robust methods (<doi:10.1007/978-3-319-96422-5>), imputation of missing values (<doi:10.1016/j.csda.2009.11.023>), methods to replace rounded zeros (<doi:10.1080/02664763.2017.1410524>, <doi:10.1016/j.chemolab.2016.04.011>, <doi:10.1016/j.csda.2012.02.012>), count zeros (<doi:10.1177/1471082X14535524>), methods to deal with essential zeros (<doi:10.1080/02664763.2016.1182135>), (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, functional data analysis (<doi:10.1016/j.csda.2015.07.007>) and p-splines (<doi:10.1016/j.csda.2015.07.007>), contingency (<doi:10.1080/03610926.2013.824980>) and compositional tables (<doi:10.1111/sjos.12326>, <doi:10.1111/sjos.12223>, <doi:10.1080/02664763.2013.856871>) and (robust) Anderson-Darling normality tests for compositional data as well as popular log-ratio transformations (addLR, cenLR, isomLR, 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

LazyData true

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation yes

Author Matthias Templ [aut, cre] (ORCID:
<https://orcid.org/0000-0002-8638-5276>),
Karel Hron [aut] (ORCID: <https://orcid.org/0000-0002-1847-6598>),
Peter Filzmoser [aut] (ORCID: <https://orcid.org/0000-0002-8014-4682>),
Kamila Facevicova [ctb],
Petra Kynclova [ctb],
Jan Walach [ctb],
Veronika Pintar [ctb],
Jiajia Chen [ctb],
Dominika Miksova [ctb],
Bernhard Meindl [ctb],
Alessandra Menafoglio [ctb] (ORCID:
<https://orcid.org/0000-0003-0682-6412>),
Alessia Di Blasi [ctb],
Federico Pavone [ctb],
Nikola Stefelova [ctb],
Gianluca Zeni [ctb],
Roman Wiederkehr [ctb]

Repository CRAN

Date/Publication 2025-08-22 15:20:02 UTC

Contents

`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).

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 and 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)
## Not run: 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 <- addLR(x, 2)
y <- addLRinv(x.alr)
addLRinv(addLR(x, 3))
data(expenditures)
x <- expenditures
y <- addLRinv(addLR(x, 5))
head(x)
head(y)
addLRinv(x.alr, ivar=2, useClassInfo=FALSE)

data(expenditures)
eclr <- cenLR(expenditures)
inveclr <- cenLRinv(eclr)
head(expenditures)
head(inveclr)
head(cenLRinv(eclr$x.clr))

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

---

addLR *Additive logratio coordinates*

Description

The additive logratio coordinates map D-part compositional data from the simplex into a (D-1)-dimensional real space.

Usage

```
addLR(x, ivar = ncol(x), base = exp(1))
```

Arguments

x	D-part compositional data
ivar	Rationing part
base	a positive or complex number: the base with respect to which logarithms are computed. Defaults to `exp(1)`.

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 resulting coordinates
`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 useful when an inverse mapping 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 and Hall Ltd., London (UK). 416p.

See Also

`addLRinv`, `pivotCoord`

Examples

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

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

addLRinv *Inverse additive logratio mapping*

Description

Inverse additive logratio mapping, often called additive logistic transformation.

Usage

```
addLRinv(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 resulting compositional data matrix

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.

See Also

`pivotCoordInv`, `cenLRinv`, `cenLR`, `addLR`

Examples

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

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

---

aDist *Aitchison distance*

Description

Computes the Aitchison distance between two observations, between two data sets or within observations of one data set.

Usage

```
aDist(x, y = NULL)

iprod(x, y)
```

Arguments

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

Details

This distance measure accounts for the relative scale property of compositional data. It measures the distance between two compositions if x and y are vectors. It evaluates the 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 computes a n times n distance matrix (with n the number of observations/compositions) if only x is provided.

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

Value

The Aitchison distance between two compositions or between two data sets, or a distance matrix in case y is not supplied.

Author(s)

Matthias Templ, Bernhard Meindl

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

pivotCoord

Examples

```
data(expenditures)
x <- xOrig <- expenditures
## Aitchison distance between two 2 observations:
aDist(x[1, ], x[2, ])
aDist(as.numeric(x[1, ]), as.numeric(x[2, ]))


## Aitchison distance of x:
aDist(x)

## Example of distances between matrices:
## set some missing values:
x[1,3] <- x[3,5] <- x[2,4] <- x[5,3] <- x[8,3] <- NA

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

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

data("expenditures")
aDist(expenditures)
x <- expenditures[, 1]
y <- expenditures[, 2]
aDist(x, y)
aDist(expenditures, expenditures)
```

`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(pivotCoord(x[,1:2]))
adtest(pivotCoord(x[,1:3]))
adtest(pivotCoord(x))
adtest(pivotCoord(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)

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

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

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
...	additional parameters for print and summary passed through
object	an object of class adtestWrapper for the summary method

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.

A print and a summary method are implemented. The latter one provides a similar output is proposed by (Pawlowsky-Glahn, et al. (2008). In addition to that, p-values are provided.

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 and Hall Ltd., London (UK). 416p.

See Also

`adtest`, `pivotCoord`

Examples

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

ageCatWorld *child, middle and eldery population*

Description

Percentages of childs, middle generation and eldery population in 195 countries.

Usage

```
data(ageCatWorld)
```

Format

A data frame with 195 rows and 4 variables

Details

`<15` Percentage of people with age below 15

`15-60` Percentage of people with age between 15 and 60

`60+` Percentage of people with age above 60

`country` country of origin

The rows sum up to 100.

Author(s)

extracted by Karel Hron and Eva Fiserova, implemented by Matthias Templ

References

Fiserova, E. and Hron, K. (2012). Statistical Inference in Orthogonal Regression for Three-Part Compositional Data Using a Linear Model with Type-II Constraints. *Communications in Statistics - Theory and Methods*, 41 (13-14), 2367-2385.

Examples

```
data(ageCatWorld)
str(ageCatWorld)
summary(ageCatWorld)
rowSums(ageCatWorld[, 1:3])
ternaryDiag(ageCatWorld[, 1:3])
plot(pivotCoord(ageCatWorld[, 1:3]))
```

alcohol — *alcohol consumptions by country and type of alcohol*

Description

- `country` Country
- `year` Year
- `beer` Consumption of pure alcohol on beer (in percentages)
- `wine` Consumption of pure alcohol on wine (in percentages)
- `spirits` Consumption of pure alcohol on spirits (in percentages)
- `other` Consumption of pure alcohol on other beverages (in percentages)

Usage

```
data(alcohol)
```

Format

A data frame with 193 rows and 6 variables

Author(s)

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

Source

Transfered from the World Health Organisation website.

Examples

```
data("alcohol")
str(alcohol)
summary(alcohol)
```

alcoholreg — *regional alcohol per capita (15+) consumption by WHO region*

Description

- `country` Country
- `year` Year
- `recorded` Recorded alcohol consumption
- `unrecorded` Unrecorded alcohol consumption

Usage

```
data(alcoholreg)
```

Format

A data frame with 6 rows and 4 variables

Author(s)

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

Source

Transfered from the World Health Organisation website.

Examples

```
data("alcoholreg")
alcoholreg
```

---

arcticLake — *arctic 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 rows and 3 variables

Details

- `sand` numeric vector of percentages of sand
- `silt` numeric vector of percentages of silt
- `clay` numeric vector of percentages of clay

The rows sum up to 100, except for rounding errors.

Author(s)

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

References

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

Examples

```
data(arcticLake)
str(arcticLake)
summary(arcticLake)
rowSums(arcticLake)
ternaryDiag(arcticLake)
plot(pivotCoord(arcticLake))
```

`balances` *Balance calculation*

Description

Given a D-dimensional compositional data set and a sequential binary partition, the function bal calculates the balances in order to express the given data in the (D-1)-dimensional real space.

Usage

```
balances(x, y)
```

Arguments

`x`	data frame or matrix, typically compositional data
`y`	binary partition

Details

The sequential binary partition constructs an orthonormal basis in the (D-1)-dimensional hyperplane in real space, resulting in orthonormal coordinates with respect to the Aitchison geometry of compositional data.

Value

`balances`	The balances represent orthonormal coordinates which allow an interpretation in sense of groups of compositional parts. Output is a matrix, the D-1 colums contain balance coordinates of the observations in the rows.
`V`	A Dx(D-1) contrast matrix associated with the orthonormal basis, corresponding to the sequential binary partition (in clr coefficients).

Author(s)

Veronika Pintar, Karel Hron, Matthias Templ

References

(Egozcue, J.J., Pawlowsky-Glahn, V. (2005) Groups of parts and their balances in compositional data analysis. Mathematical Geology, 37 (7), 795???828.)

Examples

```
data(expenditures, package = "robCompositions")
y1 <- data.frame(c(1,1,1,-1,-1),c(1,-1,-1,0,0),
                 c(0,+1,-1,0,0),c(0,0,0,+1,-1))
y2 <- data.frame(c(1,-1,1,-1,-1),c(1,0,-1,0,0),
                 c(1,-1,1,-1,1),c(0,-1,0,1,0))
y3 <- data.frame(c(1,1,1,1,-1),c(-1,-1,-1,+1,0),
                 c(-1,-1,+1,0,0),c(-1,1,0,0,0))
y4 <- data.frame(c(1,1,1,-1,-1),c(0,0,0,-1,1),
                 c(-1,-1,+1,0,0),c(-1,1,0,0,0))
y5 <- data.frame(c(1,1,1,-1,-1),c(-1,-1,+1,0,0),
                 c(0,0,0,-1,1),c(-1,1,0,0,0))
b1 <- balances(expenditures, y1)
b2 <- balances(expenditures, y5)
b1$balances
b2$balances

data(machineOperators)
sbp <- data.frame(c(1,1,-1,-1),c(-1,+1,0,0),
                 c(0,0,+1,-1))
balances(machineOperators, sbp)
```

---

biomarker *biomarker*

Description

The function for identification of biomakers and outlier diagnostics as described in paper "Robust biomarker identification in a two-class problem based on pairwise log-ratios"

Usage

```
biomarker(
  x,
  cut = qnorm(0.975, 0, 1),
  g1,
  g2,
  type = "tau",
  diag = TRUE,
  plot = FALSE,
  diag.plot = FALSE
)
```

```
## S3 method for class 'biomarker'
plot(x, cut = qnorm(0.975, 0, 1), type = "Vstar", ...)

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

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

Arguments

`x`	data frame
`cut`	cut-off value, initialy set as 0.975 quantile of standard normal distribution
`g1`	vector with locations of observations of group 1
`g2`	vector with locations of observations of group 2
`type`	type of estimation of the variation matrix. Possible values are `"sd"`, `"mad"` and `"tau"`, representing Standard deviation, Median absolute deviation and Tau estimator of scale
`diag`	logical, indicating wheter outlier diagnostic should be computed
`plot`	logical, indicating wheter Vstar values should be plotted
`diag.plot`	logical, indicating wheter outlier diagnostic plot should be made
`...`	further arguments can be passed through
`object`	object of class biomarker

Details

Robust biomarker identification and outlier diagnostics

The method computes variation matrices separately with observations from both groups and also together with all observations. Then, V statistics is then computed and normalized. The variables, for which according V* values are bigger that the cut-off value are considered as biomarkers.

Value

The function returns object of type "biomarker". Functions `print`, `plot` and `summary` are available.

`biom.ident`	List of `V`, `Vstar`, `biomarkers`
`V`	Values of V statistics
`Vstar`	Normalizes values of V statistics (V^* values))
`biomarkers`	Logical value, indicating if certain variable was identified as biomarker
`diag`	Outlier diagnostics (returned only if `diag=TRUE`)

Author(s)

Jan Walach

See Also

plot.biomarker

Examples

```
# Data simulation
set.seed(4523)
n <- 40; p <- 50
r <- runif(p, min = 1, max = 10)
conc <- runif(p, min = 0, max = 1)*5+matrix(1,p,1)*5
a <- conc*r
S <- rnorm(n,0,0.3) %*% t(rep(1,p))
B <- matrix(rnorm(n*p,0,0.8),n,p)
R <- rep(1,n) %*% t(r)
M <- matrix(rnorm(n*p,0,0.021),n,p)
# Fifth observation is an outlier
M[5,] <- M[5,]*3 + sample(c(0.5,-0.5),replace=TRUE,p)
C <- rep(1,n) %*% t(conc)
C[1:20,c(2,15,28,40)] <- C[1:20,c(2,15,28,40)]+matrix(1,20,4)*1.8
X <- (1-S)*(C*R+B)*exp(M)
# Biomarker identification
b <- biomarker(X, g1 = 1:20, g2 = 21:40, type = "tau")
```

---

biplot.factanal *Biplot method*

Description

Provides robust compositional biplots.

Usage

```
## S3 method for class 'factanal'
biplot(x, ...)
```

Arguments

x	object of class 'factanal'
...	...

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

pfa

Examples

```
data(expenditures)
res.rob <- pfa(expenditures, factors=2, scores = "regression")
biplot(res.rob)
```

---

biplot.pcaCoDa *Biplot method*

Description

Provides robust compositional biplots.

Usage

```
## S3 method for class 'pcaCoDa'
biplot(x, y, ..., choices = 1:2)
```

Arguments

x	object of class 'pcaCoDa'
y	...
...	arguments passed to plot methods
choices	selection of two principal components by number. Default: c(1,2)

Details

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

Examples

```
data(coffee)
p1 <- pcaCoDa(coffee[,-1])
p1
plot(p1, which = 2, choices = 1:2)

# exemplarly, showing the first and third PC
a <- p1$princompOutputClr
biplot(a, choices = c(1,3))


## with labels for the scores:
data(arcticLake)
rownames(arcticLake) <- paste(sample(letters[1:26], nrow(arcticLake), replace=TRUE),
                              1:nrow(arcticLake), sep="")
pc <- pcaCoDa(arcticLake, method="classical")
plot(pc, xlabs=rownames(arcticLake), which = 2)
plot(pc, xlabs=rownames(arcticLake), which = 3)
```

---

bootnComp   *Bootstrap to find optimal number of components*

Description

Combined bootstrap and cross validation procedure to find optimal number of PLS components

Usage

```
bootnComp(X, y, R = 99, plotting = FALSE)
```

Arguments

`X`	predictors as a matrix
`y`	response
`R`	number of bootstrap replicates
`plotting`	if TRUE, a diagnostic plot is drawn for each bootstrap replicate

Details

Heavily used internally in function impRZilr.

Value

Including other information in a list, the optimal number of components

Author(s)

Matthias Templ

See Also

`impRZilr`

Examples

```
## we refer to impRZilr()
```

---

bpc *Backwards pivot coordinates and their inverse*

Description

Backwards pivot coordinate representation of a set of compositional ventors as a special case of isometric logratio coordinates and their inverse mapping.

Usage

```
bpc(X, base = exp(1))
```

Arguments

`X`	object of class data.frame. Positive values only.
`base`	a positive number: the base with respect to which logarithms are computed. Defaults to exp(1).

Details

bpc

Backwards pivot coordinates map D-part compositional data from the simplex into a (D-1)-dimensional real space isometrically. The first coordinate has form of pairwise logratio log(x2/x1) and serves as an alternative to additive logratio transformation with part x1 being the rationing element. The remaining coordinates are structured as detailed in Nesrstova et al. (2023). Consequently, when a specific pairwise logratio is of the main interest, the respective columns have to be placed at the first (the compositional part in denominator of the logratio, the rationing element) and the second position (the compositional part in numerator) in the data matrix X.

Value

`Coordinates` array of orthonormal coordinates.

`Coordinates.ortg`
array of orthogonal coordinates (without the normalising constant sqrt(i/i+1).

`Contrast.matrix`
contrast matrix corresponding to the orthonormal coordinates.

`Base` the base with respect to which logarithms are computed.

`Levels` the order of compositional parts.

Author(s)

Kamila Facevicova

References

Hron, K., Coenders, G., Filzmoser, P., Palarea-Albaladejo, J., Famera, M., Matys Grygar, M. (2022). Analysing pairwise logratios revisited. Mathematical Geosciences 53, 1643 - 1666.

Nesrstova, V., Jaskova, P., Pavlu, I., Hron, K., Palarea-Albaladejo, J., Gaba, A., Pelclova, J., Facevicova, K. (2023). Simple enough, but not simpler: Reconsidering additive logratio coordinates in compositional analysis. Submitted

See Also

bpcTab bpcTabWrapper bpcPca bpcReg

Examples

```
data(expenditures)

# default setting with ln()
bpc(expenditures)

# logarithm of base 2
bpc(expenditures, base = 2)
```

bpcPca *Principal component analysis based on backwards pivot coordinates*

Description

Performs classical or robust principal component analysis on system of backwards pivot coordinates and returns the result related to pairwise logratios as well as the clr representation.

Usage

```
bpcPca(X, robust = FALSE, norm.cat = NULL)
```

Arguments

`X`	object of class data.frame. Positive values only.
`robust`	if TRUE, the MCD estimate is used. Defaults to FALSE.
`norm.cat`	the rationing category placed at the first position in the composition. If not defined, all pairwise logratios are considered. Given in quotation marks.

Details

bpcPca

The compositional data set is repeatedly expressed in a set of backwards logratio coordinates, when each set highlights one pairwise logratio (or one pairwise logratio with the selected rationing category). For each set, robust or classical principal component analysis is performed and loadings respective to the first backwards pivot coordinate are stored. The procedure results in matrix of scores (invariant to the specific coordinate system), clr loading matrix and matrix with loadings respective to pairwise logratios.

Value

`scores`	array of scores.
`loadings`	loadings related to the pairwise logratios. The names of the rows indicate the type of the respective coordinate (bpc.1 - the first backwards pivot coordinate) and the logratio quantified thereby. E.g. bpc.1_C2.to.C1 would therefore correspond to the logratio between compositional parts C1 and C2, schematically written log(C2/C1). See Nesrstova et al. (2023) for details.
`loadings.clr`	loadings in the clr space.
`sdev`	standard deviations of the principal components.
`center`	means of the pairwise logratios.
`center.clr`	means of the clr coordinates.
`n.obs`	number of observations.

Author(s)

Kamila Facevicova

References

Hron, K., Coenders, G., Filzmoser, P., Palarea-Albaladejo, J., Famera, M., Matys Grygar, M. (2022). Analysing pairwise logratios revisited. Mathematical Geosciences 53, 1643 - 1666.

Nesrstova, V., Jaskova, P., Pavlu, I., Hron, K., Palarea-Albaladejo, J., Gaba, A., Pelclova, J., Facevicova, K. (2023). Simple enough, but not simpler: Reconsidering additive logratio coordinates in compositional analysis. Submitted

See Also

bpc bpcPcaTab bpcReg

Examples

```
data(arcticLake)

# classical estimation with all pairwise logratios:
res.cla <- bpcPca(arcticLake)
summary(res.cla)
biplot(res.cla)
head(res.cla$scores)
res.cla$loadings
res.cla$loadings.clr

# similar output as from pca CoDa
res.cla2 <- pcaCoDa(arcticLake, method="classical", solve = "eigen")
biplot(res.cla2)
head(res.cla2$scores)
res.cla2$loadings

# classical estimation focusing on pairwise logratios with clay:
res.cla.clay <- bpcPca(arcticLake, norm.cat = "clay")
biplot(res.cla.clay)

# robust estimation with all pairwise logratios:
res.rob <- bpcPca(arcticLake, robust = TRUE)
biplot(res.rob)
```

---

bpcPcaTab *Principal component analysis of compositional tables based on backwards pivot coordinates*

Description

Performs classical or robust principal component analysis on a set of compositional tables, based on backwards pivot coordinates. Returns the result related to pairwise row and column balances and four-part log odds-ratios. The loadings in the clr space are available as well.

Usage

```
bpcPcaTab(
  X,
  obs.ID = NULL,
  row.factor = NULL,
  col.factor = NULL,
  value = NULL,
  robust = FALSE,
  norm.cat.row = NULL,
  norm.cat.col = NULL
)
```

Arguments

`X`	object of class data.frame with columns corresponding to row and column factors of the respective compositional table, a variable with the values of the composition (positive values only) and a factor with observation IDs.
`obs.ID`	name of the factor variable distinguishing the observations. Needs to be given with the quotation marks.
`row.factor`	name of the variable representing the row factor. Needs to be given with the quotation marks.
`col.factor`	name of the variable representing the column factor. Needs to be given with the quotation marks.
`value`	name of the variable representing the values of the composition. Needs to be given with the quotation marks.
`robust`	if TRUE, the MCD estimate is used. Defaults to FALSE.
`norm.cat.row`	the rationing category of the row factor. If not defined, all pairs are considered. Given in quotation marks.
`norm.cat.col`	the rationing category of the column factor. If not defined, all pairs are considered. Given in quotation marks.

Details

bpcPcaTab

The set of compositional tables is repeatedly expressed in a set of backwards logratio coordinates, when each set highlights different combination of pairs of row and column factor categories, as detailed in Nesrstova et al. (2023). For each set, robust or classical principal component analysis is performed and loadings respective to the first row, column and odds-ratio backwards pivot coordinates are stored. The procedure results in matrix of scores (invariant to the specific coordinate system), clr loading matrix and matrix with loadings related to the selected backwards coordinates.

Value

`scores`	array of scores.

`loadings` loadings related to the selected backwards coordinates. The names of the rows indicate the type of the respective coordinate (rbpb.1 - the first row backwards pivot balance, cbpb.1 - the first column backwards pivot balance and tbpc.1.1 - the first table backwards pivot coordinate) and the logratio or log odds-ratio quantified thereby. E.g. cbpb.1_C2.to.C1 would therefore correspond to the logratio between column categories C1 and C2, schematically written log(C2/C1), and tbpc.1.1_R2.to.R1.&.C2.to.C1 would correspond to the log odds-ratio computed from a 2x2 table, which is formed by row categories R1 and R2 and columns C1 and C2. See Nesrstova et al. (2023) for details.

`loadings.clr` loadings in the clr space. The names of the rows indicate the position of respective part in the clr representation of the compositional table, labeled as row.category_column.category.

`sdev` standard deviations of the principal components.

`center` means of the selected backwards coordinates.

`center.clr` means of the clr coordinates.

`n.obs` number of observations.

Author(s)

Kamila Facevicova

References

Nesrstova, V., Jaskova, P., Pavlu, I., Hron, K., Palarea-Albaladejo, J., Gaba, A., Pelclova, J., Facevicova, K. (2023). Simple enough, but not simpler: Reconsidering additive logratio coordinates in compositional analysis. Submitted

See Also

bpcTabWrapper bpcPca bpcRegTab

Examples

```
data(manu_abs)
manu_abs$output <- as.factor(manu_abs$output)
manu_abs$isic <- as.factor(manu_abs$isic)

# classical estimation with all pairwise balances and four-part ORs:
res.cla <- bpcPcaTab(manu_abs, obs.ID = "country", row.factor = "output",
col.factor = "isic", value = "value")
summary(res.cla)
biplot(res.cla)
head(res.cla$scores)
res.cla$loadings
res.cla$loadings.clr

# classical estimation with LAB anf 155 as rationing categories
res.cla.select <- bpcPcaTab(manu_abs, obs.ID = "country", row.factor = "output",
col.factor = "isic", value = "value", norm.cat.row = "LAB", norm.cat.col = "155")
```

```
summary(res.cla.select)
biplot(res.cla.select)
head(res.cla.select$scores)
res.cla.select$loadings
res.cla.select$loadings.clr

# robust estimation with all pairwise balances and four-part ORs:
res.rob <- bpcPcaTab(manu_abs, obs.ID = "country", row.factor = "output",
col.factor = "isic", value = "value", robust = TRUE)
summary(res.rob)
biplot(res.rob)
head(res.rob$scores)
res.rob$loadings
res.rob$loadings.clr
```

---

bpcReg *Classical and robust regression based on backwards pivot coordinates*

Description

Performs classical or robust regression analysis of real response on compositional predictors, represented in backwards pivot coordinates. Also non-compositional covariates can be included (additively).

Usage

```
bpcReg(
  X,
  y,
  external = NULL,
  norm.cat = NULL,
  robust = FALSE,
  base = exp(1),
  norm.const = F,
  seed = 8
)
```

Arguments

X	object of class data.frame with compositional (positive values only) and non-compositional predictors. The response y can be also included.
y	character with the name of response (if included in X) or an array with values of the response.
external	array with names of non-compositional predictors.
norm.cat	the rationing category placed at the first position in the composition. If not defined, all pairwise logratios are considered. Given in quotation marks.
robust	if TRUE, the MM-type estimator is used. Defaults to FALSE.

base	a positive number: the base with respect to which logarithms are computed. Defaults to exp(1).
norm.const	if TRUE, the regression coefficients corresponding to orthonormal coordinates are given a s result. Defaults to FALSE, the normalising constant is omitted.
seed	a single value.

Details

bpcReg

The compositional part of the data set is repeatedly expressed in a set of backwards logratio coordinates, when each set highlights one pairwise logratio (or one pairwise logratio with the selected rationing category). For each set (supplemented by non-compositonal predictors), robust MM or classical least squares estimate of regression coefficients is performed and information respective to the first backwards pivot coordinate is stored. The summary therefore collects results from several regression models, each leading to the same overall model characteristics, like the F statistics or R^2. The coordinates are structured as detailed in Nesrstova et al. (2023). In order to maintain consistency of the iterative results collected in the output, a seed is set before robust estimation of each of the models considered. Its specific value can be set via parameter seed.

Value

A list containing:

Summary the summary object which collects results from all coordinate systems. The names of the coefficients indicate the type of the respective coordinate (bpc.1 - the first backwards pivot coordinate) and the logratio quantified thereby. E.g. bpc.1_C2.to.C1 would therefore correspond to the logratio between compositional parts C1 and C2, schematically written log(C2/C1). See Nesrstova et al. (2023) for details.

Base the base with respect to which logarithms are computed

Norm.const the values of normalising constants (when results for orthonormal coordinates are reported).

Robust TRUE if the MM estimator was applied.

lm the lm object resulting from the first iteration.

Levels the order of compositional parts cosidered in the first iteration.

Author(s)

Kamila Facevicova

References

Hron, K., Coenders, G., Filzmoser, P., Palarea-Albaladejo, J., Famera, M., Matys Grygar, M. (2022). Analysing pairwise logratios revisited. Mathematical Geosciences 53, 1643 - 1666.

Nesrstova, V., Jaskova, P., Pavlu, I., Hron, K., Palarea-Albaladejo, J., Gaba, A., Pelclova, J., Facevicova, K. (2023). Simple enough, but not simpler: Reconsidering additive logratio coordinates in compositional analysis. Submitted

See Also

bpc bpcPca bpcRegTab

Examples

```
## How the total household expenditures in EU Member
## States depend on relative contributions of
## single household expenditures:
data(expendituresEU)
y <- as.numeric(apply(expendituresEU,1,sum))

# classical regression summarizing the effect of all pairwise logratios
lm.cla <- bpcReg(expendituresEU, y)
lm.cla

# gives the same model characteristics as lmCoDaX:
lm <- lmCoDaX(y, expendituresEU, method="classical")
lm$ilr

# robust regression, with Food as the rationing category and logarithm of base 2
# response is part of the data matrix X
expendituresEU.y <- data.frame(expendituresEU, total = y)
lm.rob <- bpcReg(expendituresEU.y, "total", norm.cat = "Food", robust = TRUE, base = 2)
lm.rob

## Illustrative example with exports and imports (categorized) as non-compositional covariates
data(economy)
X.ext <- economy[!economy$country2 %in% c("HR", "NO", "CH"), c("exports", "imports")]
X.ext$imports.cat <- cut(X.ext$imports, quantile(X.ext$imports, c(0, 1/3, 2/3, 1)),
labels = c("A", "B", "C"), include.lowest = TRUE)

X.y.ext <- data.frame(expendituresEU.y, X.ext[, c("exports", "imports.cat")])

lm.ext <- bpcReg(X.y.ext, y = "total", external = c("exports", "imports.cat"))
lm.ext
```

---

bpcRegTab — *Classical and robust regression based on backwards pivot coordinates*

Description

Performs classical or robust regression analysis of real response on a compositional table, which is represented in backwards pivot coordinates. Also non-compositional covariates can be included (additively).

Usage

```
bpcRegTab(
  X,
```

```
  y,
  obs.ID = NULL,
  row.factor = NULL,
  col.factor = NULL,
  value = NULL,
  external = NULL,
  norm.cat.row = NULL,
  norm.cat.col = NULL,
  robust = FALSE,
  base = exp(1),
  norm.const = F,
  seed = 8
)
```

Arguments

`X`	object of class data.frame with columns corresponding to row and column factors of the respective compositional table, a variable with the values of the composition (positive values only) and a factor with observation IDs. The response y and non-compositional predictors can be also included.
`y`	character with the name of response (if included in X), data frame with row names corresponding to observation IDs or a named array with values of the response.
`obs.ID`	name of the factor variable distinguishing the observations. Needs to be given with the quotation marks.
`row.factor`	name of the variable representing the row factor. Needs to be given with the quotation marks.
`col.factor`	name of the variable representing the column factor. Needs to be given with the quotation marks.
`value`	name of the variable representing the values of the composition. Needs to be given with the quotation marks.
`external`	array with names of non-compositional predictors.
`norm.cat.row`	the rationing category of the row factor. If not defined, all pairs are considered. Given in quotation marks.
`norm.cat.col`	the rationing category of the column factor. If not defined, all pairs are considered. Given in quotation marks.
`robust`	if TRUE, the MM-type estimator is used. Defaults to FALSE.
`base`	a positive number: the base with respect to which logarithms are computed. Defaults to exp(1).
`norm.const`	if TRUE, the regression coefficients corresponding to orthonormal coordinates are given a s result. Defaults to FALSE, the normalising constant is omitted.
`seed`	a single value.

Details

bpcRegTab

The set of compositional tables is repeatedly expressed in a set of backwards logratio coordinates, when each set highlights different combination of pairs of row and column factor categories, as detailed in Nesrstova et al. (2023). For each coordinates system (supplemented by non-compositonal predictors), robust MM or classical least squares estimate of regression coefficients is performed and information respective to the first row, column and table backwards pivot coordinate is stored. The summary therefore collects results from several regression models, each leading to the same overall model characteristics, like the F statistics or R^2. In order to maintain consistency of the iterative results collected in the output, a seed is set before robust estimation of each of the models considered. Its specific value can be set via parameter seed.

Value

A list containing:

Summary the summary object which collects results from all coordinate systems. The names of the coefficients indicate the type of the respective coordinate (rbpb.1 - the first row backwards pivot balance, cbpb.1 - the first column backwards pivot balance and tbpc.1.1 - the first table backwards pivot coordinate) and the logratio or log odds-ratio quantified thereby. E.g. cbpb.1_C2.to.C1 would therefore correspond to the logratio between column categories C1 and C2, schematically written log(C2/C1), and tbpc.1.1_R2.to.R1.&.C2.to.C1 would correspond to the log odds-ratio computed from a 2x2 table, which is formed by row categories R1 and R2 and columns C1 and C2. See Nesrstova et al. (2023) for details.

Base the base with respect to which logarithms are computed

Norm.const the values of normalising constants (when results for orthonormal coordinates are reported).

Robust TRUE if the MM estimator was applied.

lm the lm object resulting from the first iteration.

Row.levels the order of the row factor levels cosidered in the first iteration.

Col.levels the order of the column factor levels cosidered in the first iteration.

Author(s)

Kamila Facevicova

References

Nesrstova, V., Jaskova, P., Pavlu, I., Hron, K., Palarea-Albaladejo, J., Gaba, A., Pelclova, J., Facevicova, K. (2023). Simple enough, but not simpler: Reconsidering additive logratio coordinates in compositional analysis. Submitted

See Also

bpcTabWrapper bpcPcaTab bpcReg

Examples

```
# let's prepare some data
data(employment2)
data(unemployed)

table_data <- employment2[employment2$Contract == "FT", ]
y <- unemployed[unemployed$age == "20_24" & unemployed$year == 2015,]
countries <- intersect(levels(droplevels(y$country)), levels(table_data$Country))

table_data <- table_data[table_data$Country %in% countries, ]
y <- y[y$country %in% countries, c("country", "value")]
colnames(y) <- c("Country", "unemployed")

# response as part of X
table_data.y <- merge(table_data, y, by = "Country")
reg.cla <- bpcRegTab(table_data.y, y = "unemployed", obs.ID = "Country",
row.factor = "Sex", col.factor = "Age", value = "Value")
reg.cla

# response as named array
resp <- y$unemployed
names(resp) <- y$Country
reg.cla2 <- bpcRegTab(table_data.y, y = resp, obs.ID = "Country",
row.factor = "Sex", col.factor = "Age", value = "Value")
reg.cla2

# response as data.frame, robust estimator, 55plus as the rationing category, logarithm of base 2
resp.df <- as.data.frame(y$unemployed)
rownames(resp.df) <- y$Country
reg.rob <- bpcRegTab(table_data.y, y = resp.df, obs.ID = "Country",
row.factor = "Sex", col.factor = "Age", value = "Value",
norm.cat.col = "55plus", robust = TRUE, base = 2)
reg.rob

# Illustrative example with non-compositional predictors and response as part of X
x.ext <- unemployed[unemployed$age == "15_19" & unemployed$year == 2015,]
x.ext <- x.ext[x.ext$country %in% countries, c("country", "value")]
colnames(x.ext) <- c("Country", "15_19")

table_data.y.ext <- merge(table_data.y, x.ext, by = "Country")
reg.cla.ext <- bpcRegTab(table_data.y.ext, y = "unemployed", obs.ID = "Country",
row.factor = "Sex", col.factor = "Age", value = "Value", external = "15_19")
reg.cla.ext
```

---

bpcTab *Backwards pivot coordinates and their inverse*

Description

Backwards pivot coordinate representation of a compositional table as a special case of isometric logratio coordinates and their inverse mapping.

Usage

```
bpcTab(x, row.factor = NULL, col.factor = NULL, value = NULL, base = exp(1))
```

Arguments

`x`	object of class data.frame with columns corresponding to row and column factors of the respective compositional table and a variable with the values of the composition (positive values only).
`row.factor`	name of the variable representing the row factor. Needs to be given with the quotation marks.
`col.factor`	name of the variable representing the column factor. Needs to be given with the quotation marks.
`value`	name of the variable representing the values of the composition. Needs to be given with the quotation marks.
`base`	a positive number: the base with respect to which logarithms are computed. Defaults to exp(1).

Details

bpcTab

Backwards pivot coordinates map IxJ-part compositional table from the simplex into a (IJ-1)-dimensional real space isometrically. Particularly the first coordinate from each group (rbpb.1, cbpb.1, tbpc.1) preserves the elemental information on the two-factorial structure. The first row and column backwards pivot balances rbpb.1 and cbpb.1 represent two-factorial counterparts to the pairwise logratios. More specifically, the first two levels of the considered factor are compared in the ratio, while the first level plays the role of the rationing category (denominator of the ratio) and the second level is treated as the normalized category (numerator of the ratio). All categories of the complementary factor are aggregated with the geometric mean. The first table backwards pivot coordinate, has form of a four-part log odds-ratio (again related to the first two levels of the row and column factors) and quantifies the relations between factors. All coordinates are structured as detailed in Nesrstova et al. (2023).

Value

`Coordinates`	array of orthonormal coordinates.
`Coordinates.ortg`	array of orthogonal coordinates.
`Contrast.matrix`	contrast matrix corresponding to the orthonormal coordinates.
`Base`	the base with respect to which logarithms are computed.
`Row.levels`	order of the row factor levels.
`Col.levels`	order of the column factor levels.

Author(s)

Kamila Facevicova

References

Nesrstova, V., Jaskova, P., Pavlu, I., Hron, K., Palarea-Albaladejo, J., Gaba, A., Pelclova, J., Facevicova, K. (2023). Simple enough, but not simpler: Reconsidering additive logratio coordinates in compositional analysis. Submitted

See Also

bpc bpcTabWrapper bpcPcaTab bpcRegTab

Examples

```
data(manu_abs)
manu_USA <- manu_abs[which(manu_abs$country=='USA'),]
manu_USA$output <- as.factor(manu_USA$output)
manu_USA$isic <- as.factor(manu_USA$isic)

# default setting with ln()
bpcTab(manu_USA, row.factor = "output", col.factor = "isic", value = "value")

# logarithm of base 2
bpcTab(manu_USA, row.factor = "output", col.factor = "isic", value = "value",
base = 2)

# for base exp(1) is the result similar to tabCoord():
r <- rbind(c(-1,1,0), c(-1,-1,1))
c <- rbind(c(-1,1,0,0,0), c(-1,-1,1,0,0), c(-1,-1,-1,1,0), c(-1,-1,-1,-1,1))
tabCoord(manu_USA, row.factor = "output", col.factor = "isic", value = "value",
SBPr = r, SBPc = c)
```

---

bpcTabWrapper — *Backwards pivot coordinates and their inverse*

Description

For each compositional table in the sample a system of backwards pivot coordinates is computed as a special case of isometric logratio coordinates. For their inverse mapping, the contrast matrix is provided.

Usage

```
bpcTabWrapper(
  X,
  obs.ID = NULL,
  row.factor = NULL,
  col.factor = NULL,
  value = NULL,
  base = exp(1)
)
```

Arguments

`X`	object of class data.frame with columns corresponding to row and column factors of the respective compositional table, a variable with the values of the composition (positive values only) and a factor with observation IDs.
`obs.ID`	name of the factor variable distinguishing the observations. Needs to be given with the quotation marks.
`row.factor`	name of the variable representing the row factor. Needs to be given with the quotation marks.
`col.factor`	name of the variable representing the column factor. Needs to be given with the quotation marks.
`value`	name of the variable representing the values of the composition. Needs to be given with the quotation marks.
`base`	a positive number: the base with respect to which logarithms are computed. Defaults to exp(1).

Details

bpcTabWrapper

Backwards pivot coordinates map IxJ-part compositional table from the simplex into a (IJ-1)-dimensional real space isometrically. Particularly the first coordinate from each group (rbpb.1, cbpb.1, tbpc.1) preserves the elemental information on the two-factorial structure. The first row and column backwards pivot balances rbpb.1 and cbpb.1 represent two-factorial counterparts to the pairwise logratios. More specifically, the first two levels of the considered factor are compared in the ratio, while the first level plays the role of the rationing category (denominator of the ratio) and the second level is treated as the normalized category (numerator of the ratio). All categories of the complementary factor are aggregated with the geometric mean. The first table backwards pivot coordinate, has form of a four-part log odds-ratio (again related to the first two levels of the row and column factors) and quantifies the relations between factors. All coordinates are structured as detailed in Nesrstova et al. (2023).

Value

`Coordinates`	array of orthonormal coordinates.
`Coordinates.ortg`	array of orthogonal coordinates.
`Contrast.matrix`	contrast matrix corresponding to the orthonormal coordinates.
`Base`	the base with respect to which logarithms are computed.
`Row.levels`	order of the row factor levels.
`Col.levels`	order of the column factor levels.

Author(s)

Kamila Facevicova

References

Nesrstova, V., Jaskova, P., Pavlu, I., Hron, K., Palarea-Albaladejo, J., Gaba, A., Pelclova, J., Facevicova, K. (2023). Simple enough, but not simpler: Reconsidering additive logratio coordinates in compositional analysis. Submitted

See Also

bpc bpcPcaTab bpcRegTab

Examples

```
data(manu_abs)
manu_abs$output <- as.factor(manu_abs$output)
manu_abs$isic <- as.factor(manu_abs$isic)

# default setting with ln()
bpcTabWrapper(manu_abs, obs.ID = "country", row.factor = "output",
col.factor = "isic", value = "value")

# logarithm of base 2
bpcTabWrapper(manu_abs, obs.ID = "country", row.factor = "output",
col.factor = "isic", value = "value", base = 2)

# for base exp(1) is the result similar to tabCoordWrapper():
r <- rbind(c(-1,1,0), c(-1,-1,1))
c <- rbind(c(-1,1,0,0,0), c(-1,-1,1,0,0), c(-1,-1,-1,1,0), c(-1,-1,-1,-1,1))
tabCoordWrapper(manu_abs, obs.ID = "country", row.factor = "output",
col.factor = "isic", value = "value", SBPr = r, SBPc = c)
```

---

cancer *hospital discharges on cancer and distribution of age*

Description

Hospital discharges of in-patients on neoplasms (cancer) per 100.000 inhabitants (year 2007) and population age structure.

Format

A data set on 24 compositions on 6 variables.

Details

`country` country

`year` year

`p1` percentage of population with age below 15

`p2` percentage of population with age between 15 and 60 `p3` percentage of population with age above 60

`discharges` hospital discharges of in-patients on neoplasms (cancer) per 100.000 inhabitants

The response (discharges) is provided for the European Union countries (except Greece, Hungary and Malta) by Eurostat. As explanatory variables we use the age structure of the population in the same countries (year 2008). The age structure consists of three parts, age smaller than 15, age between 15 and 60 and age above 60 years, and they are expressed as percentages on the overall population in the countries. The data are provided by the United Nations Statistics Division.

Author(s)

conversion to R by Karel Hron and Matthias Templ <matthias.templ@tuwien.ac.at>

Source

https://www.ec.europa.eu/eurostat and https://unstats.un.org/home/

References

K. Hron, P. Filzmoser, K. Thompson (2012). Linear regression with compositional explanatory variables. *Journal of Applied Statistics*, Volume 39, Issue 5, 2012.

Examples

```
data(cancer)
str(cancer)
```

`cancerMN` *malignant neoplasms cancer*

Description

Two main types of malignant neoplasms cancer affecting colon and lung, respectively, in male and female populations. For this purpose population data (2012) from 35 OECD countries were collected.

Format

A data set on 35 compositional tables on 4 parts (row-wise sorted cells) and 5 variables.

Details

`country` country

`females-colon` number of colon cancer cases in female population

`females-lung` number of lung cancer cases in female population

`males-colon` number of colon cancer cases in male population

`males-lung` number of lung cancer cases in male population

The data are obtained from the OECD website.

Author(s)

conversion to R by Karel Hron and intergration by Matthias Templ <matthias.templ@tuwien.ac.at>

Source

From OECD website

Examples

```
data(cancerMN)
head(cancerMN)
rowSums(cancerMN[, 2:5])
```

---

ced *Compositional error deviation*

Description

Normalized Aitchison distance between two data sets

Usage

```
ced(x, y, ni)
```

Arguments

x	matrix or data frame
y	matrix or data frame of the same size as x
ni	normalization parameter. See details below.

Details

This function has been mainly written for procudures that evaluate imputation or replacement of rounded zeros. The ni parameter can thus, e.g. be used for expressing the number of rounded zeros.

Value

the compositinal error distance

Author(s)

Matthias Templ

References

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

Templ, M., Hron, K., Filzmoser, P., Gardlo, A. (2016). Imputation of rounded zeros for high-dimensional compositional data. *Chemometrics and Intelligent Laboratory Systems*, 155, 183-190.

See Also

rdcm

Examples

```
data(expenditures)
x <- expenditures
x[1,3] <- NA
xi <- impKNNa(x)$xImp
ced(expenditures, xi, ni = sum(is.na(x)))
```

---

cenLR *Centred logratio coefficients*

Description

The centred logratio (clr) coefficients map D-part compositional data from the simplex into a D-dimensional real space.

Usage

```
cenLR(x, base = exp(1))
```

Arguments

x	multivariate data, ideally of class data.frame or matrix
base	a positive or complex number: the base with respect to which logarithms are computed. Defaults to `exp(1)`.

Details

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

Value

the resulting clr coefficients, including

x.clr	clr coefficients
gm	the geometric means of the original compositional data.

Note

The resulting 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 and Hall Ltd., London (UK). 416p.

See Also

`cenLRinv`, `addLR`, `pivotCoord`, `addLRinv`, `pivotCoordInv`

Examples

```
data(expenditures)
eclr <- cenLR(expenditures)
inveclr <- cenLRinv(eclr)
head(expenditures)
head(inveclr)
head(pivotCoordInv(eclr$x.clr))
```

---

cenLRinv *Inverse centred logratio mapping*

Description

Applies the inverse centred logratio mapping.

Usage

```
cenLRinv(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 resulting compositional data set.

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.

See Also

`cenLR`, `addLR`, `pivotCoord`, `addLRinv`, `pivotCoordInv`

Examples

```
data(expenditures)
eclr <- cenLR(expenditures, 2)
inveclr <- cenLRinv(eclr)
head(expenditures)
head(inveclr)
head(cenLRinv(eclr$x.clr))
```

---

chorizonDL — *C-horizon of the Kola data with rounded zeros*

Description

This data set is almost the same as the 'chorizon' data set in package `mvoutlier` and `chorizonDL`, except that values below the detection limit are coded as zeros, and detection limits provided as attributes to the data set and less variables are included.

Format

A data frame with 606 observations on the following 62 variables.

***ID** a numeric vector

XCOO a numeric vector

YCOO a numeric vector

Ag concentration in mg/kg

Al concentration in mg/kg

Al_XRF concentration in wt. percentage

As concentration in mg/kg

Ba concentration in mg/kg

Ba_INAA concentration in mg/kg

Be concentration in mg/kg

Bi concentration in mg/kg

Ca concentration in mg/kg

Ca_XRF concentration in wt. percentage

Cd concentration in mg/kg

Ce_INAA concentration in mg/kg

Co concentration in mg/kg

Co_INAA concentration in mg/kg

Cr concentration in mg/kg

Cr_INAA concentration in mg/kg

Cu concentration in mg/kg

Eu_INAA concentration in mg/kg

Fe concentration in mg/kg

Fe_XRF concentration in wt. percentage

Hf_INAA concentration in mg/kg

K concentration in mg/kg

K_XRF concentration in wt. percentage

La concentration in mg/kg

La_INAA concentration in mg/kg

Li concentration in mg/kg

Lu_INAA concentration in mg/kg

Mg concentration in mg/kg

Mg_XRF concentration in wt. percentage

Mn concentration in mg/kg

Mn_XRF concentration in wt. percentage

Na concentration in mg/kg

Na_XRF concentration in wt. percentage

Nd_INAA concentration in mg/kg

Ni concentration in mg/kg

P concentration in mg/kg

P_XRF concentration in wt. percentage

Pb concentration in mg/kg

S concentration in mg/kg

Sc concentration in mg/kg

Sc_INAA concentration in mg/kg

Si concentration in mg/kg

Si_XRF concentration in wt. percentage

Sm_INAA concentration in mg/kg

Sr concentration in mg/kg

Th_INAA concentration in mg/kg

Ti concentration in mg/kg

Ti_XRF concentration in wt. percentage

V concentration in mg/kg

Y concentration in mg/kg

Yb_INAA concentration in mg/kg

Zn concentration in mg/kg

LOI concentration in wt. percentage

pH ph value

ELEV elevation

***COUN** country

***ASP** a numeric vector

TOPC a numeric vector

LITO information on lithography

Note

For a more detailed description of this data set, see ‘chorizon‘ in package `mvoutlier`.

Source

Kola Project (1993-1998)

References

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

See Also

‘chorizon‘, `chorizonDL`

Examples

```
data(chorizonDL, package = "robCompositions")
dim(chorizonDL)
colnames(chorizonDL)
zeroPatterns(chorizonDL)
```

clustCoDa *Cluster analysis for compositional data*

Description

Clustering in orthonormal coordinates or by using the Aitchison distance

Usage

```
clustCoDa(
  x,
  k = NULL,
  method = "Mclust",
  scale = "robust",
  transformation = "pivotCoord",
  distMethod = NULL,
  iter.max = 100,
  vals = TRUE,
  alt = NULL,
  bic = NULL,
  verbose = TRUE
)

## S3 method for class 'clustCoDa'
plot(
  x,
  y,
  ...,
  normalized = FALSE,
  which.plot = "clusterMeans",
  measure = "silwidths"
)
```

Arguments

`x`	compositional data represented as a data.frame
`k`	number of clusters
`method`	clustering method. One of Mclust, cmeans, kmeansHartigan, cmeansUfcl, pam, clara, fanny, ward.D2, single, hclustComplete, average, mcquitty, median, centroid
`scale`	if orthonormal coordinates should be normalized.
`transformation`	default are the isometric logratio coordinates. Can only used when distMethod is not Aitchison.
`distMethod`	Distance measure to be used. If “Aitchison”, then transformation should be “identity”.

`iter.max`	parameter if kmeans is chosen. The maximum number of iterations allowed
`vals`	if cluster validity measures should be calculated
`alt`	a known partitioning can be provided (for special cluster validity measures)
`bic`	if TRUE then the BIC criteria is evaluated for each single cluster as validity measure
`verbose`	if TRUE additional print output is provided
`y`	the y coordinates of points in the plot, optional if x is an appropriate structure.
`...`	additional parameters for print method passed through
`normalized`	results gets normalized before plotting. Normalization is done by z-transformation applied on each variable.
`which.plot`	currently the only plot. Plot of cluster centers.
`measure`	cluster validity measure to be considered for which.plot equals "partMeans"

Details

The compositional data set is either internally represented by orthonormal coordiantes before a cluster algorithm is applied, or - depending on the choice of parameters - the Aitchison distance is used.

Value

all relevant information such as cluster centers, cluster memberships, and cluster statistics.

Author(s)

Matthias Templ (accessing the basic features of hclust, Mclust, kmeans, etc. that are all written by others)

References

M. Templ, P. Filzmoser, C. Reimann. Cluster analysis applied to regional geochemical data: Problems and possibilities. *Applied Geochemistry*, **23** (8), 2198–2213, 2008

Templ, M., Filzmoser, P., Reimann, C. (2008) *Cluster analysis applied to regional geochemical data: Problems and possibilities*, Applied Geochemistry, 23 (2008), pages 2198 - 2213.

Examples

```
data(expenditures)
x <- expenditures
rr <- clustCoDa(x, k=6, scale = "robust", transformation = "pivotCoord")
rr2 <- clustCoDa(x, k=6, distMethod = "Aitchison", scale = "none",
                 transformation = "identity")
rr3 <- clustCoDa(x, k=6, distMethod = "Aitchison", method = "single",
                 transformation = "identity", scale = "none")

## Not run:
require(reshape2)
```

```
plot(rr)
plot(rr, normalized = TRUE)
plot(rr, normalized = TRUE, which.plot = "partMeans")

## End(Not run)
```

`clustCoDa_qmode` *Q-mode cluster analysis for compositional parts*

Description

Clustering using the variation matrix of compositional parts

Usage

```
clustCoDa_qmode(x, method = "ward.D2")
```

Arguments

`x`	compositional data represented as a data.frame
`method`	hclust method

Value

a hclust object

Author(s)

Matthias Templ (accessing the basic features of hclust that are all written by other authors)

References

Filzmoser, P., Hron, K. Templ, M. (2018) *Applied Compositional Data Analysis*, Springer, Cham.

Examples

```
data(expenditures)
x <- expenditures
cl <- clustCoDa_qmode(x)
## Not run:
require(reshape2)
plot(cl)
cl2 <- clustCoDa_qmode(x, method = "single")
plot(cl2)

## End(Not run)
```

`coffee` *coffee data set*

Description

30 commercially available coffee samples of different origins.

Usage

```
data(coffee)
```

Format

A data frame with 30 observations and 7 variables.

Details

- `sort` sort of coffee
- `acit` acetic acid
- `metpyr` methylpyrazine
- `furfu` furfural
- `furfualc` furfuryl alcohol
- `dimeth` 2,6 dimethylpyrazine
- `met5` 5-methylfurfural

In the original data set, 15 volatile compounds (descriptors of coffee aroma) were selected for a statistical analysis. We selected six compounds (compositional parts) on three sorts of coffee.

Author(s)

Matthias Templ <matthias.templ@tuwien.ac.at>, Karel Hron

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)
str(coffee)
summary(coffee)
```

`compareMahal` *Compares Mahalanobis distances from two approaches*

Description

Mahalanobis distances are calculated for each zero pattern. Two approaches are used. The first one estimates Mahalanobis distance for observations belonging to one each zero pattern each. The second method uses a more sophisticated approach described below.

Usage

```
compareMahal(x, imp = "KNNa")

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

Arguments

`x`	data frame or matrix
`imp`	imputation method
`y`	unused second argument for the plot method
`...`	additional arguments for plotting passed through

Value

`df`	a data.frame containing the Mahalanobis distances from the estimation in subgroups, the Mahalanobis distances from the imputation and covariance approach, an indicator specifiying outliers and an indicator specifying the zero pattern
`df2`	a groupwise statistics.

Author(s)

Matthias Templ, Karel Hron

References

Templ, M., Hron, K., Filzmoser, P. (2017) Exploratory tools for outlier detection in compositional data with structural zeros". *Journal of Applied Statistics*, **44** (4), 734–752

See Also

`impKNNa`, `pivotCoord`

Examples

```
data(arcticLake)
# generate some zeros
arcticLake[1:10, 1] <- 0
arcticLake[11:20, 2] <- 0
m <- compareMahal(arcticLake)
plot(m)
```

---

compositionalSpline *Compositional spline*

Description

This code implements the compositional smoothing splines grounded on the theory of Bayes spaces.

Usage

```
compositionalSpline(
  t,
  clrf,
  knots,
  w,
  order,
  der,
  alpha,
  spline.plot = FALSE,
  basis.plot = FALSE
)
```

Arguments

`t`	class midpoints
`clrf`	clr transformed values at class midpoints, i.e., fcenLR(f(t))
`knots`	sequence of knots
`w`	weights
`order`	order of the spline (i.e., degree + 1)
`der`	lth derivation
`alpha`	smoothing parameter
`spline.plot`	if TRUE, the resulting spline is plotted
`basis.plot`	if TRUE, the ZB-spline basis system is plotted

Details

The compositional splines enable to construct a spline basis in the centred logratio (clr) space of density functions (ZB-spline basis) and consequently also in the original space of densities (CB-spline basis).The resulting compositional splines in the clr space as well as the ZB-spline basis satisfy the zero integral constraint. This enables to work with compositional splines consistently in the framework of the Bayes space methodology.

Augmented knot sequence is obtained from the original knots by adding #(order-1) multiple endpoints.

Value

`J`	value of the functional J
`ZB_coef`	ZB-spline basis coeffcients
`CV`	score of cross-validation
`GCV`	score of generalized cross-validation

Author(s)

J. Machalova <jitka.machalova@upol.cz>, R. Talska <talskarenata@seznam.cz>

References

Machalova, J., Talska, R., Hron, K. Gaba, A. Compositional splines for representation of density functions. *Comput Stat* (2020). https://doi.org/10.1007/s00180-020-01042-7

Examples

```
# Example (Iris data):
SepalLengthCm <- iris$Sepal.Length
Species <- iris$Species
iris1 <- SepalLengthCm[iris$Species==levels(iris$Species)[1]]
h1 <- hist(iris1, plot = FALSE)
midx1 <- h1$mids
midy1 <- matrix(h1$density, nrow=1, ncol = length(h1$density), byrow=TRUE)
clrf  <- cenLR(rbind(midy1,midy1))$x.clr[1,]
knots <- seq(min(h1$breaks),max(h1$breaks),l=5)
order <- 4
der <- 2
alpha <- 0.99

sol1 <- compositionalSpline(t = midx1, clrf = clrf, knots = knots,
  w = rep(1,length(midx1)), order = order, der = der,
  alpha = alpha, spline.plot = TRUE)
sol1$GCV
ZB_coef <- sol1$ZB_coef
t <- seq(min(knots),max(knots),l=500)
t_step <- diff(t[1:2])
ZB_base <- ZBsplineBasis(t=t,knots,order)$ZBsplineBasis
sol1.t <- ZB_base%*%ZB_coef
```

```
sol2.t <- fcenLRinv(t,t_step,sol1.t)
h2 = hist(iris1,prob=TRUE,las=1)
points(midx1,midy1,pch=16)
lines(t,sol2.t,col="darkred",lwd=2)
# Example (normal distrubution):
# generate n values from normal distribution
set.seed(1)
n = 1000; mean = 0; sd = 1.5
raw_data = rnorm(n,mean,sd)

# number of classes according to Sturges rule
n.class = round(1+1.43*log(n),0)

# Interval midpoints
parnition = seq(-5,5,length=(n.class+1))
t.mid = c(); for (i in 1:n.class){t.mid[i]=(parnition[i+1]+parnition[i])/2}

counts = table(cut(raw_data,parnition))
prob = counts/sum(counts)                # probabilities
dens.raw = prob/diff(parnition)          # raw density data
clrf =  cenLR(rbind(dens.raw,dens.raw))$x.clr[1,]  # raw clr density data

# set the input parameters for smoothing
knots = seq(min(parnition),max(parnition),l=5)
w = rep(1,length(clrf))
order = 4
der = 2
alpha = 0.5
spline = compositionalSpline(t = t.mid, clrf = clrf, knots = knots,
  w = w, order = order, der = der, alpha = alpha,
  spline.plot=TRUE, basis.plot=FALSE)

# ZB-spline coefficients
ZB_coef = spline$ZB_coef

# ZB-spline basis evaluated on the grid "t.fine"
t.fine = seq(min(knots),max(knots),l=1000)
ZB_base = ZBsplineBasis(t=t.fine,knots,order)$ZBsplineBasis

# Compositional spline in the clr space (evaluated on the grid t.fine)
comp.spline.clr = ZB_base%*%ZB_coef

# Compositional spline in the Bayes space (evaluated on the grid t.fine)
comp.spline = fcenLRinv(t.fine,diff(t.fine)[1:2],comp.spline.clr)

# Unit-integral representation of truncated true normal density function
dens.true = dnorm(t.fine, mean, sd)/trapzc(diff(t.fine)[1:2],dnorm(t.fine, mean, sd))

# Plot of compositional spline together with raw density data
matplot(t.fine,comp.spline,type="l",
    lty=1, las=1, col="darkblue", xlab="t",
    ylab="density",lwd=2,cex.axis=1.2,cex.lab=1.2,ylim=c(0,0.28))
matpoints(t.mid,dens.raw,pch = 8, col="darkblue", cex=1.3)
```

```
# Add true normal density function
matlines(t.fine,dens.true,col="darkred",lwd=2)
```

`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, na.rm = TRUE)
```

Arguments

`x`	multivariate data ideally of class data.frame or matrix
`const`	constant, the default equals 1.
`na.rm`	removing missing values.

Value

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

Author(s)

Matthias Templ

Examples

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

`coord` *Coordinate representation of compositional tables*

Description

General approach to orthonormal coordinates for compositional tables

Usage

```
coord(x, SBPr, SBPc)

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

Arguments

`x`	an object of class "table", "data.frame" or "matrix"
`SBPr`	sequential binary partition for rows
`SBPc`	sequential binary partition for columns
`...`	further arguments passed to the print function

Details

A contingency or propability table can be considered as a two-factor composition, we refer to compositional tables. This function constructs orthonomal coordinates for compositional tables using the balances approach for given sequential binary partitions on rows and columns of the compositional table.

Value

Row and column balances and odds ratios as coordinate representations of the independence and interaction tables, respectively.

`row_balances`	row balances
`row_bin`	binary partition for rows
`col_balances`	column balances
`col_bin`	binary parition for columns
`odds_ratios_coord`	odds ratio coordinates

Author(s)

Kamila Facevicova, and minor adaption by Matthias Templ

References

Facevicova, K., Hron, K., Todorov, V., Templ, M. (2018) General approach to coordinate representation of compositional tables. *Scandinavian Journal of Statistics*, 45(4), 879-899.

Examples

```
x <- rbind(c(1,5,3,6,8,4),c(6,4,9,5,8,12),c(15,2,68,42,11,6),
           c(20,15,4,6,23,8),c(11,20,35,26,44,8))
x
SBPc <- rbind(c(1,1,1,1,-1,-1),c(1,-1,-1,-1,0,0),c(0,1,1,-1,0,0),
              c(0,1,-1,0,0,0),c(0,0,0,0,1,-1))
SBPc
SBPr <- rbind(c(1,1,1,-1,-1),c(1,1,-1,0,0),c(1,-1,0,0,0),c(0,0,0,1,-1))
SBPr
result <- coord(x, SBPr,SBPc)
result
data(socExp)
```

---

corCoDa *Correlations for compositional data*

Description

This function computes correlation coefficients between compositional parts based on symmetric pivot coordinates.

Usage

```
corCoDa(x, ...)
```

Arguments

`x`	a matrix or data frame with compositional data
`...`	additional arguments for the function `cor`

Value

A compositional correlation matrix.

Author(s)

Petra Kynclova

References

Kynclova, P., Hron, K., Filzmoser, P. (2017) Correlation between compositional parts based on symmetric balances. *Mathematical Geosciences*, 49(6), 777-796.

Examples

```
data(expenditures)
corCoDa(expenditures)
x <- arcticLake
corCoDa(x)
```

`cubeCoord` *Coordinate representation of a compositional cube and of a sample of compositional cubes*

Description

cubeCoord computes a system of orthonormal coordinates of a compositional cube. Computation of either pivot coordinates or a coordinate system based on the given SBP is possible.

Wrapper (cubeCoordWrapper): For each compositional cube in the sample cubeCoordWrapper computes a system of orthonormal coordinates and provide a simple descriptive analysis. Computation of either pivot coordinates or a coordinate system based on the given SBP is possible.

Usage

```
cubeCoord(
  x,
  row.factor = NULL,
  col.factor = NULL,
  slice.factor = NULL,
  value = NULL,
  SBPr = NULL,
  SBPc = NULL,
  SBPs = NULL,
  pivot = FALSE,
  print.res = FALSE
)

cubeCoordWrapper(
  X,
  obs.ID = NULL,
  row.factor = NULL,
  col.factor = NULL,
  slice.factor = NULL,
  value = NULL,
  SBPr = NULL,
  SBPc = NULL,
  SBPs = NULL,
  pivot = FALSE,
  test = FALSE,
  n.boot = 1000
)
```

Arguments

`x` — a data frame containing variables representing row, column and slice factors of the respective compositional cube and variable with the values of the composition.

`row.factor` — name of the variable representing the row factor. Needs to be stated with the quotation marks.

`col.factor` — name of the variable representing the column factor. Needs to be stated with the quotation marks.

`slice.factor` — name of the variable representing the slice factor. Needs to be stated with the quotation marks.

`value` — name of the variable representing the values of the composition. Needs to be stated with the quotation marks.

`SBPr` — an $I - 1 \times I$ array defining the sequential binary partition of the values of the row factor, where I is the number of the row factor levels. The values assigned in the given step to the + group are marked by 1, values from the - group by -1 and the rest by 0. If it is not provided, the pivot version of coordinates is constructed automatically.

`SBPc` — an $J - 1 \times J$ array defining the sequential binary partition of the values of the column factor, where J is the number of the column factor levels. The values assigned in the given step to the + group are marked by 1, values from the - group by -1 and the rest by 0. If it is not provided, the pivot version of coordinates is constructed automatically.

`SBPs` — an $K - 1 \times K$ array defining the sequential binary partition of the values of the slice factor, where K is the number of the slice factor levels. The values assigned in the given step to the + group are marked by 1, values from the - group by -1 and the rest by 0. If it is not provided, the pivot version of coordinates is constructed automatically.

`pivot` — logical, default is FALSE. If TRUE, or one of the SBPs is not defined, its pivot version is used.

`print.res` — logical, default is FALSE. If TRUE, the output is displayed in the Console.

`X` — a data frame containing variables representing row, column and slice factors of the respective compositional cubes, variable with the values of the composition and variable distinguishing the observations.

`obs.ID` — name of the variable distinguishing the observations. Needs to be stated with the quotation marks.

`test` — logical, default is FALSE. If TRUE, the bootstrap analysis of coordinates is provided.

`n.boot` — number of bootstrap samples.

Details

cubeCoord

This transformation moves the IJK-part compositional cubes from the simplex into a (IJK-1)-dimensional real space isometrically with respect to its three-factorial nature.

Wrapper (cubeCoordWrapper): Each of n IJK-part compositional cubes from the sample is with respect to its three-factorial nature isometrically transformed from the simplex into a (IJK-1)-dimensional real space. Sample mean values and standard deviations are computed and using bootstrap an estimate of 95 % confidence interval is given.

Value

`Coordinates` an array of orthonormal coordinates.

`Grap.rep` graphical representation of the coordinates. Parts denoted by + form the groups in the numerator of the respective computational formula, parts - form the denominator and parts . are not involved in the given coordinate.

`Row.balances` an array of row balances.

`Column.balances` an array of column balances.

`Slice.balances` an array of slice balances.

`Row.column.OR` an array of row-column OR coordinates.

`Row.slice.OR` an array of row-slice OR coordinates.

`Column.slice.OR` an array of column-slice OR coordinates.

`Row.col.slice.OR` an array of coordinates describing the mutual interaction between all three factors.

`Contrast.matrix` contrast matrix.

`Log.ratios` an array of pure log-ratios between groups of parts without the normalizing constant.

`Coda.cube` cube form of the given composition.

`Bootstrap` array of sample means, standard deviations and bootstrap confidence intervals.

`Cubes` Cube form of the given compositions.

Author(s)

Kamila Facevicova

References

Facevicova, K., Filzmoser, P. and K. Hron (2019) Compositional Cubes: Three-factorial Compositional Data. Under review.

See Also

`tabCoord` `tabCoordWrapper`

Examples

```
###################
### Coordinate representation of a CoDa Cube
## Not run:
### example from Fa\v cevicov\'a (2019)
data(employment2)
CZE <- employment2[which(employment2$Country == 'CZE'), ]

# pivot coordinates
cubeCoord(CZE, "Sex", 'Contract', "Age", 'Value')

# coordinates with given SBP

r <- t(c(1,-1))
c <- t(c(1,-1))
s <- rbind(c(1,-1,-1), c(0,1,-1))

cubeCoord(CZE, "Sex", 'Contract', "Age", 'Value', r,c,s)

## End(Not run)

###################
### Analysis of a sample of CoDa Cubes
## Not run:
### example from Fa\v cevicov\'a (2019)
data(employment2)
### Compositional tables approach,
### analysis of the relative structure.
### An example from Facevi\v cov\'a (2019)

# pivot coordinates
cubeCoordWrapper(employment2, 'Country', 'Sex', 'Contract', 'Age', 'Value',
test=TRUE)

# coordinates with given SBP (defined in the paper)

r <- t(c(1,-1))
c <- t(c(1,-1))
s <- rbind(c(1,-1,-1), c(0,1,-1))

res <- cubeCoordWrapper(employment2, 'Country', 'Sex', 'Contract',
"Age", 'Value', r,c,s, test=TRUE)

### Classical approach,
### generalized linear mixed effect model.

library(lme4)
employment2$y <- round(employment2$Value*1000)
glmer(y~Sex*Age*Contract+(1|Country),data=employment2,family=poisson)

### other relations within cube (in the log-ratio form)
### e.g. ratio between women and man in the group FT, 15to24
```

```
### and ratio between age groups 15to24 and 55plus

# transformation matrix
T <- rbind(c(1,rep(0,5), -1, rep(0,5)), c(rep(c(1/4,0,-1/4), 4)))
T %*% t(res$Contrast.matrix) %*%res$Bootstrap[,1]

## End(Not run)
```

---

daCoDa *Linear and quadratic discriminant analysis for compositional data.*

Description

Linear and quadratic discriminant analysis for compositional data using either robust or classical estimation.

Usage

```
daCoDa(x, grp, coda = TRUE, method = "classical", rule = "linear", ...)
```

Arguments

`x`	a matrix or data frame containing the explanatory variables
`grp`	grouping variable: a factor specifying the class for each observation.
`coda`	TRUE, when the underlying data are compositions.
`method`	“classical” or “robust”
`rule`	a character, either “linear” (the default) or “quadratic”.
`...`	additional arguments for the functions passed through

Details

Compositional data are expressed in orthonormal (ilr) coordinates (if `coda==TRUE`). For linear discriminant analysis the functions LdaClassic (classical) and Linda (robust) from the package rrcov are used. Similarly, quadratic discriminant analysis uses the functions QdaClassic and QdaCov (robust) from the same package.

The classical linear and quadratic discriminant rules are invariant to ilr coordinates and clr coefficients. The robust rules are invariant to ilr transformations if affine equivariant robust estimators of location and covariance are taken.

Value

An S4 object of class LdaClassic, Linda, QdaClassic or QdaCov. See package rrcov for details.

Author(s)

Jutta Gamper

References

Filzmoser, P., Hron, K., Templ, M. (2012) Discriminant analysis for compositional data and robust parameter estimation. *Computational Statistics*, 27(4), 585-604.

See Also

LdaClassic, Linda, QdaClassic, QdaCov

Examples

```
## toy data (non-compositional)
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))

clas1 <- daCoDa(X, grp, coda=FALSE, method = "classical", rule="linear")
summary(clas1)
## predict runs only with newest verison of rrcov
## Not run:
predict(clas1)

## End(Not run)
# specify different prior probabilities
clas2 <- daCoDa(X, grp, coda=FALSE, prior=c(1/3, 1/3, 1/3))
summary(clas2)

## compositional data
data(coffee)
x <- coffee[coffee$sort!="robusta",2:7]
group <- droplevels(coffee$sort[coffee$sort!="robusta"])
cof.cla <- daCoDa(x, group, method="classical", rule="quadratic")
cof.rob <- daCoDa(x, group, method="robust", rule="quadratic")
## predict runs only with newest verison of rrcov
## Not run:
predict(cof.cla)@ct
predict(cof.rob)@ct

## End(Not run)
```

---

daFisher *Discriminant analysis by Fisher Rule.*

Description

Discriminant analysis by Fishers rule using the logratio approach to compositional data.

Usage

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

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

## S3 method for class 'daFisher'
predict(object, ..., newdata)

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

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” estimation.
`plotScore`	TRUE, if the scores should be plotted automatically.
`...`	additional arguments for the print method passed through
`object`	object of class “daFisher”
`newdata`	new data in the appropriate form (CoDa, etc)

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 coordinates and clr coefficients. 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
`scores`	fisher scores
`mc`	table indicating misclassifications
`mcrate`	misclassification rate

coda	coda
grp	grouping
grppred	predicted groups
xc	xc
meanj	meanj
cv	cv
pj	pj
meanov	meanov
fdiscr	fdiscr

Author(s)

Peter Filzmoser, Matthias Templ.

References

Filzmoser, P. and Hron, K. and Templ, M. (2012) Discriminant analysis for compositional data and robust parameter estimation. *Computational Statistics*, 27(4), 585-604.

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

```
## toy data (non-compositional)
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
summary(d2)
predict(d2, newdata = X)

## example with olive data:
## Not run:
data(olive, package = "RnavGraph")
# exclude zeros (alternatively impute them if
```

```
# the detection limit is known using impRZilr())
ind <- which(olive == 0, arr.ind = TRUE)[,1]
olives <- olive[-ind, ]
x <- olives[, 4:10]
grp <- olives$Region # 3 groups
res <- daFisher(x,grp)
res
summary(res)
res <- daFisher(x, grp, plotScore = TRUE)
res <- daFisher(x, grp, method = "robust")
res
summary(res)
predict(res, newdata = x)
res <- daFisher(x,grp, plotScore = TRUE, method = "robust")

# 9 regions
grp <- olives$Area
res <- daFisher(x, grp, plotScore = TRUE)
res
summary(res)
predict(res, newdata = x)

## End(Not run)
```

`economy` *economic indicators*

Description

Household and government consumptions, gross captial formation and import and exports of goods and services.

Usage

```
data(economy)
```

Format

A data frame with 30 observations and 7 variables

Details

`country` country name

`country2` country name, short version

`HHconsumption` Household and NPISH final consumption expenditure

`GOVconsumption` Final consumption expenditure of general government

`capital` Gross capital formation

`exports` Exports of goods and services

`imports` Imports of goods and services

Author(s)

Peter Filzmoser, Matthias Templ <matthias.templ@tuwien.ac.at>

References

Eurostat, https://ec.europa.eu/eurostat/data

Examples

```
data(economy)
str(economy)
```

`educFM` *education level of father (F) and mother (M)*

Description

Education level of father (F) and mother (M) in percentages of low (l), medium (m), and high (h) of 31 countries in Europe.

Usage

```
data(educFM)
```

Format

A data frame with 31 observations and 8 variables

Details

- `country` community code
- `F.l` percentage of females with low edcuation level
- `F.m` percentage of females with medium edcuation level
- `F.h` percentage of females with high edcuation level
- `F.l` percentage of males with low edcuation level
- `F.m` percentage of males with medium edcuation level
- `F.h` percentage of males with high edcuation level

Author(s)

Peter Filzmoser, Matthias Templ

Source

from Eurostat,https://ec.europa.eu/eurostat/

Examples

```
data(educFM)
str(educFM)
```

`efsa` *efsa nutrition consumption*

Description

Comprehensive European Food Consumption Database

Format

A data frame with 87 observations on the following 22 variables.

`Country` country name

`Pop.Class` population class

`grains` Grains and grain-based products

`vegetables` Vegetables and vegetable products (including fungi)

`roots` Starchy roots and tubers

`nuts` Legumes, nuts and oilseeds

`fruit` Fruit and fruit products

`meat` Meat and meat products (including edible offal)

`fish` Fish and other seafood (including amphibians, rept)

`milk` Milk and dairy products

`eggs` Eggs and egg products

`sugar` Sugar and confectionary

`fat` Animal and vegetable fats and oils

`juices` Fruit and vegetable juice

`nonalcoholic` Non-alcoholic beverages (excepting milk based beverages)

`alcoholic` Alcoholic beverages

`water` Drinking water (water without any additives)

`herbs` Herbs, spices and condiments

`small_children_food` Food for infants and small children

`special` Products for special nutritional use

`composite` Composite food (including frozen products)

`snacks` Snacks, desserts, and other foods

Details

The Comprehensive Food Consumption Database is a source of information on food consumption across the European Union (EU). The food consumption are reported in grams per day (g/day).

Source

efsa

Examples

```
data(efsa)
```

`election` *election data*

Description

Results of a election in Germany 2013 in different federal states

Usage

```
data(election)
```

Format

A data frame with 16 observations and 8 variables

Details

Votes for the political parties in the elections (compositional variables), and their relation to the unemployment rate and the average monthly income (external non-compositional variables). Votes are for the Christian Democratic Union and Christian Social Union of Bavaria, also called The Union (CDU/CSU), Social Democratic Party (SDP), The Left (DIE LINKE), Alliance '90/The Greens (GRUNE), Free Democratic Party (FDP) and the rest of the parties participated in the elections (other parties). The votes are examined in absolute values (number of valid votes). The unemployment in the federal states is reported in percentages, and the average monthly income in Euros.

- `CDU_CSU` Christian Democratic Union and Christian Social Union of Bavaria, also called The Union
- `SDP` Social Democratic Party
- `GRUENE` Alliance '90/The Greens
- `FDP` Free Democratic Party
- `DIE_LINKE` The Left
- `other_parties` Votes for the rest of the parties participated in the elections
- `unemployment` Unemployment in the federal states in percentages
- `income` Average monthly income in Euros

Author(s)

Petra Klynclova, Matthias Templ

Source

German Federal Statistical Office

References

Eurostat, https://ec.europa.eu/eurostat/data

Examples

```
data(election)
str(election)
```

`electionATbp` *Austrian presidential election data*

Description

Results the Austrian presidential election in October 2016.

Usage

```
data(electionATbp)
```

Format

A data frame with 2202 observations and 10 variables

Details

Votes for the candidates Hofer and Van der Bellen.

`GKZ` Community code

`Name` Name of the community

`Eligible` eligible votes

`Votes_total` total votes

`Votes_invalid` invalid votes

`Votes_valid` valid votes

`Hofer_total` votes for Hofer

`Hofer_perc` votes for Hofer in percentages

`VanderBellen_total` votes for Van der Bellen

`VanderBellen_perc` votes for Van der Bellen in percentages

Author(s)

Peter Filzmoser

Source

OpenData Austria, https://www.data.gv.at/

Examples

```
data(electionATbp)
str(electionATbp)
```

`employment` *employment in different countries by gender and status.*

Description

employment in different countries by gender and status.

Usage

```
data(employment)
```

Format

A three-dimensional table

Examples

```
data(employment)
str(employment)
employment
```

`employment2` *Employment in different countries by Sex, Age, Contract, Value*

Description

Estimated number of employees in 42 countries in 2015, distributed according to gender (Women/Men), age (15-24, 25-54, 55+) and type of contract (Full- and part-time).

Usage

```
data(employment2)
```

Format

A data.frame with 504 rows and 5 columns.

Details

For each country in the sample, an estimated number of employees in the year 2015 was available, divided according to gender and age of employees and the type of the contract. The data form a sample of 42 cubes with two rows (gender), two columns (type) of contract) and three slices (age), which allow for a deeper analysis of the overall employment structure, not just from the perspective of each factor separately, but also from the perspective of the relations/interactions between them. Thorough analysis of the sample is described in Facevicova (2019).

`Country` Country

`Sex` gender, males (M) and females (F)

`Age` age class, young (category 15 - 24), middle-aged (25 - 54) and older (55+) employees

`Contract` factor, defining the type of contract, full-time (FT) and part-time (PT) contracts

`Value` Number of employees in the given category (in thousands)

Author(s)

Kamila Facevicova

Source

https://stats.oecd.org

References

Facevicova, K., Filzmoser, P. and K. Hron (2019) Compositional Cubes: Three-factorial Compositional Data. Under review.

Examples

```
data(employment2)
head(employment2)
```

`employment_df` *Employment in different countries by gender and status.*

Description

`gender` factor

`status` factor, defining if part or full time work

`country` country

`value` employment

Usage

```
data(employment_df)
```

Format

A data.frame with 132 rows and 4 columns.

Examples

```
data(employment_df)
head(employment_df)
```

---

expenditures *synthetic household expenditures toy data set*

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 on the following 5 variables.

Details

`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)

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.

Author(s)

Matthias Templ <`matthias.templ@tuwien.ac.at`>, Karel Hron

References

Aitchison, J. (1986) *The Statistical Analysis of Compositional Data* Monographs on Statistics and Applied Probability. Chapman and 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.

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

Examples

```
data(expendituresEU)
```

fBPUpChi_PLS *Title*

Description

Title

Usage

```
fBPUpChi_PLS(Yp, r3, b, version = "cov")
```

Arguments

Yp	a matrix of raw compositional data with "n" rows and "D" columns/components
r3	a response variable; can be continuous (PLS regression) or binary (PLS-DA)
b	a given balance constructed during the procedure (contains some zero value(s))
version	a parameter determining whether the balances are ordered according to max. covariance (default) or max. correlation

Value

A list with the following components:

`bal`

`varbal`

fcenLR *fcenLR transformation (functional)*

Description

fcenLR[lambda] transformation: mapping from B^2(lambda) into L^2(lambda)

Usage

```
fcenLR(z, z_step, density)
```

Arguments

z	grid of points defining the abscissa
z_step	step of the grid of the abscissa
density	grid evaluation of the lambda-density

Value

`out` grid evaluation of the lambda-density in L^2(lambda)

Author(s)

R. Talska<talskarenata@seznam.cz>, A. Menafoglio, K. Hron<karel.hron@upol.cz>, J. J. Egozcue, J. Palarea-Albaladejo

References

Talska, R., Menafoglio, A., Hron, K., Egozcue, J. J., Palarea-Albaladejo, J. (2020). Weighting the domain of probability densities in functional data analysis.*Stat*(2020). https://doi.org/10.1002/sta4.283

Examples

```
# Example (normal density)
t = seq(-4.7,4.7, length = 1000)
t_step = diff(t[1:2])

mean = 0; sd = 1.5
f = dnorm(t, mean, sd)
f1 = f/trapzc(t_step,f)

f.fcenLR = fcenLR(t,t_step,f)
f.fcenLRinv = fcenLRinv(t.fine,t_step,f.fcenLR)

plot(t,f.fcenLR, type="l",las=1, ylab="fcenLR(density)",
  cex.lab=1.2,cex.axis=1.2, col="darkblue",lwd=2)
abline(h=0, col="red")

plot(t,f.fcenLRinv, type="l",las=1,
  ylab="density",cex.lab=1.2,cex.axis=1.2, col="darkblue",lwd=2,lty=1)
lines(t,f1,lty=2,lwd=2,col="gold")
```

---

fcenLRinv *Inverse of fcenLR transformations (functional)*

Description

Inverse of fcenLR transformations

Usage

```
fcenLRinv(z, z_step, fcenLR, k = 1)
```

Arguments

z	grid of points defining the abscissa
z_step	step of the grid of the abscissa
fcenLR	grid evaluation of (i) fcenLR[lambda] transformed lambda-density, (ii) fcenLR[u] transformed P-density, (iii) fcenLR[P] transformed P-density
k	value of the integral of density; if k=1 it returns a unit-integral representation of density

Details

By default, it returns a unit-integral representation of density.

Value

out ... grid evaluation of (i) lambda-density in B2(lambda), (ii) P-density in unweighted B2(lambda), (iii) P-density in B2(P)

Author(s)

R. Talska<talskarenata@seznam.cz>, A. Menafoglio, K. Hron<karel.hron@upol.cz>, J. J. Egozcue, J. Palarea-Albaladejo

Examples

```
# Example (normal density)
t = seq(-4.7,4.7, length = 1000)
t_step = diff(t[1:2])

mean = 0; sd = 1.5
f = dnorm(t, mean, sd)
f1 = f/trapzc(t_step,f)

f.fcenLR = fcenLR(t,t_step,f)
f.fcenLRinv = fcenLRinv(t.fine,t_step,f.fcenLR)

plot(t,f.fcenLR, type="l",las=1, ylab="fcenLR(density)",
  cex.lab=1.2,cex.axis=1.2, col="darkblue",lwd=2)
abline(h=0, col="red")

plot(t,f.fcenLRinv, type="l",las=1,
  ylab="density",cex.lab=1.2,cex.axis=1.2, col="darkblue",lwd=2,lty=1)
lines(t,f1,lty=2,lwd=2,col="gold")
```

fcenLRp *fcenLRp transformation (functional)*

Description

fcenLR[P] transformation: mapping from B2(P) into L2(P)

Usage

```
fcenLRp(z, z_step, density, p)
```

Arguments

`z`	grid of points defining the abscissa
`z_step`	step of the grid of the abscissa
`density`	grid evaluation of the P-density
`p`	density of the reference measure P

Value

`out`	grid evaluation of the P-density in L2(P)

Author(s)

R. Talska<`talskarenata@seznam.cz`>, A. Menafoglio, K. Hron<`karel.hron@upol.cz`>, J.J. Egozcue, J. Palarea-Albaladejo

References

Talska, R., Menafoglio, A., Hron, K., Egozcue, J. J., Palarea-Albaladejo, J. (2020). Weighting the domain of probability densities in functional data analysis.*Stat*(2020). https://doi.org/10.1002/sta4.283

fcenLRu *fcenLRu transformation (functional)*

Description

fcenLR[u] transformation: mapping from B2(P) into unweigted L2(lambda)

Usage

```
fcenLRu(z, z_step, density, p)
```

Arguments

z	grid of points defining the abscissa
z_step	step of the grid of the abscissa
density	grid evaluation of the P-density
p	density of the reference measure P

Value

out	grid evaluation of the P-density in unweighted L2(lambda)

Author(s)

R. Talska<talskarenata@seznam.cz>, A. Menafoglio, K. Hron<karel.hron@upol.cz>, J. J. Egozcue, J. Palarea-Albaladejo

References

Talska, R., Menafoglio, A., Hron, K., Egozcue, J. J., Palarea-Albaladejo, J. (2020). Weighting the domain of probability densities in functional data analysis.*Stat*(2020). https://doi.org/10.1002/sta4.283

Examples

```
# Common example for all transformations - fcenLR, fcenLRp, fcenLRu
# Example (log normal distribution under the reference P)
t = seq(1,10, length = 1000)
t_step = diff(t[1:2])

# Log normal density w.r.t. Lebesgue reference measure in B2(lambda)
f = dlnorm(t, meanlog = 1.5, sdlog = 0.5)

# Log normal density w.r.t. Lebesgue reference measure in L2(lambda)
f.fcenLR = fcenLR(t,t_step,f)

# New reference given by exponential density
p = dexp(t,0.25)/trapzc(t_step,dexp(t,0.25))

# Plot of log normal density w.r.t. Lebesgue reference measure
# in B2(lambda) together with the new reference density p
matplot(t,f,type="l",las=1, ylab="density",cex.lab=1.2,cex.axis=1.2,
  col="black",lwd=2,ylim=c(0,0.3),xlab="t")
matlines(t,p,col="blue")
text(2,0.25,"p",col="blue")
text(4,0.22,"f",col="black")

# Log-normal density w.r.t. exponential distribution in B2(P)
# (unit-integral representation)
fp = (f/p)/trapzc(t_step,f/p)

# Log-normal density w.r.t. exponential distribution in L2(P)
fp.fcenLRp = fcenLRp(t,t_step,fp,p)
```

```
# Log-normal density w.r.t. exponential distribution in L2(lambda)
fp.fcenLRu = fcenLRu(t,t_step,fp,p)

# Log-normal density w.r.t. exponential distribution in B2(lambda)
fp.u = fcenLRinv(t,t_step,fp.fcenLRu)

# Plot
layout(rbind(c(1,2,3,4),c(7,8,5,6)))
par(cex=1.1)

plot(t, f.fcenLR, type='l', ylab=expression(fcenLR[lambda](f)),
  xlab='t',las=1,ylim=c(-3,3),
 main=expression(bold(atop(paste('(a) Representation of f in ', L^2, (lambda)),'[not weighted]'))))
abline(h=0,col="red")

plot(t, f, type='l', ylab=expression(f[lambda]),
  xlab='t',las=1,ylim=c(0,0.4),
 main=expression(bold(atop(paste('(b) Density f in ', B^2, (lambda)),'[not weighted]'))))

plot(t, fp, type='l', ylab=expression(f[P]), xlab='t',
  las=1,ylim=c(0,0.4),
  main=expression(bold(atop(paste('(c) Density f in ', B^2, (P)),'[weighted with P]'))))

plot(t, fp.fcenLRp, type='l', ylab=expression(fcenLR[P](f[P])),
  xlab='t',las=1,ylim=c(-3,3),
 main=expression(bold(atop(paste('(d) Representation of f in ', L^2, (P)),'[weighted with P]'))))
abline(h=0,col="red")

plot(t, fp.u, type='l', ylab=expression(paste(omega^(-1),(f[P]))),
  xlab='t',las=1,ylim=c(0,0.4),
 main=expression(bold(atop(paste('(e) Representation of f in ', B^2, (lambda)),'[unweighted]'))))

plot(t, fp.fcenLRu, type='l', ylab=expression(paste(fcenLR[u](f[P]))),
  xlab='t',las=1,ylim=c(-3,3),
 main=expression(bold(atop(paste('(f) Representation of f in ', L^2, (lambda)),'[unweighted]'))))
abline(h=0,col="red")
```

`foodbalance` *country food balances*

Description

Food balance in each country (2018)

Format

A data frame with 115 observations on the following 116 variables.

`country` Country name

`Cereals - Excluding Beer` Food balance on cereals
`Wheat and products` Wheat-based products
`Rice and products` Rice and rice-based products
`Barley and products` Barley and barley-based products
`Maize and products` Maize and maize-based products
`Rye and products` Rye and rye-based products
`Oats` Oats
`Millet and products` Millet and millet-based products
`Cereals, Other` Other cereals
`Starchy Roots` Starchy roots group
`Cassava and products` Cassava and derivatives
`Potatoes and products` Potatoes and related products
`Sweet potatoes` Sweet potatoes
`Roots, Other` Other root crops
`Sugar Crops` Sugar crops group
`Sugar cane` Sugar cane
`Sugar & Sweeteners` Sugar and sweeteners group
`Sugar (Raw Equivalent)` Raw equivalent sugar content
`Sweeteners, Other` Other sweeteners
`Honey` Honey
`Pulses` Pulses group
`Beans` Beans
`Peas` Peas
`Pulses, Other and products` Other pulses and products
`Treenuts` Tree nuts group
`Nuts and products` Nuts and their products
`Oilcrops` Oilcrops group
`Soyabeans` Soybeans
`Groundnuts` Groundnuts
`Rape and Mustardseed` Rape and mustard seed
`Coconuts - Incl Copra` Coconuts including copra
`Sesame seed` Sesame seeds
`Olives (including preserved)` Olives including preserved
`Vegetable Oils` Vegetable oils group
`Soyabean Oil` Soybean oil
`Groundnut Oil` Groundnut oil
`Sunflowerseed Oil` Sunflower seed oil

`Rape and Mustard Oil` Rape and mustard oil

`Cottonseed Oil` Cottonseed oil

`Palmkernel Oil` Palm kernel oil

`Palm Oil` Palm oil

`Coconut Oil` Coconut oil

`Sesameseed Oil` Sesame seed oil

`Olive Oil` Olive oil

`Ricebran Oil` Rice bran oil

`Maize Germ Oil` Maize germ oil

`Oilcrops Oil, Other` Other oilcrops oils

`Vegetables` Vegetables group

`Tomatoes and products` Tomatoes and products

`Onions` Onions

`Vegetables, Other` Other vegetables

`Fruits - Excluding Wine` Fruits group, excluding wine

`Oranges, Mandarines` Oranges and mandarins

`Lemons, Limes and products` Lemons, limes and products

`Grapefruit and products` Grapefruits and products

`Citrus, Other` Other citrus fruits

`Bananas` Bananas

`Plantains` Plantains

`Apples and products` Apples and products

`Pineapples and products` Pineapples and products

`Dates` Dates

`Grapes and products (excl wine)` Grapes and non-wine products

`Fruits, Other` Other fruits

`Stimulants` Stimulants group

`Coffee and products` Coffee and products

`Cocoa Beans and products` Cocoa beans and products

`Tea (including mate)` Tea including mate

`Spices` Spices group

`Pepper` Pepper

`Pimento` Pimento

`Cloves` Cloves

`Spices, Other` Other spices

`Alcoholic Beverages` Alcoholic beverages group

`Wine` Wine

`Beer` Beer

`Beverages, Fermented` Fermented beverages

`Beverages, Alcoholic` Alcoholic beverages (other)

`Meat` Meat group

`Bovine Meat` Beef and veal

`Mutton & Goat Meat` Mutton and goat meat

`Pigmeat` Pork

`Poultry Meat` Poultry

`Meat, Other` Other meats

`Offals` Offals group

`Offals, Edible` Edible offals

`Animal fats` Animal fats

`Butter, Ghee` Butter and ghee

`Cream` Cream

`Fats, Animals, Raw` Raw animal fats

`Eggs` Eggs

`Milk - Excluding Butter` Milk excluding butter

`Fish, Seafood` Fish and seafood group

`Freshwater Fish` Freshwater fish

`Miscellaneous` Miscellaneous group

`Infant food` Infant food

`Fish, Body Oil` Fish body oil

`Fish, Liver Oil` Fish liver oil

`Demersal Fish` Demersal fish

`Pelagic Fish` Pelagic fish

`Marine Fish, Other` Other marine fish

`Crustaceans` Crustaceans

`Cephalopods` Cephalopods

`Molluscs, Other` Other molluscs

`Aquatic Products, Other` Other aquatic products

`Aquatic Animals, Others` Other aquatic animals

`Aquatic Plants` Aquatic plants

`Sorghum and products` Sorghum and its products

`Oilcrops, Other` Other oilcrops

`Sugar beet` Sugar beet

`Yams` Yams

`Sunflower seed` Sunflower seed

`Sugar non-centrifugal` Non-centrifugal sugar

`Meat, Aquatic Mammals` Meat from aquatic mammals

`Palm kernels` Palm kernels

`value.Alcohol, Non-Food` Non-food alcohol (value)

Source

https://www.fao.org/home/en/

Examples

```
data(foodbalance)
```

GDPsatis *GDP satisfaction*

Description

Satisfaction of GDP in 31 countries. The GDP is measured per capita from the year 2012.

Usage

```
data(GDPsatis)
```

Format

A data frame with 31 observations and 8 variables

Details

`country` community code

`gdp` GDP per capita in 2012

`very.bad` satisfaction very bad

`bad` satisfaction bad

`moderately.bad` satisfaction moderately bad

`moderately.good` satisfaction moderately good

`good` satisfaction good

`very.good` satisfaction very good

Author(s)

Peter Filzmoser, Matthias Templ

Source

from Eurostat,https://ec.europa.eu/eurostat/

Examples

```
data(GDPsatis)
str(GDPsatis)
```

`gemas` *GEMAS geochemical data set*

Description

Geochemical data set on agricultural and grazing land soil

Usage

```
data(gemas)
```

Format

A data frame with 2108 observations and 30 variables

Details

`COUNTRY` country name

`longitude` longitude in WGS84

`latitude` latitude in WGS84

`Xcoord` UTM zone east

`Ycoord` UTM zone north

`MeanTemp` Annual mean temperature

`AnnPrec` Annual mean precipitation

`soilclass` soil class

`sand` sand

`silt` silt

`clay` clay

`Al` Concentration of aluminum (in mg/kg)

`Ba` Concentration of barium (in mg/kg)

`Ca` Concentration of calzium (in mg/kg)\

`Cr` Concentration of chromium (in mg/kg)

`Fe` Concentration of iron (in mg/kg)

`K` Concentration of pottasium (in mg/kg)

`Mg` Concentration of magnesium (in mg/kg)

`Mn` Concentration of manganese (in mg/kg)

`Na` Concentration of sodium (in mg/kg)

`Nb` Concentration of niobium (in mg/kg)

`Ni` Concentration of nickel (in mg/kg)

`P` Concentration of phosphorus (in mg/kg)

- `Si` Concentration of silicium (in mg/kg)
- `Sr` Concentration of strontium (in mg/kg)
- `Ti` Concentration of titanium (in mg/kg)
- `V` Concentration of vanadium (in mg/kg)\
- `Y` Concentration of yttrium (in mg/kg)
- `Zn` Concentration of zinc (in mg/kg)
- `Zr` Concentration of zirconium (in mg/kg)
- `LOI` Loss on ignition (in wt-percent)

The sampling, at a density of 1 site/2500 sq. km, was completed at the beginning of 2009 by collecting 2211 samples of agricultural soil (Ap-horizon, 0-20 cm, regularly ploughed fields), and 2118 samples from land under permanent grass cover (grazing land soil, 0-10 cm), according to an agreed field protocol. All GEMAS project samples were shipped to Slovakia for sample preparation, where they were air dried, sieved to <2 mm using a nylon screen, homogenised and split to subsamples for analysis. They were analysed for a large number of chemical elements. In this sample, the main elements by X-ray fluorescence are included as well as the composition on sand, silt, clay.

Author(s)

GEMAS is a cooperation project between the EuroGeoSurveys Geochemistry Expert Group and Eurometaux. Integration in R, Peter Filzmoser and Matthias Templ.

References

Reimann, C., Birke, M., Demetriades, A., Filzmoser, P. and O'Connor, P. (Editors), 2014. Chemistry of Europe's agricultural soils - Part A: Methodology and interpretation of the GEMAS data set. Geologisches Jahrbuch (Reihe B 102), Schweizerbarth, Hannover, 528 pp. + DVD Reimann, C., Birke, M., Demetriades, A., Filzmoser, P. & O'Connor, P. (Editors), 2014. Chemistry of Europe's agricultural soils - Part B: General background information and further analysis of the GEMAS data set. Geologisches Jahrbuch (Reihe B 103), Schweizerbarth, Hannover, 352 pp.

Examples

```
data(gemas)
str(gemas)
## sample sites
## Not run:
require(ggmap)
map <- get_map("europe", source = "stamen", maptype = "watercolor", zoom=4)
ggmap(map) + geom_point(aes(x=longitude, y=latitude), data=gemas)
map <- get_map("europe", zoom=4)
ggmap(map) + geom_point(aes(x=longitude, y=latitude), data=gemas, size=0.8)

## End(Not run)
```

`gjovik` *gjovik*

Description

Gjovik geochemical data set

Format

A data frame with 615 observations and 63 variables.

`ID` a numeric vector

`MAT` type of material

`mE32wgs` longitude

`mN32wgs` latitude

`XCOO` X coordinates

`YCOO` Y coordinates

`ALT` altitute

`kmNS` some distance north-south

`kmSN` some distance south-north

`LITHO` lithologies

`Ag` a numeric vector

`Al` a numeric vector

`As` a numeric vector

`Au` 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

`Ce` a numeric vector

`Co` a numeric vector

`Cr` a numeric vector

`Cs` a numeric vector

`Cu` a numeric vector

`Fe` a numeric vector

`Ga` a numeric vector

`Ge` a numeric vector

Hf a numeric vector

Hg a numeric vector

In a numeric vector

K a numeric vector

La a numeric vector

Li a numeric vector

Mg a numeric vector

Mn a numeric vector

Mo a numeric vector

Na a numeric vector

Nb a numeric vector

Ni a numeric vector

P a numeric vector

Pb a numeric vector

Pd a numeric vector

Pt a numeric vector

Rb a numeric vector

Re a numeric vector

S a numeric vector

Sb a numeric vector

Sc a numeric vector

Se a numeric vector

Sn a numeric vector

Sr a numeric vector

Ta a numeric vector

Te a numeric vector

Th a numeric vector

Ti a numeric vector

Tl a numeric vector

U a numeric vector

V a numeric vector

W a numeric vector

Y a numeric vector

Zn a numeric vector

Zr a numeric vector

Details

Geochemical data set. 41 sample sites have been investigated. At each site, 15 different sample materials have been collected and analyzed for the concentration of more than 40 chemical elements. Soil: CHO - C horizon, OHO - O horizon. Mushroom: LAC - milkcap. Plant: BIL - birch leaves, BLE - blueberry leaves, BLU - blueberry twigs, BTW - birch twigs, CLE - cowberry leaves, COW - cowberry twigs, EQU - horsetail, FER - fern, HYL - terrestrial moss, PIB - pine bark, SNE - spruce needles, SPR - spruce twigs.

Author(s)

Peter Filzmoser, Dominika Miksova

References

C. Reimann, P. Englmaier, B. Flem, O.A. Eggen, T.E. Finne, M. Andersson & P. Filzmoser (2018). The response of 12 different plant materials and one mushroom to Mo and Pb mineralization along a 100-km transect in southern central Norway. Geochemistry: Exploration, Environment, Analysis, 18(3), 204-215.

Examples

```
data(gjovik)
str(gjovik)
```

---

gm *gmean*

Description

This function calculates the geometric mean.

Usage

```
gm(x)
```

Arguments

x	a vector

Details

`gm` calculates the geometric mean for all positive entries of a vector. Please note that there is a faster version available implemented with Rcpp but it currently do not pass CRAN checks cause of use of Rcpp11 features. This C++ version accounts for over- and underflows. It is placed in inst/doc

Author(s)

Matthias Templ

Examples

```
gm(c(3,5,3,6,7))
```

`gmean_sum` *Geometric mean*

Description

Computes the geometric mean(s) of a numeric vector, matrix or data.frame

Usage

```
gmean_sum(x, margin = NULL)

gmean(x, margin = NULL)
```

Arguments

x	matrix or data.frame with numeric entries
margin	a vector giving the subscripts which the function will be applied over, 1 indicates rows, 2 indicates columns, 3 indicates all values.

Details

`gmean_sum` calculates the totals based on geometric means while `gmean` calculates geometric means on rows (margin = 1), on columns (margin = 2), or on all values (margin = 3)

Value

geometric means (if `gmean` is used) or totals (if `gmean_sum` is used)

Author(s)

Matthias Templ

Examples

```
data("precipitation")
gmean_sum(precipitation)
gmean_sum(precipitation, margin = 2)
gmean_sum(precipitation, margin = 1)
gmean_sum(precipitation, margin = 3)
addmargins(precipitation)
addmargins(precipitation, FUN = gmean_sum)
addmargins(precipitation, FUN = mean)
addmargins(precipitation, FUN = gmean)

data("arcticLake", package = "robCompositions")
```

```
gmean(arcticLake$sand)
gmean(as.numeric(arcticLake[1, ]))
gmean(arcticLake)
gmean(arcticLake, margin = 1)
gmean(arcticLake, margin = 2)
gmean(arcticLake, margin = 3)
```

---

govexp *government spending*

Description

Government expenditures based on COFOG categories

Format

A (tidy) data frame with 5140 observations on the following 4 variables.

- `country` Country of origin
- `category` The COFOG expenditures are divided into in the following ten categories: general public services; defence; public order and safety; economic affairs; environmental protection; housing and community amenities; health; recreation, culture and religion; education; and social protection.
- `year` Year
- `value` COFOG spendings/expenditures

Details

The general government sector consists of central, state and local governments, and the social security funds controlled by these units. The data are based on the system of national accounts, a set of internationally agreed concepts, definitions, classifications and rules for national accounting. The classification of functions of government (COFOG) is used as classification system. The central government spending by category is measured as a percentage of total expenditures.

Author(s)

translated from https://data.oecd.org/ and restructured by Matthias Templ

Source

OECD: https://data.oecd.org/

Examples

```
data(govexp)
str(govexp)
```

haplogroups *haplogroups data*

Description

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

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; Hittite, Armenian

`G2a` Caucasian, Greco-Anatolian

`E1b1b` North and Eastern Africa, 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 into genealogical groups sharing a common ancestor, called haplogroups.

Source

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

Examples

```
data(haplogroups)
```

honey *honey compositions*

Description

The contents of honey, syrup, and adulteration mineral elements.

Format

A data frame with 429 observations on the following 17 variables.

`class` adulterated honey, Honey or Syrup

`group` group information

`group3` detailed group information

`group1` less detailed group information

`region` region

`Al` chemical element

`B` chemical element

`Ba` chemical element

`Ca` chemical element

`Fe` chemical element

`K` chemical element

`Mg` chemical element

`Mn` chemical element

`Na` chemical element

`P` chemical element

`Sr` chemical element

`Zn` chemical element

Details

Discrimination of honey and adulteration by elemental chemometrics profiling.

Note

In the original paper, sparse PLS-DA were applied optimize the classify model and test effectiveness. Classify accuracy were exceed 87.7 percent.

Source

Mendeley Data, contributed by Liping Luo and translated to R by Matthias Templ

References

Tao Liu, Kang Ming, Wei Wang, Ning Qiao, Shengrong Qiu, Shengxiang Yi, Xueyong Huang, Lip-ing Luo, Discrimination of honey and syrup-based adulteration by mineral element chemometrics profiling,' Food Chemistry, Volume 343, 2021, doi:10.1016/j.foodchem.2020.128455.

Examples

```
data(honey)
```

`ilr.2x2` *ilr coordinates in 2x2 compositional tables*

Description

ilr coordinates of original, independent and interaction compositional table using SBP1 and SBP2

Usage

```
ilr.2x2(x, margin = 1, type = "independence", version = "book")
```

Arguments

`x`	a 2x2 table
`margin`	for 2x2 tables available for a whole set of another dimension. For example, if 2x2 tables are available for every country.
`type`	choose between "independence" or "interaction" table
`version`	the version used in the "paper" below or the version of the "book".

Value

The ilr coordinates

Author(s)

Kamila Facevicova, Matthias Templ

References

Facevicova, K., Hron, K., Todorov, V., Guo, D., Templ, M. (2014). Logratio approach to statistical analysis of 2x2 compositional tables. *Journal of Applied Statistics*, 41 (5), 944–958.

Examples

```
data(employment)
ilr.2x2(employment[,,"AUT"])
ilr.2x2(employment[,,"AUT"], version = "paper")
ilr.2x2(employment, margin = 3, version = "paper")
ilr.2x2(employment[,,"AUT"], type = "interaction")
```

`impAll` *Replacement of rounded zeros and missing values.*

Description

Parametric replacement of rounded zeros and missing values for compositional data using classical and robust methods based on ilr coordinates with special choice of balances. Values under detection limit should be saved with the negative value of the detection limit (per variable). Missing values should be coded as NA.

Usage

```
impAll(x)
```

Arguments

`x`	data frame

Details

This is a wrapper function that calls *impRZilr()* for the replacement of zeros and *impCoda* for the imputation of missing values sequentially. The detection limit is automatically derived form negative numbers in the data set.

Value

The imputed data set.

Note

This function is mainly used by the compositionsGUI.

References

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

Martin-Fernandez, J.A., Hron, K., Templ, M., Filzmoser, P., Palarea-Albaladejo, J. (2012) Model-based replacement of rounded zeros in compositional data: Classical and robust approaches, *Computational Statistics*, 56 (2012), 2688 - 2704.

See Also

`impCoda`, `impRZilr`

Examples

```
## see the compositionsGUI
```

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, ncol(x)),
  noise = 0.1,
  bruteforce = FALSE
)
```

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`.

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

See Also

`impKNNa`, `pivotCoord`

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

# other methods
impCoda(x, method = "lm")
impCoda(x, method = "ltsReg")
```

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

Hron, K., Templ, M., Filzmoser, P. (2010) Imputation of missing values for compositional data using classical and robust methods *Computational Statistics and Data Analysis*, 54 (12), 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 of 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 = "lm",
  step = FALSE,
  nComp = "boot",
  R = 10,
  verbose = FALSE
)
```

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
`nComp`	if determined, it fixes the number of pls components. If "boot", the number of pls components are estimated using a bootstraped cross validation approach.
`R`	number of bootstrap samples for the determination of pls components. Only important for method "pls".
`verbose`	additional print output during calculations.

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

References

Palarea-Albaladejo, J., Martin-Fernandez, J.A. Gomez-Garcia, J. (2007) A parametric approach for dealing with compositional rounded zeros. *Mathematical Geology*, 39(7), 625-645.

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 coordinates with a special choice of balances.

Usage

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

```
  variation = FALSE,
  nComp = "boot",
  bruteforce = FALSE,
  noisemethod = "residuals",
  noise = FALSE,
  R = 10,
  correction = "normal",
  verbose = FALSE
)
```

Arguments

`x`	data.frame or matrix
`maxit`	maximum number of iterations
`eps`	convergency criteria
`method`	either “lm”, “MM” or “pls”
`dl`	Detection limit for each variable. zero for variables with variables that have no detection limit problems.
`variation`	matrix is used to first select number of parts
`nComp`	if determined, it fixes the number of pls components. If “boot”, the number of pls components are estimated using a bootstraped cross validation approach.
`bruteforce`	sets imputed values above the detection limit to the detection limit. Replacement above the detection limit only exceptionally occur due to numerical instabilities. The default is FALSE!
`noisemethod`	adding noise to imputed values. Experimental
`noise`	TRUE to activate noise (experimental)
`R`	number of bootstrap samples for the determination of pls components. Only important for method “pls”.
`correction`	normal or density
`verbose`	additional print output during calculations.

Details

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

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

Value

`x`	imputed data
`criteria`	change between last and second last iteration
`iter`	number of iterations

maxit	maximum number of iterations
wind	index of zeros
nComp	number of components for method pls
method	chosen method

Author(s)

Matthias Templ and Peter Filzmoser

References

Martin-Fernandez, J.A., Hron, K., Templ, M., Filzmoser, P., Palarea-Albaladejo, J. (2012) Model-based replacement of rounded zeros in compositional data: Classical and robust approaches. *Computational Statistics and Data Analysis*, 56 (9), 2688-2704.

Templ, M., Hron, K., Filzmoser, P., Gardlo, A. (2016) Imputation of rounded zeros for high-dimensional compositional data. *Chemometrics and Intelligent Laboratory Systems*, 155, 183-190.

See Also

impRZalr

Examples

```
data(arcticLake)
x <- arcticLake
## generate rounded zeros artificially:
#x[x[,1] < 5, 1] <- 0
x[x[,2] < 44, 2] <- 0
xia <- impRZilr(x, dl=c(5,44,0), eps=0.01, method="lm")
xia$x
```

---

imputeBDLs *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 coordinates with a special choice of balances.

Usage

```
imputeBDLs(
  x,
  maxit = 10,
  eps = 0.1,
  method = "subPLS",
```

```
    dl = rep(0.05, ncol(x)),
    variation = TRUE,
    nPred = NULL,
    nComp = "boot",
    bruteforce = FALSE,
    noisemethod = "residuals",
    noise = FALSE,
    R = 10,
    correction = "normal",
    verbose = FALSE,
    test = FALSE
  )

  adjustImputed(xImp, xOrig, wind)

  checkData(x, dl)

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

Arguments

x	data.frame or matrix
maxit	maximum number of iterations
eps	convergency criteria
method	either "lm", "lmrob" or "pls"
dl	Detection limit for each variable. zero for variables with variables that have no detection limit problems.
variation	if TRUE those predictors are chosen in each step, who's variation is lowest to the predictor.
nPred	if determined and variation equals TRUE, it fixes the number of predictors
nComp	if determined, it fixes the number of pls components. If "boot", the number of pls components are estimated using a bootstraped cross validation approach.
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!
noisemethod	adding noise to imputed values. Experimental
noise	TRUE to activate noise (experimental)
R	number of bootstrap samples for the determination of pls components. Only important for method "pls".
correction	normal or density
verbose	additional print output during calculations.
test	an internal test situation (this parameter will be deleted soon)
xImp	imputed data set

xOrig	original data set
wind	index matrix of rounded zeros
...	further arguments passed through the print function

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) compositional data are expressed in pivot coordinates (2) tobit regression is applied (3) the rounded zeros are replaced by the expected values (4) the corresponding inverse ilr mapping is applied. After all parts are imputed, the algorithm starts again until the imputations do not change.

Value

x	imputed data
criteria	change between last and second last iteration
iter	number of iterations
maxit	maximum number of iterations
wind	index of zeros
nComp	number of components for method pls
method	chosen method

Author(s)

Matthias Templ, method subPLS from Jiajia Chen

References

Templ, M., Hron, K., Filzmoser, P., Gardlo, A. (2016). Imputation of rounded zeros for high-dimensional compositional data. *Chemometrics and Intelligent Laboratory Systems*, 155, 183-190.

Chen, J., Zhang, X., Hron, K., Templ, M., Li, S. (2018). Regression imputation with Q-mode clustering for rounded zero replacement in high-dimensional compositional data. *Journal of Applied Statistics*, 45 (11), 2067-2080.

See Also

imputeBDLs

Examples

```
p <- 10
n <- 50
k <- 2
T <- matrix(rnorm(n*k), ncol=k)
B <- matrix(runif(p*k,-1,1),ncol=k)
X <- T %*% t(B)
```

```
E <-  matrix(rnorm(n*p, 0,0.1), ncol=p)
XE <- X + E
data <- data.frame(pivotCoordInv(XE))
col <- ncol(data)
row <- nrow(data)
DL <- matrix(rep(0),ncol=col,nrow=1)
for(j in seq(1,col,2))
{DL[j] <- quantile(data[,j],probs=0.06,na.rm=FALSE)}

for(j in 1:col){
  data[data[,j]<DL[j],j] <- 0
}
## Not run:
# under dontrun because of long exectution time
imp <- imputeBDLs(data,dl=DL,maxit=10,eps=0.1,R=10,method="subPLS")
imp
imp <- imputeBDLs(data,dl=DL,maxit=10,eps=0.1,R=10,method="pls", variation = FALSE)
imp
imp <- imputeBDLs(data,dl=DL,maxit=10,eps=0.1,R=10,method="lm")
imp
imp <- imputeBDLs(data,dl=DL,maxit=10,eps=0.1,R=10,method="lmrob")
imp

data(mcad)
## generate rounded zeros artificially:
x <- mcad
x <- x[1:25, 2:ncol(x)]
dl <- apply(x, 2, quantile, 0.1)
for(i in seq(1, ncol(x), 2)){
  x[x[,i] < dl[i], i] <- 0
}
ni <- sum(x==0, na.rm=TRUE)
ni/(ncol(x)*nrow(x)) * 100
dl[seq(2, ncol(x), 2)] <- 0
replaced_lm <- imputeBDLs(x, dl=dl, eps=1, method="lm",
  verbose=FALSE, R=50, variation=TRUE)$x
replaced_lmrob <- imputeBDLs(x, dl=dl, eps=1, method="lmrob",
  verbose=FALSE, R=50, variation=TRUE)$x
replaced_plsfull <- imputeBDLs(x, dl=dl, eps=1,
  method="pls", verbose=FALSE, R=50,
  variation=FALSE)$x

## End(Not run)
```

`imputeUDLs` *Imputation of values above an upper detection limit in compositional data*

Description

Parametric replacement of values above upper detection limit for compositional data using classical and robust methods (possibly also the pls method) based on ilr-transformations with special choice of balances.

Usage

```
imputeUDLs(
  x,
  maxit = 10,
  eps = 0.1,
  method = "lm",
  dl = NULL,
  variation = TRUE,
  nPred = NULL,
  nComp = "boot",
  bruteforce = FALSE,
  noisemethod = "residuals",
  noise = FALSE,
  R = 10,
  correction = "normal",
  verbose = FALSE
)
```

Arguments

`x`	data.frame or matrix
`maxit`	maximum number of iterations
`eps`	convergency criteria
`method`	either "lm", "lmrob" or "pls"
`dl`	Detection limit for each variable. zero for variables with variables that have no detection limit problems.
`variation`	if TRUE those predictors are chosen in each step, who's variation is lowest to the predictor.
`nPred`	if determined and variation equals TRUE, it fixes the number of predictors
`nComp`	if determined, it fixes the number of pls components. If "boot", the number of pls components are estimated using a bootstraped cross validation approach.
`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!
`noisemethod`	adding noise to imputed values. Experimental
`noise`	TRUE to activate noise (experimental)
`R`	number of bootstrap samples for the determination of pls components. Only important for method "pls".
`correction`	normal or density
`verbose`	additional print output during calculations.

Details

imputeUDLs

An imputation method for right-censored compositional data. Statistical analysis is not possible with values reported in data, for example as ">10000". These values are replaced using tobit regression.

The algorithm iteratively imputes parts with values above upper detection limit whereas in each step (1) compositional data are expressed in pivot coordinates (2) tobit regression is applied (3) the values above upper detection limit are replaced by the expected values (4) the corresponding inverse ilr mapping is applied. After all parts are imputed, the algorithm starts again until the imputations only change marginally.

Value

`x`	imputed data
`criteria`	change between last and second last iteration
`iter`	number of iterations
`maxit`	maximum number of iterations
`wind`	index of values above upper detection limit
`nComp`	number of components for method pls
`method`	chosen method

Author(s)

Peter Filzmoser, Dominika Miksova based on function imputeBDLs code from Matthias Templ

References

Martin-Fernandez, J.A., Hron K., Templ, M., Filzmoser, P. and Palarea-Albaladejo, J. (2012). Model-based replacement of rounded zeros in compositional data: Classical and robust approaches. *Computational Statistics and Data Analysis*, 56, 2688-2704.

Templ, M. and Hron, K. and Filzmoser and Gardlo, A. (2016). Imputation of rounded zeros for high-dimensional compositional data. *Chemometrics and Intelligent Laboratory Systems*, 155, 183-190.

See Also

`imputeBDLs`

Examples

```
data(gemas)  # read data
dat <- gemas[gemas$COUNTRY=="HEL",c(12:29)]
UDL <- apply(dat,2,max)
names(UDL) <- names(dat)
UDL["Mn"] <- quantile(dat[,"Mn"], probs = 0.8)  # UDL present only in one variable
whichudl <- dat[,"Mn"] > UDL["Mn"]
# classical method
imp.lm <- dat
```

```
imp.lm[whichudl,"Mn"] <- Inf
res.lm <- imputeUDLs(imp.lm, dl=UDL, method="lm", variation=TRUE)
imp.lm <- res.lm$x
```

---

ind2x2 *Independence 2x2 compositional table*

Description

Estimates the expected frequencies from an 2x2 table under the null hypotheses of independence.

Usage

```
ind2x2(x, margin = 3, pTabMethod = c("dirichlet", "half", "classical"))
```

Arguments

x	a 2x2 table
margin	if multidimensional table (larger than 2-dimensional), then the margin determines on which dimension the independene tables should be estimated.
pTabMethod	'classical' that is function `prop.table()` from package base or method "half" that add 1/2 to each cell to avoid zero problems.

Value

The independence table(s) with either relative or absolute frequencies.

Author(s)

Kamila Facevicova, Matthias Templ

References

Facevicova, K., Hron, K., Todorov, V., Guo, D., Templ, M. (2014). Logratio approach to statistical analysis of 2x2 compositional tables. *Journal of Applied Statistics*, 41 (5), 944–958.

Examples

```
data(employment)
ind2x2(employment)
```

indTab *Independence table*

Description

Estimates the expected frequencies from an m-way table under the null hypotheses of independence.

Usage

```
indTab(
  x,
  margin = c("gmean_sum", "sum"),
  frequency = c("relative", "absolute"),
  pTabMethod = c("dirichlet", "half", "classical")
)
```

Arguments

x	an object of class table
margin	determines how the margins of the table should be estimated (default via geometric mean margins)
frequency	indicates whether absolute or relative frequencies should be computed.
pTabMethod	to estimate the propability table. Default is 'dirichlet'. Other available methods: 'classical' that is function `prop.table()` from package base or method "half" that add 1/2 to each cell to avoid zero problems.

Details

Because of the compositional nature of probability tables, the independence tables should be estimated using geometric marginals.

Value

The independence table(s) with either relative or absolute frequencies.

Author(s)

Matthias Templ

References

Egozcue, J.J., Pawlowsky-Glahn, V., Templ, M., Hron, K. (2015) Independence in contingency tables using simplicial geometry. *Communications in Statistics - Theory and Methods*, 44 (18), 3978–3996.

Examples

```
data(precipitation)
tab1 <- indTab(precipitation)
tab1
sum(tab1)

## Not run:
data("PreSex", package = "vcd")
indTab(PreSex)

## End(Not run)
```

`instw` *value added, output and input for different ISIC codes and countries.*

Description

value added, output and input for different ISIC codes and countries.

Usage

```
data(instw)
```

Format

A data.frame with 1555 rows and 7 columns:

`ct` ct

`isic` ISIC classification, Rev 3.2

`VA` value added

`OUT` output

`INP` input

`IS03` country code

`mht` mht

A data.frame with 1555 rows and 7 columns.

Examples

```
data(instw)
head(instw)
```

int2x2 *Interaction 2x2 table*

Description

Estimates the interactions from an 2x2 table under the null hypotheses of independence.

Usage

```
int2x2(x, margin = 3, pTabMethod = c("dirichlet", "half", "classical"))
```

Arguments

`x`	a 2x2 table
`margin`	if multidimensional table (larger than 2-dimensional), then the margin determines on which dimension the independene tables should be estimated.
`pTabMethod`	to estimate the propability table. Default is 'dirichlet'. Other available methods: 'classical' that is function `prop.table()` from package base or method "half" that add 1/2 to each cell to avoid zero problems.

Value

The independence table(s) with either relative or absolute frequencies.

Author(s)

Kamila Facevicova, Matthias Templ

References

Facevicova, K., Hron, K., Todorov, V., Guo, D., Templ, M. (2014). Logratio approach to statistical analysis of 2x2 compositional tables. *Journal of Applied Statistics*, 41 (5), 944–958.

Examples

```
data(employment)
int2x2(employment)
```

`intArray` *Interaction array*

Description

Estimates the interaction compositional table with normalization for further analysis according to Egozcue et al. (2015)

Usage

```
intArray(x)
```

Arguments

`x`	an object of class "intTab"

Details

Estimates the interaction table using its ilr coordinates.

Value

The interaction array

Author(s)

Matthias Templ

References

Egozcue, J.J., Pawlowsky-Glahn, V., Templ, M., Hron, K. (2015) Independence in contingency tables using simplicial geometry. *Communications in Statistics - Theory and Methods*, 44 (18), 3978–3996.

See Also

`intTab`

Examples

```
data(precipitation)
tab1prob <- prop.table(precipitation)
tab1 <- indTab(precipitation)
tabINT <- intTab(tab1prob, tab1)
intArray(tabINT)
```

intTab *Interaction table*

Description

Estimates the interaction table based on clr and inverse clr coefficients.

Usage

```
intTab(x, y, frequencies = c("relative", "absolute"))
```

Arguments

`x`	an object of class table
`y`	the corresponding independence table which is of class “intTab”.
`frequencies`	indicates whether absolute or relative frequencies should be computed.

Details

Because of the compositional nature of probability tables, the independence tables should be estimated using geometric marginals.

Value

intTab The interaction table(s) with either relative or absolute frequencies.

signs The sign illustrates if there is an excess of probability (plus), or a deficit (minus) regarding to the estimated probability table and the independece table in the clr space.

Author(s)

Matthias Templ

References

Egozcue, J.J., Pawlowsky-Glahn, V., Templ, M., Hron, K. (2015) Independence in contingency tables using simplicial geometry. *Communications in Statistics - Theory and Methods*, 44 (18), 3978–3996.

Examples

```
data(precipitation)
tab1prob <- prop.table(precipitation)
tab1 <- indTab(precipitation)
intTab(tab1prob, tab1)
```

is.equivalent *equivalence class*

Description

Checks if two vectors or two data frames are from the same equivalence class

Usage

```
is.equivalent(x, y, tollerance = .Machine$double.eps^0.5)
```

Arguments

x	either a numeric vector, or a data.frame containing such vectors.
y	either a numeric vector, or a data.frame containing such vectors.
tollerance	numeric >= 0. Differences smaller than tolerance are not considered.

Value

logical TRUE if the two vectors are from the same equivalence class.

Author(s)

Matthias Templ

References

Filzmoser, P., Hron, K., Templ, M. (2018) *Applied Compositional Data Analysis*. Springer, Cham.

See Also

all.equal

Examples

```
is.equivalent(1:10, 1:10*2)
is.equivalent(1:10, 1:10+1)
data(expenditures)
x <- expenditures
is.equivalent(x, constSum(x))
y <- x
y[1,1] <- x[1,1]+1
is.equivalent(y, constSum(x))
```

isic32 *ISIC codes by name*

Description

`code` ISIC code, Rev 3.2

`description` Description of ISIC codes

Usage

```
data(isic32)
```

Format

A data.frame with 24 rows and 2 columns.

Examples

```
data(instw)
instw
```

laborForce *labour force by status in employment*

Description

Labour force by status in employment for 124 countries, latest update: December 2009

Format

A data set on 124 compositions on 9 variables.

Details

`country` country

`year` year

`employeesW` percentage female employees

`employeesM` percentage male employees

`employersW` percentage female employers

`employersM` percentage male employers

`ownW` percentage female own-account workers and contributing family workers

`ownM` percentage male own-account workers and contributing family workers

`source` HS: household or labour force survey. OE: official estimates. PC: population census

Author(s)

conversion to R by Karel Hron and Matthias Templ <matthias.templ@tuwien.ac.at>

Source

from UNSTATS website

References

K. Hron, P. Filzmoser, K. Thompson (2012). Linear regression with compositional explanatory variables. *Journal of Applied Statistics*, Volume 39, Issue 5, 2012.

Examples

```
data(laborForce)
str(laborForce)
```

`landcover` *European land cover*

Description

Land cover data from Eurostat (2015) extended with (log) population and (log) pollution

Format

A data set on 28 compositions on 7 variables.

Details

`Woodland` Coverage in km2

`Cropland` Coverage in km2

`Grassland` Coverage in km2

`Water` Coverage in km2

`Artificial` Coverage in km2

`Pollution` log(Pollution) values per country

`PopDensity` log(PopDensity) values per country

Author(s)

conversion to R by Karel Hron

Source

Lucas land cover

Examples

```
data(landcover)
str(landcover)
```

`lifeExpGdp` *life expectancy and GDP (2008) for EU-countries*

Description

Social-economic data for compositional regression.

Format

A data set on 27 compositions on 9 variables.

Details

`country` country

`agriculture` GDP on agriculture, hunting, forestry, fishing (ISIC A-B, x1)

`manufacture` GDP on mining, manufacturing, utilities (ISIC C-E, x2)

`construction` GDP on construction (ISIC F, x3)

`wholesales` GDP on wholesale, retail trade, restaurants and hotels (ISIC G-H, x4)

`transport` GDP on transport, storage and communication (ISIC I, x5)

`other` GDP on other activities (ISIC J-P, x6)

`lifeExpMen` life expectancy for men and women

`lifeExpWomen` life expectancy for men and women

Author(s)

conversion to R by Karel Hron and Matthias Templ <matthias.templ@tuwien.ac.at>

Source

https://www.ec.europa.eu/eurostat and https://unstats.un.org/home/

References

K. Hron, P. Filzmoser, K. Thompson (2012). Linear regression with compositional explanatory variables. *Journal of Applied Statistics*, Volume 39, Issue 5, 2012.

Examples

```
data(lifeExpGdp)
str(lifeExpGdp)
```

`lmCoDaX` *Classical and robust regression of non-compositional (real) response on compositional and non-compositional predictors*

Description

Delivers appropriate inference for regression of y on a compositional matrix X or and compositional and non-compositional combined predictors.

Usage

```
lmCoDaX(
  y,
  X,
  external = NULL,
  method = "robust",
  pivot_norm = "orthonormal",
  max_refinement_steps = 200
)
```

Arguments

`y` The response which should be non-compositional

`X` The compositional and/or non-compositional predictors as a matrix, data.frame or numeric vector

`external` Specify the columns name of the external variables. The name has to be introduced as follows: external = c("variable_name"). Multiple selection is supported for the external variable. Factor variables are automatically detected.

`method` If robust, LTS-regression is applied, while with method equals "classical", the conventional least squares regression is applied.

`pivot_norm` if FALSE then the normalizing constant is not used, if TRUE sqrt((D-i)/(D-i+1)) is used (default). The user can also specify a self-defined constant.

`max_refinement_steps` (for the fast-S algorithm): maximal number of refinement steps for the fully iterated best candidates.

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 pivot coordinates is used. Further these compositional explanatory variables can be supplemented with external non-compositional data and factor variables.

Value

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

Author(s)

Peter Filzmoser, Roman Wiedemeier, Matthias Templ

References

Filzmoser, P., Hron, K., Thompsonc, K. (2012) Linear regression with compositional explanatory variables. *Journal of Applied Statistics*, 39, 1115-1128.

See Also

lm

Examples

```
## How the total household expenditures in EU Member
## States depend on relative contributions of
## single household expenditures:
data(expendituresEU)
y <- as.numeric(apply(expendituresEU,1,sum))
lmCoDaX(y, expendituresEU, method="classical")

## How the relative content of sand of the agricultural
## and grazing land soils in Germany depend on
## relative contributions of the main chemical trace elements,
## their different soil types and the Annual mean temperature:
data("gemas")
gemas$COUNTRY <- as.factor(gemas$COUNTRY)
gemas_GER <- dplyr::filter(gemas, gemas$COUNTRY == 'POL')
ssc <- cenLR(gemas_GER[, c("sand", "silt", "clay")])$x.clr
y <- ssc$sand
X <- dplyr::select(gemas_GER, c(MeanTemp, soilclass, Al:Zr))
X$soilclass <- factor(X$soilclass)
lmCoDaX(y, X, external = c('MeanTemp', 'soilclass'),
method='classical', pivot_norm = 'orthonormal')
lmCoDaX(y, X, external = c('MeanTemp', 'soilclass'),
method='robust', pivot_norm = 'orthonormal')
```

---

machineOperators *machine operators*

Description

Compositions of eight-hour shifts of 27 machine operators

Usage

```
data(machineOperators)
```

Format

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

Details

`hqproduction` high-quality production

`lqproduction` low-quality production

`setting` machine settings

`repair` machine repair

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.

Author(s)

Matthias Templ `<matthias.templ@tuwien.ac.at>`

References

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

Examples

```
data(machineOperators)
str(machineOperators)
summary(machineOperators)
rowSums(machineOperators)
```

---

manu_abs *Distribution of manufacturing output*

Description

The data consists of values of the manufacturing output in 42 countries in 2009. The output, given in national currencies, is structured according to the 3-digit ISIC category and its components. Thorough analysis of the sample is described in Facevicova (2018).

Usage

```
data(manu_abs)
```

Format

A data frame with 630 observations of 4 variables.

Details

`country` Country

`isic` 3-digit ISIC category. The categories are 151 processed meat, fish, fruit, vegetables, fats; 152 Dairy products; 153 Grain mill products, starches, animal feeds; 154 Other food products and 155 Beverages.

`output` The output components are Labour, Surplus and Input.

`value` Value of manufacturing output in the national currency

Author(s)

Kamila Facevicova

Source

Elaboration based on the INDSTAT 4 database (UNIDO 2012a), see also UNIDO, 2012b. UNIDO (2012a), INDSTAT 4 Industrial Statistics Database at 3- and 4-digit level of ISIC Revision 3 and 4. Vienna. Available from https://stat.unido.org. UNIDO (2012b) International Yearbook of Industrial Statistics, Edward Elgar Publishing Ltd, UK.

References

Facevicova, K., Hron, K., Todorov, V. and M. Templ (2018) General approach to coordinate representation of compositional tables. Scandinavian Journal of Statistics, 45(4).

Examples

```
data(manu_abs)

### Compositional tables approach
### analysis of the relative structure

result <- tabCoordWrapper(manu_abs, obs.ID='country',row.factor = 'output',
col.factor = 'isic', value='value', test = TRUE)

result$Bootstrap

### Classical approach
### generalized linear mixed effect model
## Not run:
library(lme4)
m <- glmer(value~output*as.factor(isic)+(1|country),
data=manu_abs,family=poisson)
summary(m)

## End(Not run)
```

`mcad` *metabolomics mcad data set*

Description

The aim of the experiment was to ascertain novel biomarkers of MCAD (Medium chain acyl-CoA dehydrogenase) deficiency. The data consists of 25 patients and 25 controls and the analysis was done by LC-MS. Rows represent patients and controls and columns represent chemical entities with their quantity.

Usage

```
data(mcad)
```

Format

A data frame with 50 observations and 279 variables

Details

- `group` patient group
- `...` the remaining variables columns are represented by m/z which are chemical characterizations of individual chemical components on exact mass measurements..

References

Najdekr L., Gardlo A., Madrova L., Friedeckyy D., Janeckova H., Correa E.S., Goodacre R., Adam T., Oxidized phosphatidylcholines suggest oxidative stress in patients with medium-chain acyl-CoA dehydrogenase deficiency, *Talanta* 139, 2015, 62-66.

Examples

```
data(mcad)
str(mcad)
```

`missPatterns` *missing or zero pattern structure.*

Description

Analysis of the missing or the zero patterns structure of a data set.

Usage

```
missPatterns(x)

zeroPatterns(x)
```

Arguments

`x`	a data frame or matrix.

Details

Here, one pattern defines those observations that have the same structure regarding their missingness or zeros. For all patterns a summary is calculated.

Value

`groups`	List of the different patterns and the observation numbers for each pattern
`cn`	the names of the patterns coded as vectors of 0-1's
`tabcomb`	the pattern structure - all combinations of zeros or missings in the variables
`tabcombPlus`	the pattern structure - all combinations of zeros or missings in the variables including the size of those combinations/patterns, i.e. the number of observations that belongs to each pattern.
`rsum`	the number of zeros or missing values in each row of the data set.
`rindex`	the index of zeros or missing values in each row of the data set

Author(s)

Matthias Templ. The code is based on a previous version from Andreas Alfons and Matthias Templ from package VIM

See Also

`aggr`

Examples

```
data(expenditures)
## set NA's artificial:
expenditures[expenditures < 300] <- NA
## detect the NA structure:
missPatterns(expenditures)
```

`mortality` *mortality and life expectancy in the EU*

Description

`country` country name

`country2` country name, short version

`sex` gender

`lifeExpectancy` life expectancy

`infectious` certain infectious and parasitic diseases (A00-B99)

`neoplasms` malignant neoplasms (C00-C97)

`endocrine` endocrine nutritional and metabolic diseases (E00-E90)

`mental` mental and behavioural disorders (F00-F99)

`nervous` diseases of the nervous system and the sense organs (G00-H95)

`circulatory` diseases of the circulatory system (I00-I99)

`respiratory` diseases of the respiratory system (J00-J99)

`digestive` diseases of the digestive system (K00-K93)

Usage

```
data(mortality)
```

Format

A data frame with 60 observations and 12 variables

Author(s)

Peter Filzmoser, Matthias Templ <matthias.templ@tuwien.ac.at>

References

Eurostat, https://ec.europa.eu/eurostat/data

Examples

```
data(mortality)
str(mortality)
## totals (mortality)
aggregate(mortality[,5:ncol(mortality)],
          list(mortality$country2), sum)
```

mortality_tab — *mortality table*

Description

Mortality data by gender, unknown year

Usage

```
data(mortality_tab)
```

Format

A table

Details

`female` mortality rates for females by age groups

`male` mortality rates for males by age groups

Author(s)

Matthias Templ

Examples

```
data(mortality_tab)
mortality_tab
```

norm1 — *Normalize a vector to length 1*

Description

Scales a vector to a unit vector.

Usage

```
norm1(x)
```

Arguments

`x`	a numeric vector

Author(s)

Matthias Templ

Examples

```
data(expenditures)
i <- 1
D <- 6
vec <- c(rep(-1/i, i), 1, rep(0, (D-i-1)))

norm1(vec)
```

`nutrients` *nutrient contents*

Description

Nutrients on more than 40 components and 965 generic food products

Usage

```
data(nutrients)
```

Format

A data frame with 965 observations on the following 50 variables.

Details

`ID` ID, for internal use

`ID_V4` ID V4, for internal use

`ID_SwissFIR` ID, for internal use

`name_D` Name in German

`name_F` Name in French

`name_I` Name in Italian

`name_E` Name in Spanish

`category_D` Category name in German

`category_F` Category name in French

`category_I` Category name in Italy

`category_E` Category name in Spanish

`gravity` specific gravity

‘`energy_kJ`’ energy in kJ per 100g edible portion

`energy_kcal` energy in kcal per 100g edible portion

`protein` protein in gram per 100g edible portion

`alcohol` alcohol in gram per 100g edible portion

`water` water in gram per 100g edible portion `carbohydrates` crbohydrates in gram per 100g edible portion

`starch` starch in gram per 100g edible portion

`sugars` sugars in gram per 100g edible portion

'`dietar_ fibres`' dietar fibres in gram per 100g edible portion

`fat` fat in gram per 100g edible portion

`cholesterol` cholesterolin milligram per 100g edible portion

`fattyacids_monounsaturated` fatty acids monounsatrurated in gram per 100g edible portion

`fattyacids_saturated` fatty acids saturated in gram per 100g edible portion

`fatty_acids_polyunsaturated` fatty acids polyunsaturated in gram per 100g edible portion

`vitaminA` vitamin A in retinol equivalent per 100g edible portion

'`all-trans_retinol_equivalents`' all trans-retinol equivalents in gram per 100g edible portion

'`beta-carotene-activity`' beta-carotene activity in beta-carotene equivalent per 100g edible portion

'`beta-carotene`' beta-carotene in micogram per 100g edible portion

`vitaminB1` vitamin B1 in milligram per 100g edible portion

`vitaminB2` vitamin B2 in milligram per 100g edible portion

`vitaminB6` vitamin B6 in milligram per 100g edible portion

`vitaminB12` vitamin B12 in micogram per 100g edible portion

`niacin` niacin in milligram per 100g edible portion

`folate` folate in micogram per 100g edible portion

`pantothenic_acid` pantothenic acid in milligram per 100g edible portion

`vitaminC` vitamin C in milligram per 100g edible portion

`vitaminD` vitamin D in micogram per 100g edible portion

`vitaminE` vitamin E in alpha-tocopherol equivalent per 100g edible portion

`Na` Sodium in milligram per 100g edible portion

`K` Potassium in milligram per 100g edible portion

`Cl` Chloride

`Ca` Calcium

`Mg` Magnesium

`P` Phosphorus

`Fe` Iron

`I` Iodide in milligram per 100g edible portion

`Zn` Zink

`unit` a factor with levels `per 100g edible portion` `per 100ml food volume`

Author(s)

Translated from the Swiss nutrion data base by Matthias Templ `<matthias.templ@tuwien.ac.at>`

Source

From the Swiss nutrition data base 2015 (second edition)

Examples

```
data(nutrients)
str(nutrients)
head(nutrients[, 41:49])
```

`nutrients_branded` *nutrient contents (branded)*

Description

Nutrients on more than 10 components and 9618 branded food products

Usage

```
data(nutrients_branded)
```

Format

A data frame with 9618 observations on the following 18 variables.

Details

`name_D` name (in German)

`category_D` factor specifying the category names

`category_F` factor specifying the category names

`category_I` factor specifying the category names

`category_E` factor specifying the category names

`gravity` specific gravity

`energy_kJ` energy in kJ

'`energy_kcal`' energy in kcal

`protein` protein in gram

`alcohol` alcohol in gram

`water` water in gram

`carbohydrates_available` available carbohydrates in gram

`sugars` sugars in gram

`dietary_fibres` dietary fibres in gram

`fat_total` total fat in gram

`fatty_acids_saturated` saturated acids fat in gram

`Na` Sodium in gram

`unit` a factor with levels `per 100g edible portion` `per 100ml food volume`

Author(s)

Translated from the Swiss nutrion data base by Matthias Templ <matthias.templ@tuwien.ac.at>

Source

From the Swiss nutrition data base 2015 (second edition)

Examples

```
data(nutrients_branded)
str(nutrients_branded)
```

---

orthbasis — *Orthonormal basis*

Description

Orthonormal basis from cenLR transformed data to pivotCoord transformated data.

Usage

```
orthbasis(D)
```

Arguments

D	number of parts (variables)

Details

For the chosen balances for "pivotCoord", this is the orthonormal basis that transfers the data from centered logratio to isometric logratio.

Value

the orthonormal basis.

Author(s)

Karel Hron, Matthias Templ. Some code lines of this function are a copy from function gsi.buildilr from

See Also

`pivotCoord`, `cenLR`

Examples

```
data(expenditures)
V <- orthbasis(ncol(expenditures))
xcen <- cenLR(expenditures)$x.clr
xi <- as.matrix(xcen) %*% V$V
xi
xi2 <- pivotCoord(expenditures)
xi2
```

---

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", alpha = 0.5, coda = TRUE)

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

## S3 method for class 'outCoDa'
plot(x, y, ..., which = 1)
```

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"
`alpha`	the size of the subsets for the robust covariance estimation according the MCD-estimator for which the determinant is minimized, see `covMcd`.
`coda`	if TRUE, data transformed to coordinate representation before outlier detection.
`...`	additional parameters for print and plot method passed through
`y`	unused second plot argument for the plot method
`which`	1 ... MD against index 2 ... distance-distance plot

Details

The outlier detection procedure is based on (robust) Mahalanobis distances in isometric logratio coordinates. 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. Method “robust” can be used if the number of variables is greater than the number of observations. Here the OGK estimator is chosen.

plot method: the Mahalanobis distance are plotted against the index. 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.

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., Pawlowsky-Glahn, V., Mateu-Figueras, G., Barcelo-Vidal, C. (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

`pivotCoord`

Examples

```
data(expenditures)
oD <- outCoDa(expenditures)
oD
## providing a function:
oD <- outCoDa(expenditures, coda = log)
```

```
## for high-dimensional data:
oD <- outCoDa(expenditures, method = "robustHD")
```

---

payments *special payments*

Description

Payments splitted by different NACE categories and kind of employment in Austria 2004

Usage

```
data(payments)
```

Format

A data frame with 535 rows and 11 variables

Details

`nace` NACE classification, 2 digits

`oenace_2008` Corresponding Austrian NACE classification (in German)

`year` year

`month` month

`localunit` local unit ID

`spay` special payments (total)

`spay_wc` special payments for white colar workers

`spay_bc` special payments for blue colar workers

`spay_traintrade` special payments for trainees in trade businness

`spay_home` special payments for home workers

`spay_traincomm` special payments for trainees in commercial businness

Author(s)

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

Source

statCube data base at the website of Statistics Austria. The product and all material contained therein are protected by copyright with all rights reserved by the Bundesanstalt Statistik Oesterreich (STATISTICS AUSTRIA). It is permitted to reproduce, distribute, make publicly available and process the content for non-commercial purposes. Prior to any use for commercial purposes a written consent of STATISTICS AUSTRIA must be obtained. Any use of the contained material must be correctly reproduced and clearly cite the source STATISTICS AUSTRIA. If tables published by STATISTICS AUSTRIA are partially used, displayed or otherwise changed, a note must be added at an adequate position to show data was extracted or adapted.

Examples

```
data(payments)
str(payments)
summary(payments)
```

---

pcaCoDa *Robust principal component analysis for compositional data*

Description

This function applies robust principal component analysis for compositional data.

Usage

```
pcaCoDa(
  x,
  method = "robust",
  mult_comp = NULL,
  external = NULL,
  solve = "eigen"
)

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

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

Arguments

`x`	compositional data
`method`	must be either “robust” (default) or “classical”
`mult_comp`	a list of numeric vectors holding the indices of linked compositions
`external`	external non-compositional variables
`solve`	eigen (as princomp does, i.e. eigenvalues of the covariance matrix) or svd (as prcomp does with single value decomposition instead of eigen). Only for method classical.
`...`	additional parameters for print method passed through
`object`	object of class pcaCoDa

Details

The compositional data set is expressed in isometric logratio coordinates. 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.

`mult_comp` is used when there are more than one group of compositional parts in the data. To give an illustrative example, lets assume that one variable group measures angles of the inner ear-bones of animals which sum up to 100 and another one having percentages of a whole on the thickness of the inner ear-bones included. Then two groups of variables exists which are both compositional parts. The isometric logratio coordinates are then internally applied to each group independently whenever the `mult_comp` is set correctly.

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)

Karel Hron, Peter Filzmoser, Matthias Templ and a contribution for dimnames in external variables by Amelia Landre.

References

Filzmoser, P., Hron, K., Reimann, C. (2009) Principal component analysis for compositional data with outliers. *Environmetrics*, **20**, 621-632.

Kynclova, P., Filzmoser, P., Hron, K. (2016) Compositional biplots including external non-compositional variables. *Statistics: A Journal of Theoretical and Applied Statistics*, **50**, 1132-1148.

See Also

`print.pcaCoDa`, `summary.pcaCoDa`, `biplot.pcaCoDa`, `plot.pcaCoDa`

Examples

```
data(arcticLake)

## robust estimation (default):
res.rob <- pcaCoDa(arcticLake)
res.rob
summary(res.rob)
plot(res.rob)

## classical estimation:
res.cla <- pcaCoDa(arcticLake, method="classical", solve = "eigen")
```

```
biplot(res.cla)

## just for illustration how to set the mult_comp argument:
data(expenditures)
p1 <- pcaCoDa(expenditures, mult_comp=list(c(1,2,3),c(4,5)))
p1

## example with external variables:
data(election)
# transform external variables
election$unemployment <- log((election$unemployment/100)/(1-election$unemployment/100))
election$income <- scale(election$income)

res <- pcaCoDa(election[,1:6], method="classical", external=election[,7:8])
res
biplot(res, scale=0)
```

`perturbation` *Perturbation and powering*

Description

Perturbation and powering for two compositions.

Usage

```
perturbation(x, y)

powering(x, a)
```

Arguments

x	(compositional) vector containing positive values
y	(compositional) vector containing positive values or NULL for powering
a	constant, numeric vector of length 1

Value

Result of perturbation or powering

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.

Examples

```
data(expenditures)
x <- expenditures[1 ,]
y <- expenditures[2, ]
perturbation(x, y)
powering(x, 2)
```

---

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,
  robust = TRUE,
  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
`robust`	default value is TRUE
`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.
`uniqueness`	uniqueness
`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
res.rob <- pfa(x, factors=1)
res.cla <- pfa(x, factors=1, robust=FALSE)


## the following produce always the same result:
res1 <- pfa(x, factors=1, covmat="covMcd")
res2 <- pfa(x, factors=1, covmat=robustbase::covMcd(pivotCoord(x))$cov)
res3 <- pfa(x, factors=1, covmat=robustbase::covMcd(pivotCoord(x)))
```

---

phd *PhD students in the EU*

Description

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

Format

A data set on 32 compositions and 11 variables.

Details

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

`country` country of origin (German)

`countryEN` country of origin (English)

`country2` country of origin, 2-digits

`total` total phd students (in 1.000)

`male` male phd students (in 1.000)

`female` total phd students (in 1.000)

`technical` phd students in natural and technical sciences

`socio-economic-low` phd students in social sciences, economic sciences and law sciences

`human` phd students in human sciences including teaching

`health` phd students in health and life sciences

`agriculture` phd students in agriculture

Source

Eurostat

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.

Examples

```
data(phd)
str(phd)
```

`phd_totals` *PhD students in the EU (totals)*

Description

PhD students in Europe by different kind of studies.

Format

A data set on 29 compositions and 5 variables.

Details

`technical` phd students in natural and technical sciences

`socio-economic-low` phd students in social sciences, economic sciences and law sciences

`human` phd students in human sciences including teaching

`health` phd students in health and life sciences

`agriculture` phd students in agriculture

Source

Eurostat

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.

Examples

```
data("phd_totals")
str(phd_totals)
```

`pivotCoord` *Pivot coordinates and their inverse*

Description

Pivot coordinates as a special case of isometric logratio coordinates and their inverse mapping.

Usage

```
pivotCoord(
  x,
  pivotvar = 1,
  fast = FALSE,
  method = "pivot",
  base = exp(1),
  norm = "orthonormal"
)

isomLR(x, fast = FALSE, base = exp(1), norm = "sqrt((D-i)/(D-i+1))")

isomLRinv(x)

pivotCoordInv(x, norm = "orthonormal")

isomLRp(x, fast = FALSE, base = exp(1), norm = "sqrt((D-i)/(D-i+1))")

isomLRinvp(x)
```

Arguments

`x`	object of class data.frame or matrix. Positive values only.
`pivotvar`	pivotal variable. If any other number than 1, the data are resorted in that sense that the pivotvar is shifted to the first part.
`fast`	if TRUE, it is approx. 10 times faster but numerical problems in case of high-dimensional data may occur. Only available for method "pivot".
`method`	pivot takes the method described in the description. Method "symm" uses symmetric pivot coordinates (parameters pivotvar and norm have then no effect)
`base`	a positive or complex number: the base with respect to which logarithms are computed. Defaults to `exp(1)`.
`norm`	if FALSE then the normalizing constant is not used, if TRUE `sqrt((D-i)/(D-i+1))` is used (default). The user can also specify a self-defined constant.

Details

Pivot coordinates map D-part compositional data from the simplex into a (D-1)-dimensional real space isometrically. From our choice of pivot coordinates, all the relative information about one of parts (or about two parts) is aggregated in the first coordinate (or in the first two coordinates in case of symmetric pivot coordinates, respectively).

Value

The data represented in pivot coordinates

Author(s)

Matthias Templ, Karel Hron, Peter Filzmoser

References

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

Filzmoser, P., Hron, K., Templ, M. (2018) *Applied Compositional Data Analysis*. Springer, Cham.

Examples

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

data(expenditures)
## first variable as pivot variable
pivotCoord(expenditures)
## third variable as pivot variable
pivotCoord(expenditures, 3)

x <- exp(mvrnorm(2000, mu=rep(1,10), diag(10)))
system.time(pivotCoord(x))
system.time(pivotCoord(x, fast=TRUE))

## without normalizing constant
pivotCoord(expenditures, norm = "orthogonal") # or:
pivotCoord(expenditures, norm = "1")
## other normalization
pivotCoord(expenditures, norm = "-sqrt((D-i)/(D-i+1))")

# symmetric balances (results in 2-dim symmetric pivot coordinates)
pivotCoord(expenditures, method = "symm")
```

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),
  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.
`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. Currently, the ggpairs functions from the GGally package is used.

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 and 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

Examples

```
data(expenditures)
expenditures[1,3]
expenditures[1,3] <- NA
xi <- impKNNa(expenditures)
xi
summary(xi)
## Not run: plot(xi, which=1)
plot(xi, which=2)
plot(xi, which=3)
plot(xi, which=3, seg1=FALSE)
```

---

plot.pcaCoDa *Plot method*

Description

Provides a screeplot and biplot for (robust) compositional principal components analysis.

Usage

```
## S3 method for class 'pcaCoDa'
plot(x, y, ..., which = 1, choices = 1:2)
```

Arguments

x	object of class 'pcaCoDa'
y	...
...	...
which	an integer between 1 and 3. Produces a screeplot (1), or a biplot using stats biplot.prcomp function (2), or a biplot using ggfortify's autoplot function (3).
choices	principal components to plot by number

Value

The robust compositional screeplot.

Author(s)

M. Templ, K. Hron

References

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

See Also

pcaCoDa, biplot.pcaCoDa

Examples

```
data(coffee)
## Not run:
p1 <- pcaCoDa(coffee[,-1])
plot(p1)
plot(p1, type="lines")
plot(p1, which = 2)
plot(p1, which = 3)

## End(Not run)
```

`plot.smoothSpl` *plot smoothSpl*

Description

plot densities of objects of class smoothSpl

Usage

```
## S3 method for class 'smoothSpl'
plot(x, y, ..., by = 1, n = 10, index = NULL)
```

Arguments

`x`	class smoothSpl object
`y`	ignored
`...`	further arguments passed by
`by`	stepsize
`n`	length of sequence to plot
`index`	optinally the sequence instead of by and n

Author(s)

Alessia Di Blasi, Federico Pavone, Gianluca Zeni

---

pls_pb — *Function calculating a set of (D-1) principal balances based on PLS.*

Description

Function calculating a set of (D-1) principal balances based on PLS.

Usage

```
pls_pb(Xcoda, ycoda, version = "cov")
```

Arguments

`Xcoda`	a matrix of raw compositional data with "n" rows and "D" columns/components
`ycoda`	a response variable; can be continuous (PLS regression) or binary (PLS-DA)
`version`	a parameter determining whether the balances are ordered according to max. covariance (default) or max. correlation

Details

The function creates a set of (D-1) principal balances based on PLS. The procedure builds on the method building principal balances based on PCA, introduced in Martin-Fernandez et al. (2018) For detailed information regarding PLS principal balances, see Nesrstová et al. (2023).

Value

A list with the following components:

`bal` A matrix of (D-1) principal balances.

`cov` Covariance of each balance with the response variable.

Author(s)

Viktorie Nesrstová

References

J. A. Martín-Fernández, V. Pawlowsky-Glahn, J. J. Egozcue, and R. Tolosona-Delgado. Advances in principal balances for compositional data. Mathematical Geosciences, 50(3):273–298, 2018. Available at: https://link.springer.com/article/10.1007/s11004-017-9712-z DOI: doi:10.1007/s110040179712z

Nesrstová, V, Wilms, I, Palarea-Albaladejo, J, et al. Principal balances of compositional data for regression and classification using partial least squares. Journal of Chemometrics. 2023; 37(12):e3518.. Available at: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/full/10.1002/cem.3518 DOI: doi:10.1002/cem.3518

Examples

```
## Not run:
  if (requireNamespace("MASS", quietly = TRUE)) {

# 1. Generate sample data ------------------------------------------------------
n <- 100              # observations
D <- 15               # parts/variables
Sig <- diag(D-1)      # positive-definite symmetric matrix -> covariance matrix
mu <- c(rep(0, D-1))  # means of variables

set.seed(123)
# ilr coordinates
Z <- MASS::mvrnorm(n,mu,Sigma = Sig)

# Z -> CoDa X
V <- compositions::ilrBase(D = D)  # ilrBase() in library(compositions)
X <- as.matrix(as.data.frame(acomp(exp(Z%*%t(V)))))

# Response y:
beta <- runif(D-1,0.1,1)
eps <- rnorm(n)
y <- Z%*%beta+eps

# 2. Calculate PLS PBs

PLS_balances <- fBalChip_PLS(X,y,version = "cov")    # version = "cov" -> max. covariance
balances <- PLS_balances$bal
  }

## End(Not run)
```

`precipitation` *24-hour precipitation*

Description

table containing counts for 24-hour precipitation for season at the rain-gouge.

Usage

```
data(precipitation)
```

Format

A table with 4 rows and 6 columns

Details

- `spring` numeric vector on counts for different level of precipitation
- `summer` numeric vector on counts for different level of precipitation
- `autumn` numeric vector on counts for different level of precipitation
- `winter` numeric vector on counts for different level of precipitation

Author(s)

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

References

Romero R, Guijarro J A, Ramis C, Alonso S (1998). A 30-years (196493) daily rainfall data base for the Spanish Mediterranean regions: first exploratory study. *International Journal of Climatology* 18, 541560.

Examples

```
data(precipitation)
precipitation
str(precipitation)
```

---

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
## Not run:
xi <- impCoda(expenditures)
xi
summary(xi)
plot(xi, which=1:2)

## End(Not run)
```

---

production *production splitted by nationality on enterprise level*

Description

nace NACE classification, 2 digits

oenace_2008 Corresponding Austrian NACE classification (in German)

year year

month month

enterprise enterprise ID

total total ...

home home ...

EU EU ...

non-EU non-EU ...

Usage

```
data(production)
```

Format

A data frame with 535 rows and 9 variables

Author(s)

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

Source

statCube data base at the website of Statistics Austria. The product and all material contained therein are protected by copyright with all rights reserved by the Bundesanstalt Statistik Oesterreich (STATISTICS AUSTRIA). It is permitted to reproduce, distribute, make publicly available and process the content for non-commercial purposes. Prior to any use for commercial purposes a written consent of STATISTICS AUSTRIA must be obtained. Any use of the contained material must be correctly reproduced and clearly cite the source STATISTICS AUSTRIA. If tables published by STATISTICS AUSTRIA are partially used, displayed or otherwise changed, a note must be added at an adequate position to show data was extracted or adapted.

Examples

```
data(production)
str(production)
summary(production)
```

---

pTab *Propability table*

Description

Calculates the propability table using different methods

Usage

```
pTab(x, method = "dirichlet", alpha = 1/length(as.numeric(x)))
```

Arguments

x	an object of class table
method	default is 'dirichlet'. Other available methods: 'classical' that is function `prop.table()` from package base or method "half" that add 1/2 to each cell to avoid zero problems.
alpha	constant used for method 'dirichlet'

Value

The probablity table

Author(s)

Matthias Templ

References

Egozcue, J.J., Pawlowsky-Glahn, V., Templ, M., Hron, K. (2015) Independence in contingency tables using simplicial geometry. *Communications in Statistics - Theory and Methods*, 44 (18), 3978–3996.

Examples

```
data(precipitation)
pTab(precipitation)
pTab(precipitation, method = "dirichlet")
```

`rcodes` *codes for UNIDO tables*

Description

`ISOCN` ISOCN codes

`OPERATOR` Operator

`ADESC` Country

`CCODE` Country code

`CDESC` Country destination

`ACODE` Country destination code

Usage

```
data(rcodes)
```

Format

A data.frame with 2717 rows and 6 columns.

Examples

```
data(rcodes)
str(rcodes)
```

`rdcm` *relative difference between covariance matrices*

Description

The sample covariance matrices are computed from compositions expressed in the same isometric logratio coordinates.

Usage

```
rdcm(x, y)
```

Arguments

`x`	matrix or data frame
`y`	matrix or data frame of the same size as x.

Details

The difference in covariance structure is based on the Euclidean distance between both covariance estimations.

Value

the error measures value

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*, 54 (12), 3095-3107.

Templ, M. and Hron, K. and Filzmoser and Gardlo, A. (2016). Imputation of rounded zeros for high-dimensional compositional data. *Chemometrics and Intelligent Laboratory Systems*, 155, 183-190.

See Also

`rdcm`

Examples

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

`rSDev` *Relative simplicial deviance*

Description

Relative simplicial deviance

Usage

```
rSDev(x, y)
```

Arguments

x	a propability table
y	an interaction table

Value

The relative simplicial deviance

Author(s)

Matthias Templ

References

Egozcue, J.J., Pawlowsky-Glahn, V., Templ, M., Hron, K. (2015) Independence in contingency tables using simplicial geometry. *Communications in Statistics - Theory and Methods*, 44 (18), 3978–3996.

Examples

```
data(precipitation)
tabprob <- prop.table(precipitation)
tabind <- indTab(precipitation)
tabint <- intTab(tabprob, tabind)
rSDev(tabprob, tabint$intTab)
```

`rSDev.test` *Relative simplicial deviance tests*

Description

Monte Carlo based contingency table tests considering the compositional approach to contingency tables.

Usage

```
rSDev.test(x, R = 999, method = "multinom")
```

Arguments

`x`	matrix, data.frame or table
`R`	an integer specifying the number of replicates used in the Monte Carlo test.
`method`	either “rmultinom” (default) or “permutation”.

Details

Method “rmultinom” generate multinomially distributed samples from the independent probability table, which is estimated from `x` using geometric mean marginals. The relative simplicial deviance of the original data are then compared to the generated ones.

Method “permutation” permutes the entries of `x` and compares the relative simplicial deviance estimated from the original data to the ones of the permuted data (the independence table is unchanged and originates on `x`).

Method “rmultinom” should be preferred, while method “permutation” can be used for comparisons.

Value

A list with class “htest” containing the following components:

statistic the value of the relative simplicial deviance (test statistic).

method a character string indicating what type of rSDev.test was performed.

p.value the p-value for the test.

Author(s)

Matthias Templ, Karel Hron

References

Egozcue, J.J., Pawlowsky-Glahn, V., Templ, M., Hron, K. (2015) Independence in contingency tables using simplicial geometry. *Communications in Statistics - Theory and Methods*, 44 (18), 3978–3996.

See Also

rSDev

Examples

```
data(precipitation)
rSDev.test(precipitation)
```

---

saffron *saffron compositions*

Description

Stable isotope ratio and trace metal cncentration data for saffron samples.

Format

A data frame with 53 observations on the following 36 variables.

`Sample` adulterated honey, Honey or Syrup

`Country` group information

`Batch` detailed group information

`Region` less detailed group information

`d2H` region

`d13C` chemical element

`d15N` chemical element

`Li` chemical element

`B` chemical element

`Na` chemical element

`Mg` chemical element

`Al` chemical element

`K` chemical element

`Ca` chemical element

`V` chemical element

`Mn` chemical element

`Fe` chemical element

`Co` chemical element

`Ni` chemical element

`Cu` chemical element

`Zn` chemical element

`Ga` chemical element

`As` chemical element

`Rb` chemical element

`Sr` chemical element

`Y` chemical element

`Mo` chemical element

`Cd` chemical element

`Cs` chemical element

`Ba` chemical element

`Ce` chemical element

`Pr` chemical element

`Nd` chemical element

`Sm` chemical element

`Gd` chemical element

`Pb` chemical element

Note

In the original paper, the authors applied lda for classifying the observations.

Source

Mendeley Data, contributed by Russell Frew and translated to R by Matthias Templ

References

Frew, Russell (2019), Data for: CHEMICAL PROFILING OF SAFFRON FOR AUTHENTICATION OF ORIGIN, Mendeley Data, V1, doi:10.17632/5544tn9v6c.1

Examples

```
data(saffron)
```

`SDev` *Simplicial deviance*

Description

Simplicial deviance

Usage

```
SDev(x)
```

Arguments

`x` a propability table

Value

The simplicial deviance

Author(s)

Matthias Templ

References

Juan Jose Egozcuea, Vera Pawlowsky-Glahn, Matthias Templ, Karel Hron (2015) Independence in Contingency Tables Using Simplicial Geometry. *Communications in Statistics - Theory and Methods*, Vol. 44 (18), 3978–3996. DOI:10.1080/03610926.2013.824980

Examples

```
data(precipitation)
tab1prob <- prop.table(precipitation)
SDev(tab1prob)
```

---

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.

Details

`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

Author(s)

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

References

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

Examples

```
data(skyeLavas)
str(skyeLavas)
summary(skyeLavas)
rowSums(skyeLavas)
```

---

smoothSplines *Estimate density from histogram*

Description

Given raw (discretized) distributional observations, `smoothSplines` computes the density function that 'best' fits data, in a trade-off between smooth and least squares approximation, using B-spline basis functions.

Usage

```
smoothSplines(
  k,
  l,
  alpha,
  data,
  xcp,
  knots,
  weights = matrix(1, dim(data)[1], dim(data)[2]),
  num_points = 100,
  prior = "default",
  cores = 1,
  fast = 0
)
```

Arguments

`k`	smoothing splines degree
`l`	order of derivative in the penalization term
`alpha`	weight for penalization
`data`	an object of class "matrix" containing data to be smoothed, row by row
`xcp`	vector of control points
`knots`	either vector of knots for the splines or a integer for the number of equispaced knots. The inner and outer knot must be outside the data range.

weights	matrix of weights. If not given, all data points will be weighted the same.
num_points	number of points of the grid where to evaluate the density estimated
prior	prior used for zero-replacements. This must be one of "perks", "jeffreys", "bayes_laplace", "sq" or "default"
cores	number of cores for parallel execution, if the option was enabled before installing the package
fast	1 if maximal performance is required (print statements suppressed), 0 otherwise

Details

The original discretized densities are not directly smoothed, but instead the centred logratio transformation is first applied, to deal with the unit integral constraint related to density functions.
Then the constrained variational problem is set. This minimization problem for the optimal density is a compromise between staying close to the given data, at the corresponding `xcp`, and obtaining a smooth function. The non-smoothness measure takes into account the `l`th derivative, while the fidelity term is weigthed by `alpha`.
The solution is a natural spline. The vector of its coefficients is obtained by the minimum norm solution of a linear system. The resulting splines can be either back-transformed to the original Bayes space of density functions (in order to provide their smoothed counterparts for vizualization and interpretation purposes), or retained for further statistical analysis in the clr space.

Value

An object of class `smoothSpl`, containing among the other the following variables:

bspline	each row is the vector of B-spline coefficients
Y	the values of the smoothed curve, for the grid given
Y_clr	the values of the smoothed curve, in the clr setting, for the grid given

Author(s)

Alessia Di Blasi, Federico Pavone, Gianluca Zeni, Matthias Templ

References

J. Machalova, K. Hron & G.S. Monti (2016): Preprocessing of centred logratio transformed density functions using smoothing splines. Journal of Applied Statistics, 43:8, 1419-1435.

Examples

```
SepalLengthCm <- iris$Sepal.Length
Species <- iris$Species

iris1 <- SepalLengthCm[iris$Species==levels(iris$Species)[1]]
h1 <- hist(iris1, nclass = 12, plot = FALSE)

midx1 <- h1$mids
midy1 <- matrix(h1$density, nrow=1, ncol = length(h1$density), byrow=TRUE)
knots <- 7
```

```
## Not run:
sol1 <- smoothSplines(k=3,l=2,alpha=1000,midy1,midx1,knots)
plot(sol1)

h1 <- hist(iris1, freq = FALSE, nclass = 9, xlab = "Sepal Length [cm]", main = "Iris setosa")
# black line: kernel method; red line: smoothSplines result
lines(density(iris1), col = "black", lwd = 1.5)
xx1 <- seq(sol1$Xcp[1],tail(sol1$Xcp,n=1),length.out = sol1$NumPoints)
lines(xx1,sol1$Y[1,], col = 'red', lwd = 2)

sol2 <- smoothSplines(k=3, l=2, alpha=1000, data = midy1, xcp = midx1,
         knots = seq(4.33, 5.76, length.out = 7))
plot(sol2)
h1 <- hist(iris1, freq = FALSE, nclass = 12, xlab = "Sepal Length [cm]", main = "Iris setosa")
lines(density(iris1), col = "black", lwd = 1.5)
xx1 <- seq(sol2$Xcp[1],tail(sol2$Xcp,n=1),length.out = sol1$NumPoints)
lines(xx1,sol2$Y[1,], col = 'red', lwd = 2)

## End(Not run)
```

---

smoothSplinesVal — *Estimate density from histogram - for different* `alpha`

Description

As `smoothSplines`, `smoothSplinesVal` computes the density function that 'best' fits discretized distributional data, using B-spline basis functions, for different `alpha`.
Comparing and choosing an appropriate `alpha` is the ultimate goal.

Usage

```
smoothSplinesVal(
  k,
  l,
  alpha,
  data,
  xcp,
  knots,
  weights = matrix(1, dim(data)[1], dim(data)[2]),
  prior = "default",
  cores = 1
)
```

Arguments

`k`	smoothing splines degree
`l`	order of derivative in the penalization term
`alpha`	vector of weights for penalization

`data`	an object of class "matrix" containing data to be smoothed, row by row
`xcp`	vector of control points
`knots`	either vector of knots for the splines or a integer for the number of equispaced knots
`weights`	matrix of weights. If not gives, all data points will be weighted the same.
`prior`	prior used for zero-replacements. This must be one of "perks", "jeffreys", "bayes_laplace", "sq" or "default"
`cores`	number of cores for parallel execution

Details

See `smoothSplines` for the description of the algorithm.

Value

A list of three objects:

`alpha`	the values of `alpha`
`J`	the values of the functional evaluated in the minimizing
`CV-error`	the values of the leave-one-out CV-error

Author(s)

Alessia Di Blasi, Federico Pavone, Gianluca Zeni, Matthias Templ

References

J. Machalova, K. Hron & G.S. Monti (2016): Preprocessing of centred logratio transformed density functions using smoothing splines. Journal of Applied Statistics, 43:8, 1419-1435.

Examples

```
SepalLengthCm <- iris$Sepal.Length
Species <- iris$Species

iris1 <- SepalLengthCm[iris$Species==levels(iris$Species)[1]]
h1 <- hist(iris1, nclass = 12, plot = FALSE)

## Not run:
midx1 <- h1$mids
midy1 <- matrix(h1$density, nrow=1, ncol = length(h1$density), byrow=TRUE)
knots <- 7
sol1 <- smoothSplinesVal(k=3,l=2,alpha=10^seq(-4,4,by=1),midy1,midx1,knots,cores=1)

## End(Not run)
```

socExp *social expenditures*

Description

Social expenditures according to source (public or private) and three important branches (health, old age, incapacity related) in selected OECD countries in 2010. Expenditures are always provided in the respective currency.

Usage

```
data(socExp)
```

Format

A data frame with 20 observations on the following 8 variables (country + currency + row-wise sorted cells of 2x3 compositional table).

Details

`country` Country of origin

`currency` Currency unit (in Million)

`health-public` Health from the public

`old-public` Old age expenditures from the public

`incap-public` Incapacity related expenditures from the public

`health-private` Health from private sources

`old-private` Old age expenditures from private sources

`incap-private` Incapacity related expenditures from private sources

Author(s)

conversion to R by Karel Hron Karel Hron and modifications by Matthias Templ <matthias.templ@tuwien.ac.at>

References

OECD

Examples

```
data(socExp)
str(socExp)
rowSums(socExp[, 3:ncol(socExp)])
```

spca_logrs *Function making a matrix of D(D-1) logratios and calculating sparse PCA.*

Description

Function making a matrix of D(D-1) logratios and calculating sparse PCA.

Usage

```
spca_logrs(X, alpha = 0.01, beta = 1e-04, k = (D - 1), draw = T)
```

Arguments

X	a matrix of raw compositional data with "n" rows and "D" columns/components
alpha	a sparsity parameter; the higher its value, the sparser the results; default is 0.01
beta	a tuning parameter resulting in shrinkage of the parameters towards zero; beta = 0 leads to lasso penalty; default is 1e-04
k	number of principal components (PCs) to be calculated; default is (D-1)
draw	a logical parameter stating whether a biplot should be drawn (TRUE) or not (FALSE); default is T

Details

The function creates a sparse PCA model where a matrix of pairwise logratios is taken as an input. The function spca from the library sparsepca is used for modelling, Erichson et al. (2020) for more details. For detailed information regarding sparse PCA with pairwise logratios, see Nesrstová et al. (2024).

Value

A list with the following components:

`X.pairwise` A matrix of (D-1) pairwise logratios.

`model` A sparse PCA model (using sparsepca::spca) where X.pairwise is the input.

`loadings` A matrix of loadings.

`model summary` A short summary of the model returning the explained variance by PCs.

`expl.var` A proportion of variance of each PC.

`number of zero logratios` States how many zero logratios (having zeros in all PCs) are in the model.

`table of all zero` Returns the table of all zero logratios.

Author(s)

Viktorie Nesrstová

References

Erichson, N.B., Zheng, P. Manohar, K., Brunton, S.L., Kuntz, J.N., Aravkin, Y. (2020). Sparse principal component analysis via variable projection. SIAM J Appl Math. Available at: https://epubs.siam.org/doi/10.1137/18M1211350 DOI: doi:10.1137/18M1211350

Nesrstová, V., Wilms, I., Hron, K., Filzmoser, P. (2024). Identifying Important Pairwise Logratios in Compositional Data with Sparse Principal Component Analysis. Mathematical Geosciences. Available at: https://link.springer.com/article/10.1007/s11004-024-10159-0 DOI: doi:10.1007/s11004024101590

Examples

```
## Not run:
  if (requireNamespace("MASS", quietly = TRUE)) {

# 1. Generate sample data
n <- 100              # observations
D <- 10               # parts/variables
Sig <- diag(D-1)      # positive-definite symmetric matrix -> covariance matrix
mu <- c(rep(0, D-1))  # means of variables

set.seed(1234)
# ilr coordinates
Z <- MASS::mvrnorm(n,mu,Sigma = Sig)

# Z -> CoDa X
V <- compositions::ilrBase(D = D)
X <- as.matrix(as.data.frame(acomp(exp(Z%*%t(V)))))

# 2. Apply sPCA to pairwise logratios
alpha_max <- 1 # specify max value of tuning parameter
alpha_nbr <- 50 # specify number of tuning parameters
alpha_ratio <- 1000 # specify ratio of largest to smallest tuning parameter
a <- sort(alpha_grid,decreasing=F)        # zero included

# Models for different values of alpha parameters, calculating PC1 and PC2
models <- list()
for(i in 1:length(a)){
 models[[i]] <- spca_logrs(X=X, alpha = a[i], k = 2, draw = F)
}

  }

## End(Not run)
```

---

stats   *Classical estimates for tables*

Description

Some standard/classical (non-compositional) statistics

Usage

```
stats(
  x,
  margins = NULL,
  statistics = c("phi", "cramer", "chisq", "yates"),
  maggr = mean
)
```

Arguments

`x`	a data.frame, matrix or table
`margins`	margins
`statistics`	statistics of interest
`maggr`	a function for calculating the mean margins of a table, default is the arithmetic mean

Details

statistics ‘phi’ is the values of the table divided by the product of margins. ‘cramer’ normalize these values according to the dimension of the table. ‘chisq’ are the expected values according to Pearson while ‘yates’ according to Yates.

For the `maggr` function argument, arithmetic means (`mean`) should be chosen to obtain the classical results. Any other user-provided functions should be take with care since the classical estimations relies on the arithmetic mean.

Value

List containing all statistics

Author(s)

Matthias Templ

References

Egozcue, J.J., Pawlowsky-Glahn, V., Templ, M., Hron, K. (2015) Independence in contingency tables using simplicial geometry. *Communications in Statistics - Theory and Methods*, 44 (18), 3978–3996.

Examples

```
data(precipitation)
tab1 <- indTab(precipitation)
stats(precipitation)
stats(precipitation, statistics = "cramer")
stats(precipitation, statistics = "chisq")
stats(precipitation, statistics = "yates")

## take with care
## (the provided statistics are not designed for that case):
stats(precipitation, statistics = "chisq", maggr = gmean)
```

---

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)
```

---

tabCoord *Coordinate representation of compositional tables and a sample of compositional tables*

Description

tabCoord computes a system of orthonormal coordinates of a compositional table. Computation of either pivot coordinates or a coordinate system based on the given SBP is possible.

tabCoordWrapper: For each compositional table in the sample `tabCoordWrapper` computes a system of orthonormal coordinates and provide a simple descriptive analysis. Computation of either pivot coordinates or a coordinate system based on the given SBP is possible.

Usage

```
tabCoord(
  x = NULL,
  row.factor = NULL,
  col.factor = NULL,
  value = NULL,
  SBPr = NULL,
  SBPc = NULL,
  pivot = FALSE,
  print.res = FALSE
)

tabCoordWrapper(
  X,
  obs.ID = NULL,
  row.factor = NULL,
  col.factor = NULL,
  value = NULL,
  SBPr = NULL,
  SBPc = NULL,
  pivot = FALSE,
  test = FALSE,
  n.boot = 1000
)
```

Arguments

`x` a data frame containing variables representing row and column factors of the respective compositional table and variable with the values of the composition.

`row.factor` name of the variable representing the row factor. Needs to be stated with the quotation marks.

`col.factor` name of the variable representing the column factor. Needs to be stated with the quotation marks.

`value` name of the variable representing the values of the composition. Needs to be stated with the quotation marks.

`SBPr` an $I - 1 \times I$ array defining the sequential binary partition of the values of the row factor, where I is the number of the row factor levels. The values assigned in the given step to the + group are marked by 1, values from the - group by -1 and the rest by 0. If it is not provided, the pivot version of coordinates is constructed automatically.

`SBPc` an $J - 1 \times J$ array defining the sequential binary partition of the values of the column factor, where J is the number of the column factor levels. The values assigned in the given step to the + group are marked by 1, values from the - group by -1 and the rest by 0. If it is not provided, the pivot version of coordinates is constructed automatically.

`pivot` logical, default is FALSE. If TRUE, or one of the SBPs is not defined, its pivot version is used.

`print.res` logical, default is FALSE. If TRUE, the output is displayed in the Console.

`X` a data frame containing variables representing row and column factors of the respective compositional tables, variable with the values of the composition and variable distinguishing the observations.

`obs.ID` name of the variable distinguishing the observations. Needs to be stated with the quotation marks.

`test` logical, default is `FALSE`. If `TRUE`, the bootstrap analysis of coordinates is provided.

`n.boot` number of bootstrap samples.

Details

tabCoord

This transformation moves the IJ-part compositional tables from the simplex into a (IJ-1)-dimensional real space isometrically with respect to its two-factorial nature. The coordinate system is formed by two types of coordinates - balances and log odds-ratios.

tabCoordWrapper: Each of n IJ-part compositional tables from the sample is with respect to its two-factorial nature isometrically transformed from the simplex into a (IJ-1)-dimensional real space. Sample mean values and standard deviations are computed and using bootstrap an estimate of 95 % confidence interval is given.

Value

Coordinates	an array of orthonormal coordinates.
Grap.rep	graphical representation of the coordinates. Parts denoted by + form the groups in the numerator of the respective computational formula, parts - form the denominator and parts . are not involved in the given coordinate.
Ind.coord	an array of row and column balances. Coordinate representation of the independent part of the table.
Int.coord	an array of OR coordinates. Coordinate representation of the interactive part of the table.
Contrast.matrix	contrast matrix.
Log.ratios	an array of pure log-ratios between groups of parts without the normalizing constant.
Coda.table	table form of the given composition.
Bootstrap	array of sample means, standard deviations and bootstrap confidence intervals.
Tables	Table form of the given compositions.

Author(s)

Kamila Facevicova

References

Facevicova, K., Hron, K., Todorov, V. and M. Templ (2018) General approach to coordinate representation of compositional tables. Scandinavian Journal of Statistics, 45(4), 879–899.

See Also

cubeCoord cubeCoordWrapper

Examples

```
###################
### Coordinate representation of a CoDa Table

# example from Fa\v cevicov\'a (2018):
data(manu_abs)
manu_USA <- manu_abs[which(manu_abs$country=='USA'),]
manu_USA$output <- factor(manu_USA$output, levels=c('LAB', 'SUR', 'INP'))

# pivot coordinates
tabCoord(manu_USA, row.factor = 'output', col.factor = 'isic', value='value')

# SBPs defined in paper
r <- rbind(c(-1,-1,1), c(-1,1,0))
c <- rbind(c(-1,-1,-1,-1,1), c(-1,-1,-1,1,0), c(-1,-1,1,0,0), c(-1,1,0,0,0))
tabCoord(manu_USA, row.factor = 'output', col.factor = 'isic', value='value', SBPr=r, SBPc=c)
```

```
###################
### Analysis of a sample of CoDa Tables

# example from Fa\v cevicov\'a (2018):
data(manu_abs)

### Compositional tables approach,
### analysis of the relative structure.
### An example from Facevi\v cov\'a (2018)

manu_abs$output <- factor(manu_abs$output, levels=c('LAB', 'SUR', 'INP'))

# pivot coordinates
tabCoordWrapper(manu_abs, obs.ID='country',
row.factor = 'output', col.factor = 'isic', value='value')

# SBPs defined in paper
r <- rbind(c(-1,-1,1), c(-1,1,0))
c <- rbind(c(-1,-1,-1,-1,1), c(-1,-1,-1,1,0),
c(-1,-1,1,0,0), c(-1,1,0,0,0))
tabCoordWrapper(manu_abs, obs.ID='country',row.factor = 'output',
col.factor = 'isic', value='value', SBPr=r, SBPc=c, test=TRUE)

### Classical approach,
### generalized linear mixed effect model.

## Not run:
library(lme4)
glmer(value~output*as.factor(isic)+(1|country),data=manu_abs,family=poisson)

## End(Not run)
```

`teachingStuff` *teaching stuff*

Description

Teaching stuff in selected countries

Format

A (tidy) data frame with 1216 observations on the following 4 variables.

`country` Country of origin

`subject` school type: primary, lower secondary, higher secondary and tertiary

`year` Year

`value` Number of stuff

Details

Teaching staff include professional personnel directly involved in teaching students, including classroom teachers, special education teachers and other teachers who work with students as a whole class, in small groups, or in one-to-one teaching. Teaching staff also include department chairs of whose duties include some teaching, but it does not include non-professional personnel who support teachers in providing instruction to students, such as teachers' aides and other paraprofessional personnel. Academic staff include personnel whose primary assignment is instruction, research or public service, holding an academic rank with such titles as professor, associate professor, assistant professor, instructor, lecturer, or the equivalent of any of these academic ranks. The category includes personnel with other titles (e.g. dean, director, associate dean, assistant dean, chair or head of department), if their principal activity is instruction or research.

Author(s)

translated from `https://data.oecd.org/` and restructured by Matthias Templ

Source

OECD: `https://data.oecd.org/`

References

OECD (2017), Teaching staff (indicator). doi: 10.1787/6a32426b-en (Accessed on 27 March 2017)

Examples

```
data(teachingStuff)
str(teachingStuff)
```

`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),
  text = NULL,
  grid = TRUE,
  gridCol = grey(0.6),
  mcex = 1.2,
  line = "none",
```

```
  robust = TRUE,
  group = NULL,
  tol = 0.975,
  ...
)
```

Arguments

`x`	matrix or data.frame with 3 columns
`name`	names of the variables
`text`	default NULL, text for each point can be provided
`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`>>, Matthias Templ <<`matthias.templ@fhnw.ch`>>

References

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

Examples

```
data(arcticLake)
ternaryDiag(arcticLake)

data(coffee)
x <- coffee[,2:4]
grp <- as.integer(coffee[,1])
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[,2:4]
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[,2:4]
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, ...)
```

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[,2:4]
ternaryDiag(x, grid=FALSE)
ternaryDiagPoints(x+1, col="red", pch=2)
```

trapzc *Trapezoidal formula for numerical integration*

Description

Numerical integration via trapezoidal formula.

Usage

```
trapzc(step, f)
```

Arguments

step	step of the grid
f	grid evaluation of density

Value

int	The value of integral computed numerically by trapezoidal formula.

Author(s)

R. Talska<talskarenata@seznam.cz>, K. Hron<karel.hron@upol.cz>

Examples

```
# Example (zero-integral of fcenLR density)
t = seq(-4.7,4.7, length = 1000)
t_step = diff(t[1:2])
mean = 0; sd = 1.5
f = dnorm(t, mean, sd)
f.fcenLR = fcenLR(t,t_step,f)
trapzc(t_step,f.fcenLR)
```

trondelagC *regional geochemical survey of soil C in Norway*

Description

A regional-scale geochemical survey of C horizon samples in Nord-Trondelag, Central Norway

Usage

```
data(trondelagC)
```

Format

A data frame with 754 observations and 70 variables

Details

`X.S_ID` ID

`X.Loc_ID` ID

`longitude` longitude in WGS84

`latitude` latitude in WGS84

`E32wgs` UTM zone east

`N32wgs` UTM zone north

`X.Medium`

`Ag` Concentration of silver (in mg/kg)

`Al` Concentration of aluminum (in mg/kg)

`As` Concentration of arsenic (in mg/kg)

`Au` Concentration of gold (in mg/kg)

`B` Concentration of boron (in mg/kg)

`Ba` Concentration of barium (in mg/kg)

`Be` Concentration of beryllium (in mg/kg)

`Bi` Concentration of bismuth (in mg/kg)

`Ca` Concentration of calzium (in mg/kg)

`Cd` Concentration of cadmium (in mg/kg)

`Ce` Concentration of cerium (in mg/kg)

`Co` Concentration of cobalt (in mg/kg)

`Cr` Concentration of chromium (in mg/kg)

`Cs` Concentration of cesium (in mg/kg)

`Cu` Concentration of copper (in mg/kg)

`Fe` Concentration of iron (in mg/kg)

`Ga` Concentration of gallium (in mg/kg)

`Ge` Concentration of germanium (in mg/kg)

`Hf` Concentration of hafnium (in mg/kg)

`Hg` Concentration of mercury (in mg/kg)

`In` Concentration of indium (in mg/kg)

`K` Concentration of pottasium (in mg/kg)

`La` Concentration of lanthanum (in mg/kg)

`Li` Concentration of lithium (in mg/kg)

`Mg` Concentration of magnesium (in mg/kg)

`Mn` Concentration of manganese (in mg/kg)

Mo Concentration of molybdenum (in mg/kg)

Na Concentration of sodium (in mg/kg)

Nb Concentration of niobium (in mg/kg)

Ni Concentration of nickel (in mg/kg)

P Concentration of phosphorus (in mg/kg)

Pb Concentration of lead (in mg/kg)

Pb204 Concentration of lead, 204 neutrons (in mg/kg)

Pb206 Concentration of lead, 206 neutrons (in mg/kg)

Pb207 Concentration of lead, 207 neutrons (in mg/kg)

Pb208 Concentration of lead, 208 neutrons (in mg/kg)

X6_7Pb Concentration of lead (in mg/kg)

X7_8Pb Concentration of lead (in mg/kg)

X6_4Pb Concentration of lead (in mg/kg)

X7_4Pb Concentration of lead (in mg/kg)

X8_4Pb Concentration of lead (in mg/kg)

Pd Concentration of palladium (in mg/kg)

Pt Concentration of platium (in mg/kg)

Rb Concentration of rubidium (in mg/kg)

Re Concentration of rhenium (in mg/kg)

S Concentration of sulfur (in mg/kg)

Sb Concentration of antimony (in mg/kg)

Sc Concentration of scandium (in mg/kg)

Se Concentration of selenium (in mg/kg)

Sn Concentration of tin (in mg/kg)

Sr Concentration of strontium (in mg/kg)

Ta Concentration of tantalum (in mg/kg)

Te Concentration of tellurium (in mg/kg)

Th Concentration of thorium (in mg/kg)

Ti Concentration of titanium (in mg/kg)

Tl Concentration of thalium (in mg/kg)

U Concentration of uranium (in mg/kg)

V Concentration of vanadium (in mg/kg)

W Concentration of tungsten (in mg/kg)

Y Concentration of yttrium (in mg/kg)

Zn Concentration of zinc (in mg/kg)

Zr Concentration of zirconium (in mg/kg)

The samples were analysed using aqua regia extraction. Sampling was based on a 6.6km grid, i.e. 1 sample site/36 km2.

Author(s)

NGU, https://www.ngu.no, transfered to R by Matthias Templ <matthias.templ@tuwien.ac.at>

References

C.Reimann, J.Schilling, D.Roberts, K.Fabian. A regional-scale geochemical survey of soil C horizon samples in Nord-Trondelag, Central Norway. Geology and mineral potential, *Applied Geochemistry* 61 (2015) 192-205.

Examples

```
data(trondelagC)
str(trondelagC)
```

`trondelagO` *regional geochemical survey of soil O in Norway*

Description

A regional-scale geochemical survey of O horizon samples in Nord-Trondelag, Central Norway

Usage

```
data(trondelagO)
```

Format

A data frame with 754 observations and 70 variables

Details

`X.Loc_ID` ID

`LITHO` Rock type

`longitude` langitude in WGS84

`latitude` latitude in WGS84

`E32wgs` UTM zone east

`N32wgs` UTM zone north

`X.Medium` a numeric vector

`Alt_masl` a numeric vector

`LOI_480` Loss on ignition

`pH` Numeric scale used to specify the acidity or alkalinity of an aqueous solution

`Ag` Concentration of silver (in mg/kg)

`Al` Concentration of aluminum (in mg/kg)

`As` Concentration of arsenic (in mg/kg)

Au Concentration of gold (in mg/kg)

B Concentration of boron (in mg/kg)

Ba Concentration of barium (in mg/kg)

Be Concentration of beryllium (in mg/kg)

Bi Concentration of bismuth (in mg/kg)

Ca Concentration of calzium (in mg/kg)

Cd Concentration of cadmium (in mg/kg)

Ce Concentration of cerium (in mg/kg)

Co Concentration of cobalt (in mg/kg)

Cr Concentration of chromium (in mg/kg)

Cs Concentration of cesium (in mg/kg)

Cu Concentration of copper (in mg/kg)

Fe Concentration of iron (in mg/kg)

Ga Concentration of gallium (in mg/kg)

Ge Concentration of germanium (in mg/kg)

Hf Concentration of hafnium (in mg/kg)

Hg Concentration of mercury (in mg/kg)

In Concentration of indium (in mg/kg)

K Concentration of pottasium (in mg/kg)

La Concentration of lanthanum (in mg/kg)

Li Concentration of lithium (in mg/kg)

Mg Concentration of magnesium (in mg/kg)

Mn Concentration of manganese (in mg/kg)

Mo Concentration of molybdenum (in mg/kg)

Na Concentration of sodium (in mg/kg)

Nb Concentration of niobium (in mg/kg)

Ni Concentration of nickel (in mg/kg)

P Concentration of phosphorus (in mg/kg)

Pb Concentration of lead (in mg/kg)

Pb204 Concentration of lead, 204 neutrons (in mg/kg)

Pb206 Concentration of lead, 206 neutrons (in mg/kg)

Pb207 Concentration of lead, 207 neutrons (in mg/kg)

Pb208 Concentration of lead, 208 neutrons (in mg/kg)

X6_7Pb Concentration of lead (in mg/kg)

X7_8Pb Concentration of lead (in mg/kg)

X6_4Pb Concentration of lead (in mg/kg)

X7_4Pb Concentration of lead (in mg/kg)

`X8_4Pb` Concentration of lead (in mg/kg)

`Pd` Concentration of palladium (in mg/kg)

`Pt` Concentration of platium (in mg/kg)

`Rb` Concentration of rubidium (in mg/kg)

`Re` Concentration of rhenium (in mg/kg)

`S` Concentration of sulfur (in mg/kg)

`Sb` Concentration of antimony (in mg/kg)

`Sc` Concentration of scandium (in mg/kg)

`Se` Concentration of selenium (in mg/kg)

`Sn` Concentration of tin (in mg/kg)

`Sr` Concentration of strontium (in mg/kg)

`Ta` Concentration of tantalum (in mg/kg)

`Te` Concentration of tellurium (in mg/kg)

`Th` Concentration of thorium (in mg/kg)

`Ti` Concentration of titanium (in mg/kg)

`Tl` Concentration of thalium (in mg/kg)

`U` Concentration of uranium (in mg/kg)

`V` Concentration of vanadium (in mg/kg)

`W` Concentration of tungsten (in mg/kg)

`Y` Concentration of yttrium (in mg/kg)

`Zn` Concentration of zinc (in mg/kg)

`Zr` Concentration of zirconium (in mg/kg)

The samples were analysed using aqua regia extraction. Sampling was based on a 6.6km grid, i.e. 1 sample site/36 km2.

Author(s)

NGU, https://www.ngu.no, transfered to R by Matthias Templ <matthias.templ@tuwien.ac.at>

References

C.Reimann, J.Schilling, D.Roberts, K.Fabian. A regional-scale geochemical survey of soil C horizon samples in Nord-Trondelag, Central Norway. Geology and mineral potential, *Applied Geochemistry* 61 (2015) 192-205.

Examples

```
data(trondelagO)
str(trondelagO)
```

`unemployed` *unemployed of young people*

Description

Youth not in employment, education or training (NEET) in 43 countries from 1997 till 2015

Format

A (tidy) data frame with 1216 observations on the following 4 variables.

`country` Country of origin

`age` age group

`year` Year

`value` percentage of unemployed

Details

This indicator presents the share of young people who are not in employment, education or training (NEET), as a percentage of the total number of young people in the corresponding age group, by gender. Young people in education include those attending part-time or full-time education, but exclude those in non-formal education and in educational activities of very short duration. Employment is defined according to the OECD/ILO Guidelines and covers all those who have been in paid work for at least one hour in the reference week of the survey or were temporarily absent from such work. Therefore NEET youth can be either unemployed or inactive and not involved in education or training. Young people who are neither in employment nor in education or training are at risk of becoming socially excluded - individuals with income below the poverty-line and lacking the skills to improve their economic situation.

Author(s)

translated from https://data.oecd.org/ and restructured by Matthias Templ

Source

OECD: https://data.oecd.org/

References

OECD (2017), Youth not in employment, education or training (NEET) (indicator). doi: 10.1787/72d1033a-en (Accessed on 27 March 2017)

Examples

```
data(unemployed)
str(unemployed)
```

`variation` *Robust and classical variation matrix*

Description

Estimates the variation matrix with robust methods.

Usage

```
variation(x, method = "robustPivot", algorithm = "MCD")
```

Arguments

`x`	data frame or matrix with positive entries
`method`	method used for estimating covariances. See details.
`algorithm`	kind of robust estimator (MCD or MM)

Details

The variation matrix is estimated for a given compositional data set. Instead of using the classical standard deviations the miniminm covariance estimator is used (`covMcd`) is used when parameter robust is set to TRUE.

For method `robustPivot` forumala 5.8. of the book (see second reference) is used. Here robust (mcd-based) covariance estimation is done on pivot coordinates. Method `robustPairwise` uses a mcd covariance estimation on pairwise log-ratios. Methods `Pivot` (see second reference) and `Pairwise` (see first reference) are the non-robust counterparts. Naturally, `Pivot` and `Pairwise` gives the same results, but the computational time is much less for method `Pairwise`.

Value

The (robust) variation matrix.

Author(s)

Karel Hron, 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.

#' Filzmoser, P., Hron, K., Templ, M. (2018) *Applied Compositional Data Analysis*. Springer, Cham.

Examples

```
data(expenditures)
variation(expenditures) # default is method "robustPivot"
variation(expenditures, method = "Pivot")
variation(expenditures, method = "robustPairwise")
variation(expenditures, method = "Pairwise") # same results as Pivot
```

`weightedPivotCoord` *Weighted pivot coordinates*

Description

Weighted pivot coordinates as a special case of isometric logratio coordinates.

Usage

```
weightedPivotCoord(
  x,
  pivotvar = 1,
  option = "var",
  method = "classical",
  pow = 1,
  yvar = NULL
)
```

Arguments

`x`	object of class ‘data.frame‘ or ‘matrix‘; positive values only
`pivotvar`	pivotal variable; if any other number than 1, the data are resorted in that sense that pivotvar is shifted to the first part
`option`	Option for the choice of weights. If `option = "var"` (default), weights are based on variation matrix elements: $(1/t_{1j})^{\mathrm{pow}}$. If `option = "cor"`, weights are based on correlations between the variable specified in `yvar` and the logratios: $\left\lvert\int_0^{r_j} f(x)\,dx\right\rvert$, where $f(x)$ is the kernel density estimator for s_j; $s_j = 0$ if $\lvert r_j\rvert < \mathrm{cut}$, otherwise $s_j = r_j$. The cutoff is calculated as: $$\mathrm{cut} = \min\left(\frac{\#r_j \geq 0}{\#r_j}, \frac{\#r_j < 0}{\#r_j}\right)$$ using a Gaussian kernel function with bandwidth $h = 0.05$.
`method`	method for estimation of variation/correlation, if ‘option = "classical"‘ (default), classical estimation is applied, if ‘option = "robust"‘, robust estimation is applied;
`pow`	if ‘option = "var"‘, power ‘pow‘ is applied on unnormalized weights; default is 1;
`yvar`	if ‘option = "cor"‘, weights are based on correlation between logratios and variable specified in ‘yvar‘;

Details

Weighted pivot coordinates map D-part compositional data from the simplex into a (D-1)-dimensional real space isometrically. The relevant relative information about one of parts is contained in the first coordinate. Unlike in the (ordinary) pivot coordinates, the pairwise logratios aggregated into the first coordinate are weighted according to their relevance for the purpose of the analysis.

Value

`WPC`	weighted pivot coordinates (matrix with n rows and (D-1) columns)
`w`	logcontrasts (matrix with D rows and (D-1) columns)

Author(s)

Nikola Stefelova

References

Hron K, Filzmoser P, de Caritat P, Fiserova E, Gardlo A (2017) Weighted 'pivot coordinates for compositional data and their application to geochemical mapping. Mathematical Geosciences 49(6):797-814.

Stefelova N, Palarea-Albaladejo J, and Hron K (2021) Weighted pivot coordinates for PLS-based marker discovery in high-throughput compositional data. Statistical Analysis and Data Mining: The ASA Data Science Journal 14(4):315-330.

See Also

pivotCoord

Examples

```
###################
data(phd)
x <- phd[, 7:ncol(phd)]
x[x == 0] <- 0.1 # better: impute with one
                 # of the zero imputation methods
                 # from robCompositions

# first variable as pivotal, weights based on variation matrix
wpc_var <- weightedPivotCoord(x)
coordinates <- wpc_var$WPC
logcontrasts <- wpc_var$w

# third variable as pivotal, weights based on variation matrix,
# robust estimation of variance, effect of weighting enhanced
wpc_var <- weightedPivotCoord(x, pivotvar = 3, method = "robust", pow = 2)
coordinates = wpc_var$WPC
logcontrasts = wpc_var$w

# first variable as pivotal, weights based on correlation between pairwise logratios and y
wpc_cor <- weightedPivotCoord(x, option = "cor", yvar = phd$female)
```

```
coordinates <- wpc_cor$WPC
logcontrasts <- wpc_cor$w

# fifth variable as pivotal, weights based on correlation between pairwise logratios
# and y, robust estimation of correlation
wpc_cor <- weightedPivotCoord(x, pivotvar = 5, option = "cor", method = "robust", yvar = phd$female)
coordinates <- wpc_cor$WPC
logcontrasts <- wpc_cor$w
```

ZBsplineBasis *ZB-spline basis*

Description

Spline basis system having zero-integral on I=[a,b] of the L^2_0 space (called ZB-splines) has been proposed for an basis representation of fcenLR transformed probability density functions. The ZB-spline basis functions can be back transformed to Bayes spaces using inverse of fcenLR transformation, resulting in compositional B-splines (CB-splines), and forming a basis system of the Bayes spaces.

Usage

```
ZBsplineBasis(t, knots, order, basis.plot = FALSE)
```

Arguments

`t`	a vector of argument values at which the ZB-spline basis functions are to be evaluated
`knots`	sequence of knots
`order`	order of the ZB-splines (i.e., degree + 1)
`basis.plot`	if TRUE, the ZB-spline basis system is plotted

Value

`ZBsplineBasis`	matrix of ZB-spline basis functions evaluated at a vector of argument values t
`nbasis`	number of ZB-spline basis functions

Author(s)

J. Machalova `<jitka.machalova@upol.cz>`, R. Talska `<talskarenata@seznam.cz>`

References

Machalova, J., Talska, R., Hron, K. Gaba, A. Compositional splines for representation of density functions. *Comput Stat* (2020). https://doi.org/10.1007/s00180-020-01042-7

Examples

```
# Example: ZB-spline basis functions evaluated at a vector of argument values t
t = seq(0,20,l=500)
knots = c(0,2,5,9,14,20)
order = 4

ZBsplineBasis.out = ZBsplineBasis(t,knots,order, basis.plot=TRUE)

# Back-transformation of ZB-spline basis functions from L^2_0 to Bayes space ->
# CB-spline basis functions
CBsplineBasis=NULL
for (i in 1:ZBsplineBasis.out$nbasis)
{
 CB_spline = fcenLRinv(t,diff(t)[1:2],ZBsplineBasis.out$ZBsplineBasis[,i])
 CBsplineBasis = cbind(CBsplineBasis,CB_spline)
}

matplot(t,CBsplineBasis, type="l",lty=1, las=1,
  col=rainbow(ZBsplineBasis.out$nbasis), xlab="t",
  ylab="CB-spline basis",
cex.lab=1.2,cex.axis=1.2)
abline(v=knots, col="gray", lty=2)
```

`zeroOut` *Detection of outliers of zero-inflated data*

Description

detects outliers in compositional zero-inflated data

Usage

```
zeroOut(x, impute = "knn")
```

Arguments

`x`	a data frame
`impute`	imputation method internally used

Details

XXX

Value

XXX

Author(s)

Matthias Templ

Examples

```
### Installing and loading required packages
data(expenditures)
```

Index