Package 'Compositional'

June 13, 2025

Type Package

Title Compositional Data Analysis

Version 7.5

Date 2025-06-13

Author Michail Tsagris [aut, cre],
Giorgos Athineou [aut],
Abdulaziz Alenazi [ctb],
Christos Adam [ctb]

Maintainer Michail Tsagris <mtsagris@uoc.gr>

Depends R (>= 4.0)

Imports bigstatsr, cluster, doParallel, emplik, energy, foreach, glmnet, graphics, grDevices, quantreg, MASS, Matrix, mda, minpack.lm, mixture, nnet, quadprog, Rfast, Rfast2, Rnanoflann, sn, stats

Suggests bigparallelr, codalm, FlexDir

Description Regression, classification, contour plots, hypothesis testing and fitting of distributions for compositional data are some of the functions included. We further include functions for percentages (or proportions).
The standard textbook for such data is John Aitchison's (1986) ``The statistical analysis of compositional data''. Relevant papers include:
a) Tsagris M.T., Preston S. and Wood A.T.A. (2011). ``A data-based power transformation for compositional data''. Fourth International International Workshop on Compositional Data Analysis. <doi:10.48550/arXiv.1106.1451>
b) Tsagris M. (2014). ``The k-NN algorithm for compositional data: a revised approach with and without zero values present''. Journal of Data Science, 12(3): 519--534. <doi:10.6339/JDS.201407_12(3).0008>.
c) Tsagris M. (2015). ``A novel, divergence based, regression for compositional data''. Proceedings of the 28th Panhellenic Statistics Conference, 15-18 April 2015, Athens, Greece, 430--444. <doi:10.48550/arXiv.1511.07600>.
d) Tsagris M. (2015). ``Regression analysis with compositional data containing zero values''. Chilean Journal of Statistics, 6(2): 47--57. <https://soche.cl/chjs/volumes/06/02/Tsagris(2015).pdf>.

e) Tsagris M., Preston S. and Wood A.T.A. (2016). ``Improved supervised classification for compositional data using the alpha-transformation". Journal of Classification, 33(2): 243--261. <doi:10.1007/s00357-016-9207-5>.

f) Tsagris M., Preston S. and Wood A.T.A. (2017). ``Nonparametric hypothesis testing for equality of means on the simplex". Journal of Statistical Computation and Simulation, 87(2): 406--422. <doi:10.1080/00949655.2016.1216554>.

g) Tsagris M. and Stewart C. (2018). ``A Dirichlet regression model for compositional data with zeros". Lobachevskii Journal of Mathematics, 39(3): 398--412. <doi:10.1134/S1995080218030198>.

h) Alenazi A. (2019). ``Regression for compositional data with compositional data as predictor variables with or without zero values". Journal of Data Science, 17(1): 219--238. <doi:10.6339/JDS.201901_17(1).0010>.

i) Tsagris M. and Stewart C. (2020). ``A folded model for compositional data analysis". Australian and New Zealand Journal of Statistics, 62(2): 249--277. <doi:10.1111/anzs.12289>.

j) Alenazi A.A. (2022). ``f-divergence regression models for compositional data". Pakistan Journal of Statistics and Operation Research, 18(4): 867--882. <doi:10.18187/pjsor.v18i4.3969>.

k) Tsagris M. and Stewart C. (2022). ``A Review of Flexible Transformations for Modeling Compositional Data". In Advances and Innovations in Statistics and Data Science, pp. 225--234. <doi:10.1007/978-3-031-08329-7_10>.

l) Alenazi A. (2023). ``A review of compositional data analysis and recent advances". Communications in Statistics--Theory and Methods, 52(16): 5535--5567. <doi:10.1080/03610926.2021.2014890>.

m) Tsagris M., Alenazi A. and Stewart C. (2023). ``Flexible non-parametric regression models for compositional response data with zeros". Statistics and Computing, 33(106). <doi:10.1007/s11222-023-10277-5>.

n) Tsagris. M. (2025). ``Constrained least squares simplicial-simplicial regression". Statistics and Computing, 35(27). <doi:10.1007/s11222-024-10560-z>.

o) Sevinc V. and Tsagris. M. (2024). ``Energy Based Equality of Distributions Testing for Compositional Data". <doi:10.48550/arXiv.2412.05199>.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2025-06-12 22:20:08 UTC

Contents

Compositional-package *Compositional Data Analysis*

Description

A Collection of Functions for Compositional Data Analysis.

Details

Package:	Compositional
Type:	Package
Version:	7.5
Date:	2025-06-13
License:	GPL-2

Maintainers

Michail Tsagris <mtsagris@uoc.gr>

Note

Acknowledgments:

Michail Tsagris would like to express his acknowledgments to Professor Andy Wood and Professor Simon Preston from the university of Nottingham for being his supervisors during his PhD in compositional data analysis.

We would also like to express our acknowledgments to Profesor Kurt Hornik (and also the rest of the R core team) for his help with this package.

Manos Papadakis, undergraduate student in the department of computer science, university of Crete, is also acknowledged for his programming tips.

Ermanno Affuso from the university of South Alabama suggested that I have a default value in the function `mkde`.

Van Thang Hoang from Hasselt university spotted a bug in the function `js.compreg`.

Claudia Wehrhahn Cortes spotted a bug in the function `diri.reg`.

Philipp Kynast from Bruker Daltonik GmbH found a mistake in the function `mkde` which is now fixed.

Jasmine Heyse from the university of Ghent spotted a bug in the function `kl.compreg` which is now fixed.

Magne Neby suggested to add names in the covariance matrix of the divergence based regression models.

John Barry from the Centre for Environment, Fisheries, and Aquaculture Science (UK) suggested that I should add more explanation in the function `diri.est`. I hope it is clearer now.

Charlotte Fabri and Laura Byrne spotted a possible problem in the function `zadr`.

Levi Bankston found a bug in the bootstrap version of the function `kl.compreg`.

Sucharitha Dodamgodage suggested to add an extra case in the function `dirimean.test`.

Loic Mangnier found a bug in the function `lc.glm` which is now fixed and also became faster.

Ravi Varadhan found a bug in `diri.reg` and he is acknowledged for that.

Author(s)

Michail Tsagris <mtsagris@uoc.gr>, Giorgos Athineou <gioathineou@gmail.com>, Abdulaziz Alenazi <a.alenazi@nbu.edu.sa> and Christos Adam <pada4m4@gmail.com>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

`Aitchison's test for two mean vectors and/or covariance matrices`

Aitchison's test for two mean vectors and/or covariance matrices

---

Description

Aitchison's test for two mean vectors and/or covariance matrices.

Usage

```
ait.test(x1, x2, type = 1, alpha = 0.05)
```

Arguments

x1	A matrix containing the compositional data of the first sample. Zeros are not allowed.
x2	A matrix containing the compositional data of the second sample. Zeros are not allowed.
type	The type of hypothesis test to perform. Type=1 refers to testing the equality of the mean vectors and the covariance matrices. Type=2 refers to testing the equality of the covariance matrices. Type=2 refers to testing the equality of the mean vectors.
alpha	The significance level, set to 0.05 by default.

Details

The test is described in Aitchison (2003). See the references for more information.

Value

A vector with the test statistic, the p-value, the critical value and the degrees of freedom of the chi-square distribution.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

John Aitchison (2003). The Statistical Analysis of Compositional Data, p. 153-157. Blackburn Press.

See Also

comp.test

Examples

```
x1 <- as.matrix(iris[1:50, 1:4])
x1 <- x1 / rowSums(x1)
x2 <- as.matrix(iris[51:100, 1:4])
x2 <- x2 / rowSums(x2)
ait.test(x1, x2, type = 1)
ait.test(x1, x2, type = 2)
ait.test(x1, x2, type = 3)
```

All pairwise additive log-ratio transformations

All pairwise additive log-ratio transformations

Description

All pairwise additive log-ratio transformations.

Usage

```
alr.all(x)
```

Arguments

x	A numerical matrix with the compositional data.

Details

The additive log-ratio transformation with the first component being the commn divisor is applied. Then all the other pairwise log-ratios are computed and added next to each column. For example, divide by the first component, then divide by the second component and so on. This means that no zeros are allowed.

Value

A matrix with all pairwise alr transformed data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

alr, alfa

Examples

```
x <- as.matrix(iris[, 2:4])
x <- x / rowSums(x)
y <- alr.all(x)
```

Alpha-generalised correlations between two compositional datasets

α-generalised correlations between two compositional datasets

---

Description

α-generalised correlations between two compositional datasets.

Usage

```
acor(y, x, a, type = "dcor")
```

Arguments

`y` A matrix with the compositional data.

`x` A matrix with the compositional data.

`a` The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied. If more than one valuesare supplied the distance or canonical correlation are computed for all values.

`type` The type of correlation to compute, the distance correlation ("dcor"), the canonical correlation ("cancor") or "both".

Details

The α-transformation is applied to each composition and then the distance correlation or the canonical correlation is computed. If one value of α is supplied the type="cancor" will return all eigenvalues. If more than one values of α are provided then the first eigenvalue only will be returned.

Value

A vector or a matrix depending on the length of the values of α and the type of the correlation to be computed.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

G.J. Szekely, M.L. Rizzo and N. K. Bakirov (2007). Measuring and Testing Independence by Correlation of Distances. Annals of Statistics, 35(6): 2769-2794.

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

Tsagris M. and Papadakis M. (2025). Fast and light-weight energy statistics using the R package Rfast. https://arxiv.org/abs/2501.02849v2

See Also

`acor.tune`, `aeqdist.etest`, `alfa`, `alfa.profile`

Examples

```
y <- rdiri(30, runif(3) )
x <- rdiri(30, runif(4) )
acor(y, x, a = 0.4)
```

ANOVA for the log-contrast GLM versus the uncostrained GLM
ANOVA for the log-contrast GLM versus the uncostrained GLM

Description

ANOVA for the log-contrast GLM versus the uncostrained GLM.

Usage

```
lcglm.aov(mod0, mod1)
```

Arguments

mod0	The log-contrast GLM. The object returned by lc.glm.
mod1	The unconstrained GLM. The object returned by ulc.glm.

Details

A chi-square test is performed to test the zero-to-sum constraints of the regression coefficients.

Value

A vector with two values, the chi-square test statistic and its associated p-value.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

lc.glm, ulc.glm

Examples

```
y <- rbinom(150, 1, 0.5)
x <- as.matrix(iris[, 2:4])
x <- x / rowSums(x)
mod0 <- lc.glm(y, x)
mod1 <- ulc.glm(y, x)
lcglm.aov(mod0, mod1)
```

ANOVA for the log-contrast regression versus the uncostrained linear regression

ANOVA for the log-contrast regression versus the uncostrained linear regression

Description

ANOVA for the log-contrast regression versus the uncostrained linear regression.

Usage

```
lcreg.aov(mod0, mod1)
```

Arguments

mod0	The log-contrast regression model. The object returned by lc.reg.
mod1	The unconstrained linear regression model. The object returned by ulc.reg.

Details

An F-test is performed to test the zero-to-sum constraints of the regression coefficients.

Value

A vector with two values, the F test statistic and its associated p-value.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

lc.reg, ulc.reg, alfa.pcr, alfa.knn.reg

Examples

```
y <- iris[, 1]
x <- as.matrix(iris[, 2:4])
x <- x / rowSums(x)
mod0 <- lc.reg(y, x)
mod1 <- ulc.reg(y, x)
lcreg.aov(mod0, mod1)
```

`Beta regression` *Beta regression*

Description

Beta regression.

Usage

```
beta.reg(y, x, xnew = NULL)
```

Arguments

`y`	The response variable. It must be a numerical vector with proportions excluding 0 and 1.
`x`	The indendent variable(s). It can be a vector, a matrix or a dataframe with continuous only variables, a data frame with mixed or only categorical variables.
`xnew`	If you have new values for the predictor variables (dataset) whose response values you want to predict insert them here.

Details

Beta regression is fitted.

Value

A list including:

`phi`	The estimated precision parameter.
`info`	A matrix with the estimated regression parameters, their standard errors, Wald statistics and associated p-values.
`loglik`	The log-likelihood of the regression model.
`est`	The estimated values if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Ferrari S.L.P. and Cribari-Neto F. (2004). Beta Regression for Modelling Rates and Proportions. Journal of Applied Statistics, 31(7): 799-815.

See Also

`beta.est`, `propreg`, `diri.reg`

Examples

```
y <- rbeta(300, 3, 5)
x <- matrix( rnorm(300 * 2), ncol = 2)
beta.reg(y, x)
```

`Column-wise MLE of some univariate distributions`
Column-wise MLE of some univariate distributions

Description

Column-wise MLE of some univariate distributions.

Usage

```
colbeta.est(x, tol = 1e-07, maxiters = 100, parallel = FALSE)
collogitnorm.est(x)
colunitweibull.est(x, tol = 1e-07, maxiters = 100, parallel = FALSE)
colzilogitnorm.est(x)
```

Arguments

`x`	A numerical matrix with data. Each column refers to a different vector of observations of the same distribution. The values must by percentages, exluding 0 and 1,
`tol`	The tolerance value to terminate the Newton-Fisher algorithm.
`maxiters`	The maximum number of iterations to implement.
`parallel`	Do you want to calculations to take place in parallel? The default value is FALSE

Details

For each column, the same distribution is fitted and its parameters and log-likelihood are computed.

Value

A matrix with two, three or four columns. The first one, two or three columns contain the parameter(s) of the distribution, while the last column contains the relevant log-likelihood.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

N.L. Johnson, S. Kotz & N. Balakrishnan (1994). Continuous Univariate Distributions, Volume 1 (2nd Edition).

N.L. Johnson, S. Kotz & N. Balakrishnan (1970). Distributions in statistics: continuous univariate distributions, Volume 2.

J. Mazucheli, A. F. B. Menezes, L. B. Fernandes, R. P. de Oliveira & M. E. Ghitany (2020). The unit-Weibull distribution as an alternative to the Kumaraswamy distribution for the modeling of quantiles conditional on covariates. Journal of Applied Statistics, DOI:10.1080/02664763.2019.1657813.

See Also

`beta.est`

Examples

```
x <- matrix( rbeta(200, 3, 4), ncol = 4 )
a <- colbeta.est(x)
```

`Contour plot of mixtures of Dirichlet distributions in S^2`

Contour plot of mixtures of Dirichlet distributions in $S\hat{}2$

Description

Contour plot of mixtures of Dirichlet distributions in S^2.

Usage

```
mixdiri.contour(a, prob, n = 100, x = NULL, cont.line = FALSE)
```

Arguments

`a`	A matrix where each row contains the parameters of each Dirichlet disctribution.
`prob`	A vector with the mixing probabilities.
`n`	The number of grid points to consider over which the density is calculated.
`x`	This is either NULL (no data) or contains a 3 column matrix with compositional data.
`cont.line`	Do you want the contour lines to appear? If yes, set this TRUE.

Details

The user can plot only the contour lines of a Dirichlet with a given vector of parameters, or can also add the relevant data should he/she wish to.

Value

A ternary diagram with the points and the Dirichlet contour lines.

Author(s)

Michail Tsagris and Christos Adam.

R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>` and Christos Adam `<pada4m4@gmail.com>`.

References

Ng Kai Wang, Guo-Liang Tian and Man-Lai Tang (2011). Dirichlet and related distributions: Theory, methods and applications. John Wiley & Sons.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`diri.contour`, `gendiri.contour`, `compnorm.contour`,`comp.kerncontour`, `mix.compnorm.contour`,`diri.nr`, `dda`

Examples

```
a <- matrix( c(12, 30, 45, 32, 50, 16), byrow = TRUE,ncol = 3)
prob <- c(0.5, 0.5)
mixdiri.contour(a, prob)
```

`Contour plot of the alpha multivariate normal in S^2`

Contour plot of the α multivariate normal in $S\hat{}2$

Description

Contour plot of the α multivariate normal in S^2.

Usage

```
alfa.contour(m, s, a, n = 100, x = NULL, cont.line = FALSE)
```

Arguments

`m`	The mean vector of the α multivariate normal model.
`s`	The covariance matrix of the α multivariate normal model.
`a`	The value of a for the α-transformation.
`n`	The number of grid points to consider over which the density is calculated.
`x`	This is either NULL (no data) or contains a 3 column matrix with compositional data.
`cont.line`	Do you want the contour lines to appear? If yes, set this TRUE.

Details

The α-transformation is applied to the compositional data and then for a grid of points within the 2-dimensional simplex, the density of the α multivariate normal is calculated and the contours are plotted.

Value

The contour plot of the α multivariate normal appears.

Author(s)

Michail Tsagris and Christos Adam.

R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>` and Christos Adam `<pada4m4@gmail.com>`.

References

Tsagris M. and Stewart C. (2022). A Review of Flexible Transformations for Modeling Compositional Data. In Advances and Innovations in Statistics and Data Science, pp. 225–234. https://link.springer.com/chapter/10.10
3-031-08329-7_10

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

`folded.contour`, `compnorm.contour`, `diri.contour`, `mix.compnorm.contour`, `bivt.contour`, `skewnorm.contour`

Examples

```
x <- as.matrix(iris[, 1:3])
x <- x / rowSums(x)
a <- a.est(x)$best
m <- colMeans(alfa(x, a)$aff)
s <- cov(alfa(x, a)$aff)
alfa.contour(m, s, a)
```

`Contour plot of the alpha-folded model in S^2`

Contour plot of the α-folded model in $S\hat{}2$

Description

Contour plot of the α-folded model in S^2.

Usage

```
folded.contour(mu, su, p, a, n = 100, x = NULL, cont.line = FALSE)
```

Arguments

`mu` The mean vector of the folded model.

`su` The covariance matrix of the folded model.

`p` The probability inside the simplex of the folded model.

`a` The value of a for the α-transformation.

`n` The number of grid points to consider over which the density is calculated.

`x` This is either NULL (no data) or contains a 3 column matrix with compositional data.

`cont.line` Do you want the contour lines to appear? If yes, set this TRUE.

Details

The α-transformation is applied to the compositional data and then for a grid of points within the 2-dimensional simplex the folded model's density is calculated and the contours are plotted.

Value

The contour plot of the folded model appears.

Author(s)

Michail Tsagris and Christos Adam.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Christos Adam <pada4m4@gmail.com>.

References

Tsagris M. and Stewart C. (2022). A Review of Flexible Transformations for Modeling Compositional Data. In Advances and Innovations in Statistics and Data Science, pp. 225–234. https://link.springer.com/chapter/10.10
3-031-08329-7_10

Tsagris M. and Stewart C. (2020). A folded model for compositional data analysis. Australian and New Zealand Journal of Statistics, 62(2): 249-277. https://arxiv.org/pdf/1802.07330.pdf

See Also

alfa.contour, compnorm.contour, diri.contour, mix.compnorm.contour,bivt.contour, skewnorm.contour

Examples

```
x <- as.matrix(iris[, 1:3])
x <- x / rowSums(x)
a <- a.est(x)$best
mod <- alpha.mle(x, a)
folded.contour(mod$mu, mod$su, mod$p, a)
```

`Contour plot of the Dirichlet distribution in S^2`

Contour plot of the Dirichlet distribution in $S\hat{}2$

Description

Contour plot of the Dirichlet distribution in S^2.

Usage

```
diri.contour(a, n = 100, x = NULL, cont.line = FALSE)
```

Arguments

`a`	A vector with three elements corresponding to the 3 (estimated) parameters.
`n`	The number of grid points to consider over which the density is calculated.
`x`	This is either NULL (no data) or contains a 3 column matrix with compositional data.
`cont.line`	Do you want the contour lines to appear? If yes, set this TRUE.

Details

The user can plot only the contour lines of a Dirichlet with a given vector of parameters, or can also add the relevant data should he/she wish to.

Value

A ternary diagram with the points and the Dirichlet contour lines.

Author(s)

Michail Tsagris and Christos Adam.

R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>` and Christos Adam `<pada4m4@gmail.com>`.

References

Ng Kai Wang, Guo-Liang Tian and Man-Lai Tang (2011). Dirichlet and related distributions: Theory, methods and applications. John Wiley & Sons.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`mixdiri.contour`, `gendiri.contour`, `compnorm.contour`,`comp.kerncontour`, `mix.compnorm.contour`

Examples

```
x <- as.matrix( iris[, 1:3] )
x <- x / rowSums(x)
diri.contour( a = c(3, 4, 2) )
```

`Contour plot of the Flexible Dirichlet distribution in S^2`

Contour plot of the Flexible Dirichlet distribution in $S\hat{}2$

Description

Contour plot of the Flexible Dirichlet distribution in S^2.

Usage

```
fd.contour(alpha, prob, tau, n = 100, x = NULL, cont.line = FALSE)
```

Arguments

`alpha`	A vector of the non-negative α parameters.
`prob`	A vector of the clusters' probabilities. It must sum to one.
`tau`	The non-negative scalar tau parameter.
`n`	The number of grid points to consider over which the density is calculated.
`x`	This is either NULL (no data) or contains a 3 column matrix with compositional data.
`cont.line`	Do you want the contour lines to appear? If yes, set this TRUE.

Details

The user can plot only the contour lines of a Dirichlet with a given vector of parameters, or can also add the relevant data should they wish to.

Value

A ternary diagram with the points and the Flexible Dirichlet contour lines.

Author(s)

Michail Tsagris and Christos Adam.

R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>` and Christos Adam `<pada4m4@gmail.com>`.

References

Ongaro A. and Migliorati S. (2013). A generalization of the Dirichlet distribution. Journal of Multivariate Analysis, 114, 412–426.

Migliorati S., Ongaro A. and Monti G. S. (2017). A structured Dirichlet mixture model for compositional data: inferential and applicative issues. Statistics and Computing, 27, 963–983.

See Also

`compnorm.contour`, `folded.contour`, `bivt.contour`,`comp.kerncontour`, `mix.compnorm.contour`

Examples

```
fd.contour(alpha = c(10, 11, 12), prob = c(0.25, 0.25, 0.5), tau = 4)
```

`Contour plot of the Gaussian mixture model in S^2`

Contour plot of the Gaussian mixture model in $S\hat{}2$

Description

Contour plot of the Gaussian mixture model in S^2.

Usage

```
mix.compnorm.contour(mod, type = "alr", n = 100, x = NULL, cont.line = FALSE)
```

Arguments

`mod`	An object containing the output of a `mix.compnorm` model.
`type`	The type of trasformation used, either the additive log-ratio ("alr"), the isometric log-ratio ("ilr") or the pivot coordinate ("pivot") transformation.
`n`	The number of grid points to consider over which the density is calculated.
`x`	A matrix with the compositional data.
`cont.line`	Do you want the contour lines to appear? If yes, set this TRUE.

Details

The contour plot of a Gaussian mixture model is plotted. For this you need the (fitted) model.

Value

A ternary plot with the data and the contour lines of the fitted Gaussian mixture model.

Author(s)

Michail Tsagris and Christos Adam.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Christos Adam <pada4m4@gmail.com>.

References

Ryan P. Browne, Aisha ElSherbiny and Paul D. McNicholas (2015). R package mixture: Mixture Models for Clustering and Classification

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

mix.compnorm, bic.mixcompnorm, diri.contour

Examples

```
x <- as.matrix(iris[, 1:3])
x <- x / rowSums(x)
mod <- mix.compnorm(x, 3, model = "EII")
mix.compnorm.contour(mod, "alr")
```

Contour plot of the generalised Dirichlet distribution in S^2
Contour plot of the generalised Dirichlet distribution in $S\hat{}2$

Description

Contour plot of the generalised Dirichlet distribution in S^2.

Usage

```
gendiri.contour(a, b, n = 100, x = NULL, cont.line = FALSE)
```

Arguments

a	A vector with three elements corresponding to the 3 (estimated) shape parameter values.
b	A vector with three elements corresponding to the 3 (estimated) scale parameter values.
n	The number of grid points to consider over which the density is calculated.
x	This is either NULL (no data) or contains a 3 column matrix with compositional data.
cont.line	Do you want the contour lines to appear? If yes, set this TRUE.

Details

The user can plot only the contour lines of a Dirichlet with a given vector of parameters, or can also add the relevant data should he/she wish to.

Value

A ternary diagram with the points and the Dirichlet contour lines.

Author(s)

Michail Tsagris and Christos Adam.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Christos Adam <pada4m4@gmail.com>.

References

Ng Kai Wang, Guo-Liang Tian and Man-Lai Tang (2011). Dirichlet and related distributions: Theory, methods and applications. John Wiley & Sons.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

diri.contour, mixdiri.contour, compnorm.contour,comp.kerncontour, mix.compnorm.contour

Examples

```
x <- as.matrix( iris[, 1:3] )
x <- x / rowSums(x)
gendiri.contour( a = c(3, 4, 2), b = c(1, 2, 3) )
```

Contour plot of the kernel density estimate in S^2

Contour plot of the kernel density estimate in $S\hat{}2$

Description

Contour plot of the kernel density estimate in S^2.

Usage

```
comp.kerncontour(x, type = "alr", n = 50, cont.line = FALSE)
```

Arguments

`x` A matrix with the compositional data. It has to be a 3 column matrix.

`type` This is either "alr" or "ilr", corresponding to the additive and the isometric log-ratio transformation respectively.

`n` The number of grid points to consider, over which the density is calculated.

`cont.line` Do you want the contour lines to appear? If yes, set this TRUE.

Details

The alr or the ilr transformation are applied to the compositional data. Then, the optimal bandwidth using maximum likelihood cross-validation is chosen. The multivariate normal kernel density is calculated for a grid of points. Those points are the points on the 2-dimensional simplex. Finally the contours are plotted.

Value

A ternary diagram with the points and the kernel contour lines.

Author(s)

Michail Tsagris and Christos Adam.

R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>` and Christos Adam `<pada4m4@gmail.com>`.

References

M.P. Wand and M.C. Jones (1995). Kernel smoothing, CrC Press.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`diri.contour`, `mix.compnorm.contour`, `bivt.contour`, `compnorm.contour`

Examples

```
x <- as.matrix(iris[, 1:3])
x <- x / rowSums(x)
comp.kerncontour(x, type = "alr", n = 20)
comp.kerncontour(x, type = "ilr", n = 20)
```

`Contour plot of the normal distribution in S^2`

Contour plot of the normal distribution in $S\hat{}2$

Description

Contour plot of the normal distribution in S^2.

Usage

```
compnorm.contour(m, s, type = "alr", n = 100, x = NULL, cont.line = FALSE)
```

Arguments

`m` The mean vector.

`s` The covariance matrix.

`type` The type of trasformation used, either the additive log-ratio ("alr"), the isometric log-ratio ("ilr") or the pivot coordinate ("pivot") transformation.

`n` The number of grid points to consider over which the density is calculated.

`x` This is either NULL (no data) or contains a 3 column matrix with compositional data.

`cont.line` Do you want the contour lines to appear? If yes, set this TRUE.

Details

The alr or the ilr transformation is applied to the compositional data at first. Then for a grid of points within the 2-dimensional simplex the bivariate normal density is calculated and the contours are plotted along with the points.

Value

A ternary diagram with the points (if appear = TRUE) and the bivariate normal contour lines.

Author(s)

Michail Tsagris and Christos Adam.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Christos Adam <`pada4m4@gmail.com`>.

See Also

`diri.contour`, `mix.compnorm.contour`, `bivt.contour`, `skewnorm.contour`

Examples

```
x <- as.matrix(iris[, 1:3])
x <- x / rowSums(x)
y <- Compositional::alr(x)
m <- colMeans(y)
s <- cov(y)
compnorm.contour(m, s)
```

Contour plot of the skew skew-normal distribution in S^2

Contour plot of the skew skew-normal distribution in $S\hat{}2$

Description

Contour plot of the skew skew-normal distribution in S^2.

Usage

```
skewnorm.contour(x, type = "alr", n = 100, appear = TRUE, cont.line = FALSE)
```

Arguments

`x`	A matrix with the compositional data. It has to be a 3 column matrix.
`type`	This is either "alr" or "ilr", corresponding to the additive and the isometric log-ratio transformation respectively.
`n`	The number of grid points to consider over which the density is calculated.
`appear`	Should the available data appear on the ternary plot (TRUE) or not (FALSE)?
`cont.line`	Do you want the contour lines to appear? If yes, set this TRUE.

Details

The alr or the ilr transformation is applied to the compositional data at first. Then for a grid of points within the 2-dimensional simplex the bivariate skew skew-normal density is calculated and the contours are plotted along with the points.

Value

A ternary diagram with the points (if appear = TRUE) and the bivariate skew skew-normal contour lines.

Author(s)

Michail Tsagris and Christos Adam.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Christos Adam <pada4m4@gmail.com>.

References

Azzalini A. and Valle A. D. (1996). The multivariate skew-skewnormal distribution. Biometrika 83(4): 715–726.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`diri.contour`, `mix.compnorm.contour`, `bivt.contour`, `compnorm.contour`

Examples

```
x <- as.matrix(iris[51:100, 1:3])
x <- x / rowSums(x)
skewnorm.contour(x)
```

`Contour plot of the t distribution in S^2`

Contour plot of the t distribution in $S\hat{}2$

Description

Contour plot of the t distribution in S^2.

Usage

```
bivt.contour(x, type = "alr", n = 100, appear = TRUE, cont.line = FALSE)
```

Arguments

`x`	A matrix with compositional data. It has to be a 3 column matrix.
`type`	This is either "alr" or "ilr", corresponding to the additive and the isometric log-ratio transformation respectively.
`n`	The number of grid points to consider over which the density is calculated.
`appear`	Should the available data appear on the ternary plot (TRUE) or not (FALSE)?
`cont.line`	Do you want the contour lines to appear? If yes, set this TRUE.

Details

The alr or the ilr transformation is applied to the compositional data at first and the location, scatter and degrees of freedom of the bivariate t distribution are computed. Then for a grid of points within the 2-dimensional simplex the bivariate t density is calculated and the contours are plotted along with the points.

Value

A ternary diagram with the points (if appear = TRUE) and the bivariate t contour lines.

Author(s)

Michail Tsagris and Christos Adam.

R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>` and Christos Adam `<pada4m4@gmail.com>`.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`diri.contour`, `mix.compnorm.contour`, `compnorm.contour`, `skewnorm.contour`

Examples

```
x <- as.matrix( iris[, 1:3] )
x <- x / rowSums(x)
bivt.contour(x)
bivt.contour(x, type = "ilr")
```

Cross validation for some compositional regression models

Cross validation for some compositional regression models

Description

Cross validation for some compositional regression models.

Usage

```
cv.comp.reg(y, x, type = "comp.reg", nfolds = 10, folds = NULL, seed = NULL)
```

Arguments

`y`	A matrix with compositional data. Zero values are allowed for some regression models.
`x`	The predictor variable(s).
`type`	This can be one of the following: "comp.reg", "robust", "kl.compreg", "js.compreg", "diri.reg" or "zadr".
`nfolds`	The number of folds to be used. This is taken into consideration only if the folds argument is not supplied.
`folds`	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
`seed`	If seed is TRUE the results will always be the same.

Details

A k-fold cross validation for a compositional regression model is performed.

Value

A list including:

`runtime`	The runtime of the cross-validation procedure.
`kl`	The Kullback-Leibler divergences for all runs.
`js`	The Jensen-Shannon divergences for all runs.
`perf`	The average Kullback-Leibler divergence and average Jensen-Shannon divergence.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

See Also

`comp.reg`, `kl.compreg`, `compppr.tune`, `aknnreg.tune`

Examples

```
y <- as.matrix( iris[, 1:3] )
y <- y / rowSums(y)
x <- iris[, 4]
mod <- cv.comp.reg(y, x)
```

Cross validation for the alpha-k-NN regression with compositional predictor variables

Cross validation for the α-k-NN regression with compositional predictor variables

---

Description

Cross validation for the α-k-NN regression with compositional predictor variables.

Usage

```
alfaknnreg.tune(y, x, a = seq(-1, 1, by = 0.1), k = 2:10, nfolds = 10,
apostasi = "euclidean", method = "average", folds = NULL, seed = NULL, graph = FALSE)
```

Arguments

`y`	The response variable, a numerical vector.
`x`	A matrix with the available compositional data. Zeros are allowed.
`a`	A vector with a grid of values of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
`k`	The number of nearest neighbours to consider. It can be a single number or a vector.
`nfolds`	The number of folds. Set to 10 by default.
`apostasi`	The type of distance to use, either "euclidean" or "manhattan".
`method`	If you want to take the average of the reponses of the k closest observations, type "average". For the median, type "median" and for the harmonic mean, type "harmonic".
`folds`	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
`seed`	If seed is TRUE the results will always be the same.
`graph`	If graph is TRUE (default value) a filled contour plot will appear.

Details

A k-fold cross validation for the α-k-NN regression for compositional response data is performed.

Value

A list including:

`mspe`	The mean square error of prediction.
`performance`	The minimum mean square error of prediction.
`opt_a`	The optimal value of α.
`opt_k`	The optimal value of k.
`runtime`	The runtime of the cross-validation procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris M., Alenazi A. and Stewart C. (2023). Flexible non-parametric regression models for compositional response data with zeros. Statistics and Computing, 33(106).

https://link.springer.com/article/10.1007/s11222-023-10277-5

See Also

`alfa.rda`, `alfa.fda`

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y <- fgl[, 1]
mod <- alfaknnreg.tune(y, x, a = seq(0.2, 0.4, by = 0.1), k = 2:4, nfolds = 5)
```

`Cross validation for the alpha-k-NN regression with compositional response data`

Cross validation for the α-k-NN regression with compositional response data

---

Description

Cross validation for the α-k-NN regression with compositional response data.

Usage

```
aknnreg.tune(y, x, a = seq(0.1, 1, by = 0.1), k = 2:10, apostasi = "euclidean",
nfolds = 10, folds = NULL, seed = NULL, rann = FALSE)
```

Arguments

y	A matrix with the compositional response data. Zeros are allowed.
x	A matrix with the available predictor variables.
a	A vector with a grid of values of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
k	The number of nearest neighbours to consider. It can be a single number or a vector.
apostasi	The type of distance to use, either "euclidean" or "manhattan".
nfolds	The number of folds. Set to 10 by default.
folds	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
seed	You can specify your own seed number here or leave it NULL.
rann	If you have large scale datasets and want a faster k-NN search, you can use kd-trees implemented in the R package "Rnanoflann". In this case you must set this argument equal to TRUE. Note however, that in this case, the only available distance is by default "euclidean".

Details

A k-fold cross validation for the α-k-NN regression for compositional response data is performed.

Value

A list including:

`kl` The Kullback-Leibler divergence for all combinations of α and k.

`js` The Jensen-Shannon divergence for all combinations of α and k.

`klmin` The minimum Kullback-Leibler divergence.

`jsmin` The minimum Jensen-Shannon divergence.

`kl.alpha` The optimal α that leads to the minimum Kullback-Leibler divergence.

`kl.k` The optimal k that leads to the minimum Kullback-Leibler divergence.

`js.alpha` The optimal α that leads to the minimum Jensen-Shannon divergence.

`js.k` The optimal k that leads to the minimum Jensen-Shannon divergence.

`runtime` The runtime of the cross-validation procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris M., Alenazi A. and Stewart C. (2023). Flexible non-parametric regression models for compositional response data with zeros. Statistics and Computing, 33(106).

https://link.springer.com/article/10.1007/s11222-023-10277-5

See Also

`aknn.reg`, `akernreg.tune`, `akern.reg`, `alfa.rda`, `alfa.fda`

Examples

```
y <- as.matrix( iris[, 1:3] )
y <- y / rowSums(y)
x <- iris[, 4]
mod <- aknnreg.tune(y, x, a = c(0.4, 0.6), k = 2:4, nfolds = 5)
```

`Cross validation for the alpha-kernel regression with compositional response data`

Cross validation for the α-kernel regression with compositional response data

---

Description

Cross validation for the α-kernel regression with compositional response data.

Usage

```
akernreg.tune(y, x, a = seq(0.1, 1, by = 0.1), h = seq(0.1, 1, length = 10),
type = "gauss", nfolds = 10, folds = NULL, seed = NULL)
```

Arguments

`y`	A matrix with the compositional response data. Zeros are allowed.
`x`	A matrix with the available predictor variables.
`a`	A vector with a grid of values of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
`h`	A vector with the bandwidth value(s) to consider.
`type`	The type of kernel to use, "gauss" or "laplace".
`nfolds`	The number of folds. Set to 10 by default.
`folds`	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
`seed`	You can specify your own seed number here or leave it NULL.

Details

A k-fold cross validation for the α-kernel regression for compositional response data is performed.

Value

A list including:

`kl`	The Kullback-Leibler divergence for all combinations of α and h.
`js`	The Jensen-Shannon divergence for all combinations of α and h.
`klmin`	The minimum Kullback-Leibler divergence.
`jsmin`	The minimum Jensen-Shannon divergence.
`kl.alpha`	The optimal α that leads to the minimum Kullback-Leibler divergence.
`kl.h`	The optimal h that leads to the minimum Kullback-Leibler divergence.
`js.alpha`	The optimal α that leads to the minimum Jensen-Shannon divergence.
`js.h`	The optimal h that leads to the minimum Jensen-Shannon divergence.
`runtime`	The runtime of the cross-validation procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris M., Alenazi A. and Stewart C. (2023). Flexible non-parametric regression models for compositional response data with zeros. Statistics and Computing, 33(106).

https://link.springer.com/article/10.1007/s11222-023-10277-5

See Also

`akern.reg`, `aknnreg.tune`, `aknn.reg`, `alfa.rda`, `alfa.fda`

Examples

```
y <- as.matrix( iris[, 1:3] )
y <- y / rowSums(y)
x <- iris[, 4]
mod <- akernreg.tune(y, x, a = c(0.4, 0.6), h = c(0.1, 0.2), nfolds = 5)
```

`Cross validation for the kernel regression with Euclidean response data`

Cross validation for the kernel regression with Euclidean response data

---

Description

Cross validation for the kernel regression with Euclidean response data.

Usage

```
kernreg.tune(y, x, h = seq(0.1, 1, length = 10), type = "gauss",
nfolds = 10, folds = NULL, seed = NULL, graph = FALSE, ncores = 1)
```

Arguments

`y`	A matrix or a vector with the Euclidean response.
`x`	A matrix with the available predictor variables.
`h`	A vector with the bandwidth value(s) h to consider.
`type`	The type of kernel to use, "gauss" or "laplace".
`nfolds`	The number of folds. Set to 10 by default.
`folds`	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
`seed`	You can specify your own seed number here or leave it NULL.
`graph`	If graph is TRUE (default value) a plot will appear.
`ncores`	The number of cores to use. Default value is 1.

Details

A k-fold cross validation for the kernel regression with a euclidean response is performed.

Value

A list including:

mspe	The mean squared prediction error (MSPE) for each fold and value of h.
h	The optimal h that leads to the minimum MSPE.
performance	The minimum MSPE.
runtime	The runtime of the cross-validation procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Wand M. P. and Jones M. C. (1994). Kernel smoothing. CRC press.

See Also

kern.reg, aknnreg.tune, aknn.reg

Examples

```
y <- iris[, 1]
x <- iris[, 2:4]
mod <- kernreg.tune(y, x, h = c(0.1, 0.2, 0.3) )
```

Cross validation for the regularised and flexible discriminant analysis with compositional data using the alpha-transformation

Cross validation for the regularised and flexible discriminant analysis with compositional data using the α-transformation

---

Description

Cross validation for the regularised and flexible discriminant analysis with compositional data using the α-transformation.

Usage

```
alfarda.tune(x, ina, a = seq(-1, 1, by = 0.1), nfolds = 10,
gam = seq(0, 1, by = 0.1), del = seq(0, 1, by = 0.1),
ncores = 1, folds = NULL, stratified = TRUE, seed = NULL)

alfafda.tune(x, ina, a = seq(-1, 1, by = 0.1), nfolds = 10,
folds = NULL, stratified = TRUE, seed = NULL, graph = FALSE)
```

Arguments

`x` A matrix with the available compositional data. Zeros are allowed.

`ina` A group indicator variable for the avaiable data.

`a` A vector with a grid of values of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.

`nfolds` The number of folds. Set to 10 by default.

`gam` A vector of values between 0 and 1. It is the weight of the pooled covariance and the diagonal matrix.

`del` A vector of values between 0 and 1. It is the weight of the LDA and QDA.

`ncores` The number of cores to use. If it is more than 1 parallel computing is performed. It is advisable to use it if you have many observations and or many variables, otherwise it will slow down th process.

`folds` If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.

`stratified` Do you want the folds to be created in a stratified way? TRUE or FALSE.

`seed` You can specify your own seed number here or leave it NULL.

`graph` If graph is TRUE (default value) a plot will appear.

Details

A k-fold cross validation is performed.

Value

For the alfa.rda a list including:

`res` The estimated optimal rate and the best values of α, γ and δ.

`percent` For the best value of α the averaged over all folds best rates of correct classification. It is a matrix, where rows correspond to the γ values and columns correspond to δ values.

`se` The estimated standard errors of the "percent" matrix.

`runtime` The runtime of the cross-validation procedure.

For the alfa.fda a graph (if requested) with the estimated performance for each value of α and a list including:

`per` The performance of the fda in each fold for each value of α.

`performance` The average performance for each value of α.

`opt_a` The optimal value of α.

`runtime` The runtime of the cross-validation procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>.

References

Friedman Jerome, Trevor Hastie and Robert Tibshirani (2009). The elements of statistical learning, 2nd edition. Springer, Berlin

Tsagris M.T., Preston S. and Wood A.T.A. (2016). Improved classification for compositional data using the α-transformation. Journal of Classification, 33(2):243-261.

Hastie, Tibshirani and Buja (1994). Flexible Disriminant Analysis by Optimal Scoring. Journal of the American Statistical Association, 89(428):1255-1270.

See Also

alfa.rda, alfanb.tune, cv.dda, compknn.tune cv.compnb

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
ina <- fgl[, 10]
moda <- alfarda.tune(x, ina, a = seq(0.7, 1, by = 0.1), nfolds = 10,
gam = seq(0.1, 0.3, by = 0.1), del = seq(0.1, 0.3, by = 0.1) )
```

Cross validation for the ridge regression

Cross validation for the ridge regression

Description

Cross validation for the ridge regression is performed. There is an option for the GCV criterion which is automatic.

Usage

```
ridge.tune(y, x, nfolds = 10, lambda = seq(0, 2, by = 0.1), folds = NULL,
ncores = 1, seed = NULL, graph = FALSE)
```

Arguments

`y` A numeric vector containing the values of the target variable. If the values are proportions or percentages, i.e. strictly within 0 and 1 they are mapped into R using the logit transformation.

`x` A numeric matrix containing the variables.

`nfolds` The number of folds in the cross validation.

`lambda` A vector with the a grid of values of λ to be used.

`folds` If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.

`ncores` The number of cores to use. If it is more than 1 parallel computing is performed.

`seed` You can specify your own seed number here or leave it NULL.

`graph` If graph is set to TRUE the performances for each fold as a function of the λ values will appear.

Details

A k-fold cross validation is performed. This function is used by `alfaridge.tune`.

Value

A list including:

`msp` The performance of the ridge regression for every fold.

`mspe` The values of the mean prediction error for each value of λ.

`lambda` The value of λ which corresponds to the minimum MSPE.

`performance` The minimum MSPE.

`runtime` The time required by the cross-validation procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Giorgos Athineou <gioathineou@gmail.com> and Michail Tsagris <`mtsagris@uoc.gr`>.

References

Hoerl A.E. and R.W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1):55-67.

Brown P. J. (1994). Measurement, Regression and Calibration. Oxford Science Publications.

See Also

`ridge.reg`, `alfaridge.tune`

Examples

```
y <- as.vector(iris[, 1])
x <- as.matrix(iris[, 2:4])
ridge.tune( y, x, nfolds = 10, lambda = seq(0, 2, by = 0.1), graph = TRUE )
```

Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation

Cross validation for the ridge regression with compositional data as predictor using the α-transformation

---

Description

Cross validation for the ridge regression is performed. There is an option for the GCV criterion which is automatic. The predictor variables are compositional data and the α-transformation is applied first.

Usage

```
alfaridge.tune(y, x, nfolds = 10, a = seq(-1, 1, by = 0.1),
lambda = seq(0, 2, by = 0.1), folds = NULL, ncores = 1,
graph = TRUE, col.nu = 15, seed = NULL)
```

Arguments

`y`	A numeric vector containing the values of the target variable. If the values are proportions or percentages, i.e. strictly within 0 and 1 they are mapped into R using the logit transformation.
`x`	A numeric matrix containing the compositional data, i.e. the predictor variables. Zero values are allowed.
`nfolds`	The number of folds in the cross validation.
`a`	A vector with the a grid of values of α to be used.
`lambda`	A vector with the a grid of values of λ to be used.
`folds`	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
`ncores`	The number of cores to use. If it is more than 1 parallel computing is performed. It is advisable to use it if you have many observations and or many variables, otherwise it will slow down th process.
`graph`	If graph is TRUE (default value) a filled contour plot will appear.
`col.nu`	A number parameter for the filled contour plot, taken into account only if graph is TRUE.
`seed`	You can specify your own seed number here or leave it NULL.

Details

A k-fold cross validation is performed.

Value

If graph is TRUE a fileld contour a filled contour will appear. A list including:

`mspe` The MSPE where rows correspond to the α values and the columns to the number of principal components.

`best.par` The best pair of α and λ.

`performance` The minimum mean squared error of prediction.

`runtime` The run time of the cross-validation procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Giorgos Athineou <gioathineou@gmail.com> and Michail Tsagris <`mtsagris@uoc.gr`>.

References

Hoerl A.E. and R.W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1):55-67.

Brown P. J. (1994). Measurement, Regression and Calibration. Oxford Science Publications.

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

`alfa.ridge`, `ridge.tune`

Examples

```
library(MASS)
y <- as.vector(fgl[, 1])
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
alfaridge.tune( y, x, nfolds = 10, a = seq(0.1, 1, by = 0.1),
lambda = seq(0, 1, by = 0.1) )
```

`Cross validation for the TFLR model`

Cross validation for the TFLR model

Description

Cross validation for the TFLR model.

Usage

```
cv.tflr(y, x, nfolds = 10, folds = NULL, seed = NULL)
```

Arguments

`y`	A matrix with compositional response data. Zero values are allowed.
`x`	A matrix with compositional predictors. Zero values are allowed.
`nfolds`	The number of folds to be used. This is taken into consideration only if the folds argument is not supplied.
`folds`	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
`seed`	If seed is TRUE the results will always be the same.

Details

A k-fold cross validation for the transformation-free linear regression for compositional responses and predictors is performed.

Value

A list including:

`runtime`	The runtime of the cross-validation procedure.
`kl`	The Kullback-Leibler divergences for all runs.
`js`	The Jensen-Shannon divergences for all runs.
`perf`	The average Kullback-Leibler divergence and average Jensen-Shannon divergence.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Fiksel J., Zeger S. and Datta A. (2022). A transformation-free linear regression for compositional outcomes and predictors. Biometrics, 78(3): 974–987.

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

`tflr`, `cv.scls`, `klalfapcr.tune`

Examples

```
library(MASS)
y <- rdiri(100, runif(3, 1, 3))
x <- as.matrix(fgl[1:100, 2:9])
x <- x / rowSums(x)
mod <- cv.tflr(y, x)
mod
```

`Cross-validation for LASSO with compositional predictors using the alpha-transformation`

Cross-validation for LASSO with compositional predictors using the $alpha$-transformation

---

Description

Cross-validation for LASSO with compositional predictors using the $alpha$-transformation.

Usage

```
alfalasso.tune(y, x, a = seq(-1, 1, by = 0.1), model = "gaussian", lambda = NULL,
type.measure = "mse", nfolds = 10, folds = NULL, stratified = FALSE)
```

Arguments

`y` A numerical vector or a matrix for multinomial logistic regression.

`x` A numerical matrix containing the predictor variables, compositional data, where zero values are allowed..

`a` A vector with a grid of values of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.

`model` The type of the regression model, "gaussian", "binomial", "poisson", "multinomial", or "mgaussian".

`lambda`	**This information is copied from the package glmnet.** A user supplied lambda sequence. Typical usage is to have the program compute its own lambda sequence based on nlambda and lambda.min.ratio. Supplying a value of lambda overrides this. WARNING: use with care. Avoid supplying a single value for lambda (for predictions after CV use predict() instead). Supply instead a decreasing sequence of lambda values. glmnet relies on its warms starts for speed, and its often faster to fit a whole path than compute a single fit.
`type.measure`	**This information is taken from the package glmnet.** The loss function to use for cross-validation. For gaussian models this can be "mse", "deviance" for logistic and poisson regression, "class" applies to binomial and multinomial logistic regression only, and gives misclassification error. "auc" is for two-class logistic regression only, and gives The area under the ROC curve. "mse" or "mae" (mean absolute error) can be used by all models.
`nfolds`	The number of folds. Set to 10 by default.
`folds`	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
`stratified`	Do you want the folds to be created in a stratified way? TRUE or FALSE.

Details

The function uses the glmnet package to perform LASSO penalised regression. For more details see the function in that package.

Value

A matrix with two columns and number of rows equal to the number of α values used. Each row contains, the optimal value of the λ penalty parameter for the LASSO and the optimal value of the loss function, for each value of α.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Friedman, J., Hastie, T. and Tibshirani, R. (2010) Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, Vol. 33(1), 1–22.

See Also

`alfa.lasso`, `cv.lasso.klcompreg`, `lasso.compreg`, `alfa.knn.reg`

Examples

```
y <- iris[, 1]
x <- rdiri(150, runif(20, 2, 5) )
mod <- alfalasso.tune( y, x, a = c(0.2, 0.5, 1) )
```

Cross-validation for the alpha-SCLS model

Cross-validation for the $alpha$-SCLS model

Description

Cross-validation for the $alpha$-SCLS model.

Usage

```
cv.ascls(y, x, a = seq(0.1, 1, by = 0.1), nfolds = 10, folds = NULL, seed = NULL)
```

Arguments

y	A numerical matrix with the simplicial response data. Zero values are allowed.
x	A matrix with the simplicial predictor variables. Zero values are allowed.
a	A vector or a single number of values of the α-parameter. This has to be different from zero, and it can take negative values if there are no zeros in the simplicial response (y).
nfolds	The number of folds for the K-fold cross validation, set to 10 by default.
folds	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
seed	You can specify your own seed number here or leave it NULL.

Details

The K-fold cross validation is performed in order to select the optimal value for α of the α-SCLS model.

Value

A list including:

runtime	The runtime of the cross-validation procedure.
kl	The Kullback-Leibler divergence for every value of α.
js	The Jensen-Shannon divergence for every value of α.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

ascls, cv.atflr

Examples

```
library(MASS)
y <- rdiri( 214, runif(4, 1, 3) )
x <- as.matrix( fgl[, 2:9] )
mod <- cv.ascls(y, x, nfolds = 5)
```

Cross-validation for the alpha-TFLR model

Cross-validation for the $alpha$-TFLR model

Description

Cross-validation for the $alpha$-TFLR model.

Usage

```
cv.atflr(y, x, a = seq(0.1, 1, by = 0.1), nfolds = 10, folds = NULL, seed = NULL)
```

Arguments

y	A numerical matrix with the simplicial response data. Zero values are allowed.
x	A matrix with the simplicial predictor variables. Zero values are allowed.
a	A vector or a single number of values of the α-parameter. This has to be different from zero, and it can take negative values if there are no zeros in the simplicial response (y).
nfolds	The number of folds for the K-fold cross validation, set to 10 by default.
folds	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
seed	You can specify your own seed number here or leave it NULL.

Details

The K-fold cross validation is performed in order to select the optimal value for α of the α-TFLR model.

Value

A list including:

runtime	The runtime of the cross-validation procedure.
kl	The Kullback-Leibler divergence for every value of α.
js	The Jensen-Shannon divergence for every value of α.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Fiksel J., Zeger S. and Datta A. (2022). A transformation-free linear regression for compositional outcomes and predictors. Biometrics, 78(3): 974–987.

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

atflr, cv.ascls

Examples

```
library(MASS)
y <- rdiri( 214, runif(4, 1, 3) )
x <- as.matrix( fgl[, 2:9] )
mod <- cv.ascls(y, x, nfolds = 2, a = c(0.5, 1))
```

Cross-validation for the Dirichlet discriminant analysis

Cross-validation for the Dirichlet discriminant analysis

Description

Cross-validation for the Dirichlet discriminant analysis.

Usage

```
cv.dda(x, ina, nfolds = 10, folds = NULL, stratified = TRUE, seed = NULL)
```

Arguments

x	A matrix with the available data, the predictor variables.
ina	A vector of data. The response variable, which is categorical (factor is acceptable).
folds	A list with the indices of the folds.
nfolds	The number of folds to be used. This is taken into consideration only if "folds" is NULL.
stratified	Do you want the folds to be selected using stratified random sampling? This preserves the analogy of the samples of each group. Make this TRUE if you wish.
seed	If you set this to TRUE, the same folds will be created every time.

Details

This function estimates the performance of the Dirichlet discriminant analysis via k-fold cross-validation.

Value

A list including:

`percent` The percentage of correct classification

`runtime` The duration of the cross-validation proecdure.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Friedman J., Hastie T. and Tibshirani R. (2017). The elements of statistical learning. New York: Springer.

Thomas P. Minka (2003). Estimating a Dirichlet distribution. http://research.microsoft.com/en-us/um/people/minka/papers/dirichlet/minka-dirichlet.pdf

See Also

`dda`, `alfanb.tune`, `alfarda.tune`, `compknn.tune`, `cv.compnb`

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
mod <- cv.dda(x, ina = iris[, 5] )
```

`Cross-validation for the LASSO Kullback-Leibler divergence based regression`

Cross-validation for the LASSO Kullback-Leibler divergence based regression

---

Description

Cross-validation for the LASSO Kullback-Leibler divergence based regression.

Usage

```
cv.lasso.klcompreg(y, x, alpha = 1, type = "grouped", nfolds = 10,
folds = NULL, seed = NULL, graph = FALSE)
```

Arguments

`y` A numerical matrix with compositional data with or without zeros.

`x` A matrix with the predictor variables.

`alpha` The elastic net mixing parameter, with $0 \leq \alpha \leq 1$. The penalty is defined as a weighted combination of the ridge and of the Lasso regression. When $\alpha = 1$ LASSO is applied, while $\alpha = 0$ yields the ridge regression.

`type` **This information is copied from the package glmnet.**. If "grouped" then a grouped lasso penalty is used on the multinomial coefficients for a variable. This ensures they are all in our out together. The default in our case is "grouped".

`nfolds` The number of folds for the K-fold cross validation, set to 10 by default.

`folds` If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.

`seed` You can specify your own seed number here or leave it NULL.

`graph` If graph is TRUE (default value) a filled contour plot will appear.

Details

The K-fold cross validation is performed in order to select the optimal value for λ, the penalty parameter in LASSO.

Value

The outcome is the same as in the R package glmnet. The extra addition is that if "graph = TRUE", then the plot of the cross-validated object is returned. The contains the logarithm of λ and the deviance. The numbers on top of the figure show the number of set of coefficients for each component, that are not zero.

Author(s)

Michail Tsagris and Abdulaziz Alenazi.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Abdulaziz Alenazi <`a.alenazi@nbu.edu.sa`>.

References

Alenazi, A. A. (2022). f-divergence regression models for compositional data. Pakistan Journal of Statistics and Operation Research, 18(4): 867–882.

Friedman, J., Hastie, T. and Tibshirani, R. (2010) Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, Vol. 33(1), 1-22.

See Also

`lasso.klcompreg`, `lassocoef.plot`, `lasso.compreg`, `cv.lasso.compreg`, `kl.compreg`

Examples

```
library(MASS)
y <- rdiri( 214, runif(4, 1, 3) )
x <- as.matrix( fgl[, 2:9] )
mod <- cv.lasso.klcompreg(y, x)
```

Cross-validation for the LASSO log-ratio regression with compositional response

Cross-validation for the LASSO log-ratio regression with compositional response

Description

Cross-validation for the LASSO log-ratio regression with compositional response.

Usage

```
cv.lasso.compreg(y, x, alpha = 1, nfolds = 10,
folds = NULL, seed = NULL, graph = FALSE)
```

Arguments

y	A numerical matrix with compositional data. Zero values are not allowed as the additive log-ratio transformation (alr) is applied to the compositional response prior to implementing the LASSO algortihm.
x	A matrix with the predictor variables.
alpha	The elastic net mixing parameter, with $0 \leq \alpha \leq 1$. The penalty is defined as a weighted combination of the ridge and of the Lasso regression. When $\alpha = 1$ LASSO is applied, while $\alpha = 0$ yields the ridge regression.
nfolds	The number of folds for the K-fold cross validation, set to 10 by default.
folds	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
seed	You can specify your own seed number here or leave it NULL.
graph	If graph is TRUE (default value) a filled contour plot will appear.

Details

The K-fold cross validation is performed in order to select the optimal value for λ, the penalty parameter in LASSO.

Value

The outcome is the same as in the R package glmnet. The extra addition is that if "graph = TRUE", then the plot of the cross-validated object is returned. The contains the logarithm of λ and the mean squared error. The numbers on top of the figure show the number of set of coefficients for each component, that are not zero.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Friedman, J., Hastie, T. and Tibshirani, R. (2010) Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, Vol. 33(1), 1-22.

See Also

lasso.compreg, lasso.klcompreg, lassocoef.plot, cv.lasso.klcompreg, comp.reg

Examples

```
library(MASS)
y <- rdiri( 214, runif(4, 1, 3) )
x <- as.matrix( fgl[, 2:9] )
mod <- cv.lasso.compreg(y, x)
```

Cross-validation for the naive Bayes classifiers for compositional data

Cross-validation for the naive Bayes classifiers for compositional data

---

Description

Cross-validation for the naive Bayes classifiers for compositional data.

Usage

```
cv.compnb(x, ina, type = "beta", folds = NULL, nfolds = 10,
      stratified = TRUE, seed = NULL, pred.ret = FALSE)
```

Arguments

x	A matrix with the available data, the predictor variables.
ina	A vector of data. The response variable, which is categorical (factor is acceptable).
type	The type of naive Bayes, "beta", "logitnorm", "cauchy", "laplace", "gamma", "normlog" or "weibull". For the last 4 distributions, the negative of the logarithm of the compositional data is applied first.
folds	A list with the indices of the folds.
nfolds	The number of folds to be used. This is taken into consideration only if "folds" is NULL.

`stratified` Do you want the folds to be selected using stratified random sampling? This preserves the analogy of the samples of each group. Make this TRUE if you wish.

`seed` You can specify your own seed number here or leave it NULL.

`pred.ret` If you want the predicted values returned set this to TRUE.

Value

A list including:

`preds` If pred.ret is TRUE the predicted values for each fold are returned as elements in a list.

`crit` A vector whose length is equal to the number of k and is the accuracy metric for each k. For the classification case it is the percentage of correct classification.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Friedman J., Hastie T. and Tibshirani R. (2017). The elements of statistical learning. New York: Springer.

See Also

`comp.nb`

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
mod <- cv.compnb(x, ina = iris[, 5] )
```

`Cross-validation for the naive Bayes classifiers for compositional data using the alpha-transformation`

Cross-validation for the naive Bayes classifiers for compositional data using the α-transformation

---

Description

Cross-validation for the naive Bayes classifiers for compositional data using the α-transformation.

Usage

```
alfanb.tune(x, ina, a = seq(-1, 1, by = 0.1), type = "gaussian",
folds = NULL, nfolds = 10, stratified = TRUE, seed = NULL)
```

Arguments

`x` A matrix with the available data, the predictor variables.

`ina` A vector of data. The response variable, which is categorical (factor is acceptable).

`a` The value of α for the α-transformation. This can be a vector of values or a single number.

`type` The type of naive Bayes, "gaussian", "cauchy" or "laplace".

`folds` A list with the indices of the folds.

`nfolds` The number of folds to be used. This is taken into consideration only if "folds" is NULL.

`stratified` Do you want the folds to be selected using stratified random sampling? This preserves the analogy of the samples of each group. Make this TRUE if you wish.

`seed` You can specify your own seed number here or leave it NULL.

Details

This function estimates the performance of the naive Bayes classifier for each value of α of the α-transformation.

Value

A list including:

`crit` A vector whose length is equal to the number of k and is the accuracy metric for each k. For the classification case it is the percentage of correct classification.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Friedman J., Hastie T. and Tibshirani R. (2017). The elements of statistical learning. New York: Springer.

See Also

`alfa.nb`, `alfarda.tune`, `compknn.tune`, `cv.dda`, `cv.compnb`

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
mod <- alfanb.tune(x, ina = iris[, 5], a = c(0, 0.1, 0.2) )
```

Cross-validation for the SCLS model

Cross-validation for the SCLS model

Description

Cross-validation for the SCLS model.

Usage

```
cv.scls(y, x, nfolds = 10, folds = NULL, seed = NULL)
```

Arguments

y	A matrix with compositional response data. Zero values are allowed.
x	A matrix with compositional predictors. Zero values are allowed.
nfolds	The number of folds to be used. This is taken into consideration only if the folds argument is not supplied.
folds	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
seed	You can specify your own seed number here or leave it NULL.

Details

The function performs k-fold cross-validation for the least squares regression where the beta coefficients are constained to be positive and sum to 1.

Value

A list including:

runtime	The runtime of the cross-validation procedure.
kl	The Kullback-Leibler divergences for all runs.
js	The Jensen-Shannon divergences for all runs.
perf	The average Kullback-Leibler divergence and average Jensen-Shannon divergence.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

`scls`, `cv.tflr`, `klalfapcr.tune`

Examples

```
library(MASS)
set.seed(1234)
y <- rdiri(214, runif(3, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- cv.scls(y, x, nfolds = 5, seed = 12345)
mod
```

`Cross-validation for the SCRQ model`

Cross-validation for the SCRQ model

Description

Cross-validation for the SCRQ model.

Usage

```
cv.scrq(y, x, nfolds = 10, folds = NULL, seed = NULL)
```

Arguments

`y`	A matrix with compositional response data. Zero values are allowed.
`x`	A matrix with compositional predictors. Zero values are allowed.
`nfolds`	The number of folds to be used. This is taken into consideration only if the folds argument is not supplied.
`folds`	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
`seed`	You can specify your own seed number here or leave it NULL.

Details

The function performs k-fold cross-validation for the absolute regression where the beta coefficients are constained to be positive and sum to 1.

Value

A list including:

`runtime` The runtime of the cross-validation procedure.

`kl` The Kullback-Leibler divergences for all runs.

`js` The Jensen-Shannon divergences for all runs.

`perf` The average Kullback-Leibler divergence and average Jensen-Shannon divergence.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

`scrq`, `cv.scls`, `cv.tflr`

Examples

```
y <- rdiri(500, runif(3, 1, 3))
x <- rdiri(500, runif(3, 1, 3))
mod <- cv.scrq(y, x, nfolds = 5)
```

`Density of compositional data from Gaussian mixture models`
Simulation of compositional data from Gaussian mixture models

---

Description

Simulation of compositional data from Gaussian mixture models.

Usage

```
dmix.compnorm(x, mu, sigma, prob, type = ”alr”, logged = TRUE)
```

Arguments

`x` A vector or a matrix with compositional data.

`prob` A vector with mixing probabilities. Its length is equal to the number of clusters.

`mu` A matrix where each row corresponds to the mean vector of each cluster.

`sigma` An array consisting of the covariance matrix of each cluster.

`type` The type of trasformation used, either the additive log-ratio ("alr"), the isometric log-ratio ("ilr") or the pivot coordinate ("pivot") transformation.

`logged` A boolean variable specifying whether the logarithm of the density values to be returned. It is set to TRUE by default.

Details

A sample from a multivariate Gaussian mixture model is generated.

Value

A vector with the density values.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Ryan P. Browne, Aisha ElSherbiny and Paul D. McNicholas (2015). R package mixture: Mixture Models for Clustering and Classification.

See Also

`mix.compnorm`, `bic.mixcompnorm`

Examples

```
p <- c(1/3, 1/3, 1/3)
mu <- matrix(nrow = 3, ncol = 4)
s <- array( dim = c(4, 4, 3) )
x <- as.matrix(iris[, 1:4])
ina <- as.numeric(iris[, 5])
mu <- rowsum(x, ina) / 50
s[, , 1] <- cov(x[ina == 1, ])
s[, , 2] <- cov(x[ina == 2, ])
s[, , 3] <- cov(x[ina == 3, ])
y <- rmixcomp(100, p, mu, s, type = "alr")$x
mod <- dmix.compnorm(y, mu, s, p)
```

Density of the Flexible Dirichlet distribution

Density of the Flexible Dirichlet distribution

Description

Density of the Flexible Dirichlet distribution

Usage

```
dfd(x, alpha, prob, tau)
```

Arguments

x	A vector or a matrix with compositional data.
alpha	A vector of the non-negative α parameters.
prob	A vector of the clusters' probabilities. It must sum to one.
tau	The non-negative scalar tau parameter.

Details

For more information see the references and the package FlxeDir.

Value

The density value(s).

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Ongaro A. and Migliorati S. (2013). A generalization of the Dirichlet distribution. Journal of Multivariate Analysis, 114, 412–426.

Migliorati S., Ongaro A. and Monti G. S. (2017). A structured Dirichlet mixture model for compositional data: inferential and applicative issues. Statistics and Computing, 27, 963–983.

See Also

rfd

Examples

```
alpha <- c(12, 11, 10)
prob <- c(0.25, 0.25, 0.5)
tau <- 8
x <- rfd(20, alpha, prob, tau)
dfd(x, alpha, prob, tau)
```

`Density of the folded normal distribution`

Density of the folded model normal distribution

Description

Density of the folded model normal distribution.

Usage

```
dfolded(x, a, p, mu, su, logged = TRUE)
```

Arguments

`x`	A vector or a matrix with compositional data. No zeros are allowed.
`a`	The value of α.
`p`	The probability inside the simplex of the folded model.
`mu`	The mean vector.
`su`	The covariance matrix.
`logged`	A boolean variable specifying whether the logarithm of the density values to be returned. It is set to TRUE by default.

Details

Density values of the folded model.

Value

The density value(s).

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris M. and Stewart C. (2020). A folded model for compositional data analysis. Australian and New Zealand Journal of Statistics, 62(2): 249-277. https://arxiv.org/pdf/1802.07330.pdf

See Also

rfolded, a.est, folded.contour

Examples

```
s <- c(0.1490676523, -0.4580818209,  0.0020395316, -0.0047446076, -0.4580818209,
1.5227259250,  0.0002596411,  0.0074836251,  0.0020395316,  0.0002596411,
0.0365384838, -0.0471448849, -0.0047446076,  0.0074836251, -0.0471448849,
0.0611442781)
s <- matrix(s, ncol = 4)
m <- c(1.715, 0.914, 0.115, 0.167)
x <- rfolded(100, m, s, 0.5)
mod <- a.est(x)
den <- dfolded(x, mod$best, mod$p, mod$mu, mod$su)
```

Density values of a Dirichlet distribution

Density values of a Dirichlet distribution

Description

Density values of a Dirichlet distribution.

Usage

```
ddiri(x, a, logged = TRUE)
```

Arguments

x	A matrix containing compositional data. This can be a vector or a matrix with the data.
a	A vector of parameters. Its length must be equal to the number of components, or columns of the matrix with the compositional data and all values must be greater than zero.
logged	A boolean variable specifying whether the logarithm of the density values to be returned. It is set to TRUE by default.

Details

The density of the Dirichlet distribution for a vector or a matrix of compositional data is returned.

Value

A vector with the density values.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Ng Kai Wang, Guo-Liang Tian and Man-Lai Tang (2011). Dirichlet and related distributions: Theory, methods and applications. John Wiley & Sons.

See Also

dgendiri, diri.nr, diri.est, diri.contour, rdiri, dda

Examples

```
x <- rdiri( 100, c(5, 7, 4, 8, 10, 6, 4) )
a <- diri.est(x)
f <- ddiri(x, a$param)
sum(f)
a
```

Density values of a generalised Dirichlet distribution

Density values of a generalised Dirichlet distribution

Description

Density values of a generalised Dirichlet distribution.

Usage

```
dgendiri(x, a, b, logged = TRUE)
```

Arguments

x	A matrix containing compositional data. This can be a vector or a matrix with the data.
a	A numerical vector with the shape parameter values of the Gamma distribution.
b	A numerical vector with the scale parameter values of the Gamma distribution.
logged	A boolean variable specifying whether the logarithm of the density values to be returned. It is set to TRUE by default.

Details

The density of the Dirichlet distribution for a vector or a matrix of compositional data is returned.

Value

A vector with the density values.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Ng Kai Wang, Guo-Liang Tian and Man-Lai Tang (2011). Dirichlet and related distributions: Theory, methods and applications. John Wiley & Sons.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

ddiri, rgendiri, diri.est, diri.contour, rdiri, dda

Examples

```
a <- c(1, 2, 3)
b <- c(2, 3, 4)
x <- rgendiri(100, a, b)
y <- dgendiri(x, a, b)
```

Density values of a mixture of Dirichlet distributions

Density values of a mixture of Dirichlet distributions

Description

Density values of a mixture of Dirichlet distributions.

Usage

```
dmixdiri(x, a, prob, logged = TRUE)
```

Arguments

x	A vector or a matrix with compositional data. Zeros are not allowed.
a	A matrix where each row contains the parameters of each Dirichlet component.
prob	A vector with the mixing probabilities.
logged	A boolean variable specifying whether the logarithm of the density values to be returned. It is set to TRUE by default.

Details

The density of the mixture of Dirichlet distribution for a vector or a matrix of compositional data is returned.

Value

A vector with the density values.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Ye X., Yu Y. K. and Altschul S. F. (2011). On the inference of Dirichlet mixture priors for protein sequence comparison. Journal of Computational Biology, 18(8), 941-954.

See Also

rmixdiri, mixdiri.contour

Examples

```
a <- matrix( c(12, 30, 45, 32, 50, 16), byrow = TRUE,ncol = 3)
prob <- c(0.5, 0.5)
x <- rmixdiri(100, a, prob)$x
f <- dmixdiri(x, a, prob)
```

Dirichlet discriminant analysis

Dirichlet discriminant analysis

Description

Dirichlet discriminant analysis.

Usage

```
dda(xnew, x, ina)
```

Arguments

xnew	A matrix with the new compositional predictor data whose class you want to predict. Zeros are allowed.
x	A matrix with the available compositional predictor data. Zeros are allowed.
ina	A vector of data. The response variable, which is categorical (factor is acceptable).

Details

The funcitons performs maximum likelihood discriminant analysis using the Dirichlet distribution.

Value

A vector with the estimated group.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Friedman J., Hastie T. and Tibshirani R. (2017). The elements of statistical learning. New York: Springer.

Thomas P. Minka (2003). Estimating a Dirichlet distribution. http://research.microsoft.com/en-us/um/people/minka/papers/dirichlet/minka-dirichlet.pdf

Ng Kai Wang, Guo-Liang Tian and Man-Lai Tang (2011). Dirichlet and related distributions: Theory, methods and applications. John Wiley & Sons.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

cv.dda, comp.nb, alfa.rda, alfa.knn,comp.knn, mix.compnorm, diri.reg, zadr

Examples

```
x <- Compositional::rdiri(100, runif(5) )
ina <- rbinom(100, 1, 0.5) + 1
mod <- dda(x, x, ina )
```

`Dirichlet random values simulation`

Dirichlet random values simulation

Description

Dirichlet random values simulation.

Usage

```
rdiri(n, a)
```

Arguments

n	The sample size, a numerical value.
a	A numerical vector with the parameter values.

Details

The algorithm is straightforward, for each vector, independent gamma values are generated and then divided by their total sum.

Value

A matrix with the simulated data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>.

References

Ng Kai Wang, Guo-Liang Tian and Man-Lai Tang (2011). Dirichlet and related distributions: Theory, methods and applications. John Wiley & Sons.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

diri.est, diri.nr, diri.contour, rgendiri

Examples

```
x <- rdiri( 100, c(5, 7, 1, 3, 10, 2, 4) )
diri.est(x)
```

Dirichlet regression *Dirichlet regression*

Description

Dirichlet regression.

Usage

```
diri.reg(y, x, plot = FALSE, xnew = NULL)

diri.reg2(y, x, xnew = NULL)

diri.reg3(y, x, xnew = NULL)
```

Arguments

`y`	A matrix with the compositional data (dependent variable). Zero values are not allowed.
`x`	The predictor variable(s), they can be either continuous or categorical or both.
`plot`	A boolean variable specifying whether to plot the leverage values of the observations or not. This is taken into account only when xnew = NULL.
`xnew`	If you have new data use it, otherwise leave it NULL.

Details

A Dirichlet distribution is assumed for the regression. This involves numerical optimization. The function "diri.reg2()" allows for the covariates to be linked with the precision parameter ϕ via the exponential link function $\phi = e^{x*b}$. The function "diri.reg3()" links the covariates to the alpha parameters of the Dirichlet distribution, i.e. it uses the classical parametrization of the distribution. This means, that there is a set of regression parameters for each component.

Value

A list including:

`runtime`	The time required by the regression.
`loglik`	The value of the log-likelihood.
`phi`	The precision parameter. If covariates are linked with it (function "diri.reg2()"), this will be a vector.
`phipar`	The coefficients of the phi parameter if it is linked to the covariates.
`std.phi`	The standard errors of the coefficients of the phi parameter is it linked to the covariates.
`log.phi`	The logarithm of the precision parameter.
`std.logphi`	The standard error of the logarithm of the precision parameter.
`be`	The beta coefficients.
`seb`	The standard error of the beta coefficients.
`sigma`	Th covariance matrix of the regression parameters (for the mean vector and the phi parameter)".
`lev`	The leverage values.
`est`	For the "diri.reg" this contains the fitted or the predicted values (if xnew is not NULL). For the "diri.reg2" if xnew is NULL, this is also NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>` and Giorgos Athineou <gioathineou@gmail.com>.

References

Maier, Marco J. (2014) DirichletReg: Dirichlet Regression for Compositional Data in R. Research Report Series/Department of Statistics and Mathematics, 125. WU Vienna University of Economics and Business, Vienna. http://epub.wu.ac.at/4077/1/Report125.pdf

Gueorguieva, Ralitza, Robert Rosenheck, and Daniel Zelterman (2008). Dirichlet component regression and its applications to psychiatric data. Computational statistics & data analysis 52(12): 5344-5355.

Ng Kai Wang, Guo-Liang Tian and Man-Lai Tang (2011). Dirichlet and related distributions: Theory, methods and applications. John Wiley & Sons.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`js.compreg`, `kl.compreg`, `ols.compreg`, `comp.reg`, `alfa.reg`, `diri.nr`, `dda`

Examples

```
x <- as.vector(iris[, 4])
y <- as.matrix(iris[, 1:3])
y <- y / rowSums(y)
mod1 <- diri.reg(y, x)
mod2 <- diri.reg2(y, x)
mod3 <- comp.reg(y, x)
```

`Distance based regression models for proportions`

Distance based regression models for proportions

Description

Distance based regression models for proportions.

Usage

```
ols.prop.reg(y, x, cov = FALSE, tol = 1e-07, maxiters = 100)
helling.prop.reg(y, x, tol = 1e-07, maxiters = 100)
```

Arguments

`y`	A numerical vector proportions. 0s and 1s are allowed.
`x`	A matrix or a data frame with the predictor variables.
`cov`	Should the covariance matrix be returned? TRUE or FALSE.
`tol`	The tolerance value to terminate the Newton-Raphson algorithm. This is set to 10^{-9} by default.
`maxiters`	The maximum number of iterations before the Newton-Raphson is terminated automatically.

Details

We are using the Newton-Raphson, but unlike R's built-in function "glm" we do no checks and no extra calculations, or whatever. Simply the model. The functions accept binary responses as well (0 or 1).

Value

A list including:

`sse`	The sum of squres of errors for the "ols.prop.reg" function.
`be`	The estimated regression coefficients.
`seb`	The standard error of the regression coefficients if "cov" is TRUE.
`covb`	The covariance matrix of the regression coefficients in "ols.prop.reg" if "cov" is TRUE.
`H`	The Hellinger distance between the true and the obseervd proportions in "helling.prop.reg".
`iters`	The number of iterations required by the Newton-Raphson.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Papke L. E. & Wooldridge J. (1996). Econometric methods for fractional response variables with an application to 401(K) plan participation rates. Journal of Applied Econometrics, 11(6): 619–632.

McCullagh, Peter, and John A. Nelder. Generalized linear models. CRC press, USA, 2nd edition, 1989.

See Also

`propreg`, `beta.reg`

Examples

```
y <- rbeta(100, 1, 4)
x <- matrix(rnorm(100 * 2), ncol = 2)
a1 <- ols.prop.reg(y, x)
a2 <- helling.prop.reg(y, x)
```

Divergence based regression for compositional data
Divergence based regression for compositional data

Description

Regression for compositional data based on the Kullback-Leibler the Jensen-Shannon divergence and the symmetric Kullback-Leibler divergence.

Usage

```
kl.compreg(y, x, con = TRUE, B = 1, ncores = 1, xnew = NULL, tol = 1e-07, maxiters = 50)
js.compreg(y, x, con = TRUE, B = 1, ncores = 1, xnew = NULL)
tv.compreg(y, x, con = TRUE, B = 1, ncores = 1, xnew = NULL)
symkl.compreg(y, x, con = TRUE, B = 1, ncores = 1, xnew = NULL)
hellinger.compreg(y, x, con = TRUE, B = 1, ncores = 1, xnew = NULL)
```

Arguments

y	A matrix with the compositional data (dependent variable). Zero values are allowed.
x	The predictor variable(s), they can be either continnuous or categorical or both.
con	If this is TRUE (default) then the constant term is estimated, otherwise the model includes no constant term.
B	If B is greater than 1 bootstrap estimates of the standard error are returned. If B=1, no standard errors are returned.
ncores	If ncores is 2 or more parallel computing is performed. This is to be used for the case of bootstrap. If B=1, this is not taken into consideration.
xnew	If you have new data use it, otherwise leave it NULL.
tol	The tolerance value to terminate the Newton-Raphson procedure.
maxiters	The maximum number of Newton-Raphson iterations.

Details

In the kl.compreg() the Kullback-Leibler divergence is adopted as the objective function. In case of problematic convergence the "multinom" function by the "nnet" package is employed. This will obviously be slower. The js.compreg() uses the Jensen-Shannon divergence and the symkl.compreg() uses the symmetric Kullback-Leibler divergence. The tv.compreg() uses the Total Variation divergence. There is no actual log-likelihood for the last three regression models. The hellinger.compreg() minimizes the Hellinger distance.

Value

A list including:

`runtime`	The time required by the regression.
`iters`	The number of iterations required by the Newton-Raphson in the kl.compreg function.
`loglik`	The log-likelihood. This is actually a quasi multinomial regression. This is bascially half the negative deviance, or $-\sum_{i=1}^n y_i \log y_i/\hat{y}_i$.
`be`	The beta coefficients.
`covbe`	The covariance matrix of the beta coefficients, if bootstrap is chosen, i.e. if B > 1.
`est`	The fitted values of xnew if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Giorgos Athineou <gioathineou@gmail.com>.

References

Murteira Jose MR, and Joaquim JS Ramalho (2016). Regression analysis of multivariate fractional data. Econometric Reviews 35(4): 515-552.

Tsagris Michail (2015). A novel, divergence based, regression for compositional data. Proceedings of the 28th Panhellenic Statistics Conference, 15-18/4/2015, Athens, Greece. https://arxiv.org/pdf/1511.07600.pdf

Endres D. M. and Schindelin J. E. (2003). A new metric for probability distributions. Information Theory, IEEE Transactions on 49, 1858-1860.

Osterreicher F. and Vajda I. (2003). A new class of metric divergences on probability spaces and its applicability in statistics. Annals of the Institute of Statistical Mathematics 55, 639-653.

Alenazi A. A. (2022). f-divergence regression models for compositional data. Pakistan Journal of Statistics and Operation Research, 18(4): 867–882.

See Also

`diri.reg`, `ols.compreg`, `comp.reg`

Examples

```
library(MASS)
x <- as.vector(fgl[, 1])
y <- as.matrix(fgl[, 2:9])
y <- y / rowSums(y)
mod1<- kl.compreg(y, x, B = 1, ncores = 1)
mod2 <- js.compreg(y, x, B = 1, ncores = 1)
```

`Divergence based regression for compositional data with compositional data in the covariates side using the alpha-transformation`

Divergence based regression for compositional data with compositional data in the covariates side using the α-transformation

Description

Divergence based regression for compositional data with compositional data in the covariates side using the α-transformation.

Usage

```
kl.alfapcr(y, x, covar = NULL, a, k, xnew = NULL, B = 1, ncores = 1, tol = 1e-07,
maxiters = 50)
```

Arguments

`y`	A numerical matrixc with compositional data with or without zeros.
`x`	A matrix with the predictor variables, the compositional data. Zero values are allowed.
`covar`	If you have other covariates as well put themn here.
`a`	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
`k`	A number at least equal to 1. How many principal components to use.
`xnew`	A matrix containing the new compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.
`B`	If B is greater than 1 bootstrap estimates of the standard error are returned. If B=1, no standard errors are returned.
`ncores`	If ncores is 2 or more parallel computing is performed. This is to be used for the case of bootstrap. If B=1, this is not taken into consideration.
`tol`	The tolerance value to terminate the Newton-Raphson procedure.
`maxiters`	The maximum number of Newton-Raphson iterations.

Details

The α-transformation is applied to the compositional data first, the first k principal component scores are calcualted and used as predictor variables for the Kullback-Leibler divergence based regression model.

Value

A list including:

`runtime` The time required by the regression.

`iters` The number of iterations required by the Newton-Raphson in the kl.compreg function.

`loglik` The log-likelihood. This is actually a quasi multinomial regression. This is bascially minus the half deviance, or $-sum_{i=1}^n y_i \log y_i/\hat{y}_i$.

`be` The beta coefficients.

`seb` The standard error of the beta coefficients, if bootstrap is chosen, i.e. if B > 1.

`est` The fitted values of xnew if xnew is not NULL.

Author(s)

Initial code by Abdulaziz Alenazi. Modifications by Michail Tsagris.

R implementation and documentation: Abdulaziz Alenazi <a.alenazi@nbu.edu.sa> and Michail Tsagris <mtsagris@uoc.gr>.

References

Alenazi A. (2019). Regression for compositional data with compositional data as predictor variables with or without zero values. Journal of Data Science, 17(1): 219-238. https://jds-online.org/journal/JDS/article/136/file/pdf

Tsagris M. (2015). Regression analysis with compositional data containing zero values. Chilean Journal of Statistics, 6(2): 47-57. http://arxiv.org/pdf/1508.01913v1.pdf

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. http://arxiv.org/pdf/1106.1451.pdf

See Also

klalfapcr.tune, tflr, glm.pcr, alfapcr.tune

Examples

```
library(MASS)
y <- rdiri(214, runif(4, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- alfa.pcr(y = y, x = x, a = 0.7, k = 1)
mod
```

`Divergence matrix of compositional data` *Divergence matrix of compositional data*

Description

Divergence matrix of compositional data.

Usage

```
divergence(x, type = "kullback_leibler", vector = FALSE)
```

Arguments

`x`	A matrix with the compositional data.
`type`	This is either "kullback_leibler" (Kullback-Leibler, which computes the symmetric Kullback-Leibler divergence) or "jensen_shannon" (Jensen-Shannon) divergence.
`vector`	For return a vector instead a matrix.

Details

The function produces the distance matrix either using the Kullback-Leibler (distance) or the Jensen-Shannon (metric) divergence. The Kullback-Leibler refers to the symmetric Kullback-Leibler divergence.

Value

if the vector argument is FALSE a symmetric matrix with the divergences, otherwise a vector with the divergences.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Endres, D. M. and Schindelin, J. E. (2003). A new metric for probability distributions. Information Theory, IEEE Transactions on 49, 1858-1860.

Osterreicher, F. and Vajda, I. (2003). A new class of metric divergences on probability spaces and its applicability in statistics. Annals of the Institute of Statistical Mathematics 55, 639-653.

See Also

`comp.knn`, `js.compreg`

Examples

```
x <- as.matrix(iris[1:20, 1:4])
x <- x / rowSums(x)
divergence(x)
```

Empirical likelihood hypothesis testing for two mean vectors

Empirical likelihood hypothesis testing for two mean vectors

Description

Empirical likelihood hypothesis testing for two mean vectors.

Usage

```
el.test2(y1, y2, R = 0, ncores = 1, graph = FALSE)
```

Arguments

y1	A matrix containing the Euclidean data of the first group.
y2	A matrix containing the Euclidean data of the second group.
R	If R is 0, the classical chi-square distribution is used, if R = 1, the corrected chi-square distribution (James, 1954) is used and if R = 2, the modified F distribution (Krishnamoorthy and Yanping, 2006) is used. If R is greater than 3 bootstrap calibration is performed.
ncores	How many to cores to use.
graph	A boolean variable which is taken into consideration only when bootstrap calibration is performed. IF TRUE the histogram of the bootstrap test statistic values is plotted.

Details

The H_0 is that $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$ and the two constraints imposed by EL are

$$\frac{1}{n_j}\sum_{i=1}^{n_j}\left\{\left[1+\boldsymbol{\lambda}_j^T\left(\mathbf{x}_{ji}-\boldsymbol{\mu}\right)\right]^{-1}\left(\mathbf{x}_{ij}-\boldsymbol{\mu}\right)\right\}=\mathbf{0},$$

where $j = 1, 2$ and the $\boldsymbol{\lambda}_j s$ are Lagrangian parameters introduced to maximize the above expression. Note that the maximization of is with respect to the $\boldsymbol{\lambda}_j s$. The probabilities of the j-th sample have the following form

$$p_{ji} = \frac{1}{n_j}\left[1+\boldsymbol{\lambda}_j^T\left(\mathbf{x}_{ji}-\boldsymbol{\mu}\right)\right]^{-1}$$

. The log-likelihood ratio test statistic can be written as

$$\Lambda = \sum_{j=1}^{2}\sum_{i=1}^{n_j}\log n_j p_{ij}.$$

The test is implemented by searching for the mean vector that minimizes the sum of the two one sample EL test statistics.

Value

A list including:

`test`	The empirical likelihood test statistic value.
`modif.test`	The modified test statistic, either via the chi-square or the F distribution.
`dof`	Thre degrees of freedom of the chi-square or the F distribution.
`pvalue`	The asymptotic or the bootstrap p-value.
`mu`	The estimated common mean vector.
`runtime`	The runtime of the bootstrap calibration.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Amaral G.J.A., Dryden I.L. and Wood A.T.A. (2007). Pivotal bootstrap methods for k-sample problems in directional statistics and shape analysis. Journal of the American Statistical Association, 102(478): 695–707.

Owen A. B. (2001). Empirical likelihood. Chapman and Hall/CRC Press.

Owen A.B. (1988). Empirical likelihood ratio confidence intervals for a single functional. Biometrika, 75(2): 237–249.

Preston S.P. and Wood A.T.A. (2010). Two-Sample Bootstrap Hypothesis Tests for Three-Dimensional Labelled Landmark Data. Scandinavian Journal of Statistics, 37(4): 568–587.

See Also

`eel.test2`, `maovjames`, `hotel2T2`, `james`

Examples

```
el.test2( y1 = as.matrix(iris[1:25, 1:4]), y2 = as.matrix(iris[26:50, 1:4]), R = 0 )
el.test2( y1 = as.matrix(iris[1:25, 1:4]), y2 = as.matrix(iris[26:50, 1:4]), R = 1 )
el.test2( y1 =as.matrix(iris[1:25, 1:4]), y2 = as.matrix(iris[26:50, 1:4]), R = 2 )
```

Energy test of equality of distributions using the alpha-transformation

Energy test of equality of distributions using the α-transformation

Description

Energy test of equality of distributions using the α-transformation.

Usage

```
aeqdist.etest(x, sizes, a = 1, R = 999, ms = FALSE)
```

Arguments

`x`	A matrix with the compositional data with all groups stacked one under the other.
`sizes`	A numeric vector matrix with the sample sizes.
`a`	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied. If more than one values are supplied the energy distance of equality of distributions is applied for each value of α.
`R`	The number of permutations to apply in order to compute the approximate p-value.
`ms`	Set this to true for the memory-saving algorithm, which is slower though, but can work with tens of thousands of vectors.

Details

The α-transformation is applied to each composition and then the energy distance of equality of distributions is applied for each value of α or for the single value of α.

Value

A numerical value or a numerical vector, depending on the length of the values of α, with the permutation based p-value(s) of the energy test.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Szekely, G. J. and Rizzo, M. L. (2004) Testing for Equal Distributions in High Dimension. InterStat, November (5).

Szekely, G. J. (2000) Technical Report 03-05: E-statistics: Energy of Statistical Samples. Department of Mathematics and Statistics, Bowling Green State University.

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451

Sevinc V. and Tsagris. M. (2024). Energy Based Equality of Distributions Testing for Compositional Data. https://arxiv.org/pdf/2412.05199

See Also

`acor`, `acor.tune`, `alfa`, `alfa.profile`

Examples

```
y <- rdiri(50, c(3, 4, 5) )
x <- rdiri(60, c(3, 4, 5) )
aeqdist.etest( rbind(x, y), c(dim(x)[1], dim(y)[1]), a = c(-1, 0, 1) )
```

`Energy test of equality of two distributions`

Energy test of equality of two distributions

Description

Energy test of equality of two distributions.

Usage

```
eqdist.etest(x, y, R = 999)
```

Arguments

x	A matrix with the data of the first sample.
y	A matrix with the data of the second sample.
R	The number of permutations to apply in order to compute the approximate p-value.

Details

The energy distance of equality of two distributions is applied. The main advantage of this implementation is that it is light-weight, memory saving, however it works for two distributions only.

Value

The permutation based p-value of the energy test.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Szekely, G. J. and Rizzo, M. L. (2004) Testing for Equal Distributions in High Dimension. InterStat, November (5).

Szekely, G. J. (2000) Technical Report 03-05: E-statistics: Energy of Statistical Samples. Department of Mathematics and Statistics, Bowling Green State University.

See Also

aeqdist.etest, acor, acor.tune, alfa

Examples

```
x <- as.matrix(iris[1:50, 1:4])
y <- as.matrix(iris[51:100, 1:4])
eqdist.etest(x, y)
```

Estimating location and scatter parameters for compositional data

Estimating location and scatter parameters for compositional data

---

Description

Estimating location and scatter parameters for compositional data in a robust and non robust way.

Usage

```
comp.den(x, type = "alr", dist = "normal", tol = 1e-07)
```

Arguments

x	A matrix containing compositional data. No zero values are allowed.
type	A boolean variable indicating the transformation to be used. Either "alr" or "ilr" corresponding to the additive or the isometric log-ratio transformation respectively.
dist	Takes values "normal", "t", "skewnorm", "rob" and "spatial". They first three options correspond to the parameters of the normal, t and skew normal distribution respectively. If it set to "rob" the MCD estimates are computed and if set to "spatial" the spatial median and spatial sign covariance matrix are computed.

`tol` A tolerance level to terminate the process of finding the spatial median when dist = "spatial". This is set to 1e-09 by default.

Details

This function calculates robust and non robust estimates of location and scatter.

Value

A list including: The mean vector and covariance matrix mainly. Other parameters are also returned depending on the value of the argument "dist".

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

P. J. Rousseeuw and K. van Driessen (1999) A fast algorithm for the minimum covariance determinant estimator. Technometrics 41, 212–223.

Mardia K.V., Kent J.T., and Bibby J.M. (1979). Multivariate analysis. Academic press.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

T. Karkkaminen and S. Ayramo (2005). On computation of spatial median for robust data mining. Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems EUROGEN 2005.

A Durre, D Vogel, DE Tyler (2014). The spatial sign covariance matrix with unknown location. Journal of Multivariate Analysis, 130: 107–117.

J. T. Kent, D. E. Tyler and Y. Vardi (1994) A curious likelihood identity for the multivariate t-distribution. Communications in Statistics-Simulation and Computation 23, 441–453.

Azzalini A. and Dalla Valle A. (1996). The multivariate skew-normal distribution. Biometrika 83(4): 715–726.

See Also

`spatmed.reg`, `multivt`

Examples

```
library(MASS)
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
comp.den(x)
comp.den(x, type = "alr", dist = "t")
comp.den(x, type = "alr", dist = "spatial")
```

Estimation of the probability left outside the simplex when using the alpha-transformation

Estimation of the probability left outside the simplex when using the alpha-transformation

Description

Estimation of the probability left outside the simplex when using the alpha-transformationn.

Usage

```
probout(mu, su, a)
```

Arguments

`mu`	The mean vector.
`su`	The covariance matrix.
`a`	The value of α.

Details

When applying the α-transformation based on a multivariate normal there might be probability left outside the simplex as the space of this transformation is a subspace of the Euclidean space. The function estimates the missing probability via Monte Carlo simulation using 40 million generated vectors.

Value

The estimated probability left outside the simplex.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris M. and Stewart C. (2020). A folded model for compositional data analysis. Australian and New Zealand Journal of Statistics, 62(2): 249-277. https://arxiv.org/pdf/1802.07330.pdf

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

`alfa`, `alpha.mle`, `a.est`, `rfolded`

Examples

```
s <-  c(0.1490676523, -0.4580818209,  0.0020395316, -0.0047446076, -0.4580818209,
1.5227259250,  0.0002596411,  0.0074836251,  0.0020395316,  0.0002596411,
0.0365384838, -0.0471448849, -0.0047446076,  0.0074836251, -0.0471448849,
0.0611442781)
s <- matrix(s, ncol = 4)
m <- c(1.715, 0.914, 0.115, 0.167)
probout(m, s, 0.5)
```

`Estimation of the value of alpha in the folded model`

Estimation of the value of α in the folded model

Description

Estimation of the value of α in the folded model.

Usage

```
a.est(x)
```

Arguments

`x` A matrix with the compositional data. No zero vaues are allowed.

Details

This is a function for choosing or estimating the value of α in the folded model (Tsagris and Stewart, 2020).

Value

A list including:

`runtime` The runtime of the algorithm.

`best` The estimated optimal α of the folded model.

`loglik` The maximimised log-likelihood of the folded model.

`p` The estimated probability inside the simplex of the folded model.

`mu` The estimated mean vector of the folded model.

`su` The estimated covariance matrix of the folded model.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris M. and Stewart C. (2022). A Review of Flexible Transformations for Modeling Compositional Data. In Advances and Innovations in Statistics and Data Science, pp. 225–234. https://link.springer.com/chapter/10.10
3-031-08329-7_10

Tsagris M. and Stewart C. (2020). A folded model for compositional data analysis. Australian and New Zealand Journal of Statistics, 62(2): 249-277. https://arxiv.org/pdf/1802.07330.pdf

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

`alfa.profile`, `alfa`, `alfainv`, `alpha.mle`

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
alfa.tune(x)
a.est(x)
```

`Estimation of the value of alpha via the profile log-likelihood`
Estimation of the value of α via the alfa profile log-likelihood

---

Description

Estimation of the value of α via the alfa profile log-likelihood.

Usage

```
alfa.profile(x, a = seq(-1, 1, by = 0.01))
```

Arguments

`x`	A matrix with the compositional data. Zero values are not allowed.
`a`	A grid of values of α.

Details

For every value of α the normal likelihood (see the refernece) is computed. At the end, the plot of the values is constructed.

Value

A list including:

res The chosen value of α, the corresponding log-likelihood value and the log-likelihood when $\alpha = 0$.

ci An asympotic 95% confidence interval computed from the log-likelihood ratio test.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>.

References

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

alfa.tune, alfa, alfainv

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
alfa.tune(x)
alfa.profile(x)
```

Exponential empirical likelihood hypothesis testing for two mean vectors

Exponential empirical likelihood hypothesis testing for two mean vectors

---

Description

Exponential empirical likelihood hypothesis testing for two mean vectors.

Usage

```
eel.test2(y1, y2, tol = 1e-07, R = 0, graph = FALSE)
```

Arguments

`y1` A matrix containing the Euclidean data of the first group.

`y2` A matrix containing the Euclidean data of the second group.

`tol` The tolerance level used to terminate the Newton-Raphson algorithm.

`R` If R is 0, the classical chi-square distribution is used, if R = 1, the corrected chi-square distribution (James, 1954) is used and if R = 2, the modified F distribution (Krishnamoorthy and Yanping, 2006) is used. If R is greater than 3 bootstrap calibration is performed.

`graph` A boolean variable which is taken into consideration only when bootstrap calibration is performed. IF TRUE the histogram of the bootstrap test statistic values is plotted.

Details

Exponential empirical likelihood or exponential tilting was first introduced by Efron (1981) as a way to perform a "tilted" version of the bootstrap for the one sample mean hypothesis testing. Similarly to the empirical likelihood, positive weights p_i, which sum to one, are allocated to the observations, such that the weighted sample mean $\bar{\mathbf{x}}$ is equal to some population mean $\boldsymbol{\mu}$, under the H_0. Under H_1 the weights are equal to $\frac{1}{n}$, where n is the sample size. Following Efron (1981), the choice of p_is will minimize the Kullback-Leibler distance from H_0 to H_1

$$D\left(L_0, L_1\right) = \sum_{i=1}^n p_i \log\left(np_i\right),$$

subject to the constraint $\sum_{i=1}^n p_i \mathbf{x}_i = \boldsymbol{\mu}$. The probabilities take the form

$$p_i = \frac{e^{\boldsymbol{\lambda}^T \mathbf{x}_i}}{\sum_{j=1}^n e^{\boldsymbol{\lambda}^T \mathbf{x}_j}}$$

and the constraint becomes

$$\frac{\sum_{i=1}^n e^{\boldsymbol{\lambda}^T \mathbf{x}_i}\left(\mathbf{x}_i - \boldsymbol{\mu}\right)}{\sum_{j=1}^n e^{\boldsymbol{\lambda}^T \mathbf{x}_j}} = 0 \Rightarrow \frac{\sum_{i=1}^n \mathbf{x}_i e^{\boldsymbol{\lambda}^T \mathbf{x}_i}}{\sum_{j=1}^n e^{\boldsymbol{\lambda}^T \mathbf{x}_j}} - \boldsymbol{\mu} = 0.$$

Similarly to empirical likelihood a numerical search over $\boldsymbol{\lambda}$ is required.

We can derive the asymptotic form of the test statistic in the two sample means case but in a simpler form, generalizing the approach of Jing and Robinson (1997) to the multivariate case as follows. The three constraints are

$$\begin{array}{rcl}
\left(\sum_{j=1}^{n_1} e^{\boldsymbol{\lambda}_1^T \mathbf{x}_j}\right)^{-1} \left(\sum_{i=1}^{n_1} \mathbf{x}_i e^{\boldsymbol{\lambda}_1^T \mathbf{x}_i}\right) - \boldsymbol{\mu} & = & \mathbf{0} \\
\left(\sum_{j=1}^{n_2} e^{\boldsymbol{\lambda}_2^T \mathbf{y}_j}\right)^{-1} \left(\sum_{i=1}^{n_2} \mathbf{y}_i e^{\boldsymbol{\lambda}_2^T \mathbf{y}_i}\right) - \boldsymbol{\mu} & = & \mathbf{0} \\
n_1\boldsymbol{\lambda}_1 + n_2\boldsymbol{\lambda}_2 & = & \mathbf{0}.
\end{array}$$

Similarly to EL the sum of a linear combination of the $\boldsymbol{\lambda}s$ is set to zero. We can equate the first two constraints of

$$\left(\sum_{j=1}^{n_1} e^{\boldsymbol{\lambda}_1^T \mathbf{x}_j}\right)^{-1} \left(\sum_{i=1}^{n_1} \mathbf{x}_i e^{\boldsymbol{\lambda}_1^T \mathbf{x}_i}\right) = \left(\sum_{j=1}^{n_2} e^{\boldsymbol{\lambda}_2^T \mathbf{y}_j}\right)^{-1} \left(\sum_{i=1}^{n_2} \mathbf{y}_i e^{\boldsymbol{\lambda}_2^T \mathbf{y}_i}\right).$$

Also, we can write the third constraint of as $\boldsymbol{\lambda}_2 = -\frac{n_1}{n_2}\boldsymbol{\lambda}_1$ and thus rewrite the first two constraints as

$$\left(\sum_{j=1}^{n_1} e^{\boldsymbol{\lambda}^T\mathbf{x}_j}\right)^{-1}\left(\sum_{i=1}^{n_1}\mathbf{x}_i e^{\boldsymbol{\lambda}^T\mathbf{x}_i}\right) = \left(\sum_{j=1}^{n_2} e^{-\frac{n_1}{n_2}\boldsymbol{\lambda}^T\mathbf{y}_j}\right)^{-1}\left(\sum_{i=1}^{n_2}\mathbf{y}_i e^{-\frac{n_1}{n_2}\boldsymbol{\lambda}^T\mathbf{y}_i}\right).$$

This trick allows us to avoid the estimation of the common mean. It is not possible though to do this in the empirical likelihood method. Instead of minimisation of the sum of the one-sample test statistics from the common mean, we can define the probabilities by searching for the $\boldsymbol{\lambda}$ which makes the last equation hold true. The third constraint of is a convenient constraint, but Jing and Robinson (1997) mention that even though as a constraint is simple it does not lead to second-order accurate confidence intervals unless the two sample sizes are equal. Asymptotically, the test statistic follows a χ^2_d under the null hypothesis.

Value

A list including:

`test` The empirical likelihood test statistic value.

`modif.test` The modified test statistic, either via the chi-square or the F distribution.

`dof` The degrees of freedom of the chi-square or the F distribution.

`pvalue` The asymptotic or the bootstrap p-value.

`mu` The estimated common mean vector.

`runtime` The runtime of the bootstrap calibration.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Efron B. (1981) Nonparametric standard errors and confidence intervals. Canadian Journal of Statistics, 9(2): 139–158.

Jing B.Y. and Wood A.T.A. (1996). Exponential empirical likelihood is not Bartlett correctable. Annals of Statistics, 24(1): 365–369.

Jing B.Y. and Robinson J. (1997). Two-sample nonparametric tilting method. Australian Journal of Statistics, 39(1): 25–34.

Owen A.B. (2001). Empirical likelihood. Chapman and Hall/CRC Press.

Preston S.P. and Wood A.T.A. (2010). Two-Sample Bootstrap Hypothesis Tests for Three-Dimensional Labelled Landmark Data. Scandinavian Journal of Statistics 37(4): 568–587.

Tsagris M., Preston S. and Wood A.T.A. (2017). Nonparametric hypothesis testing for equality of means on the simplex. Journal of Statistical Computation and Simulation, 87(2): 406–422.

See Also

`el.test2`, `maovjames`, `hotel2T2`,`james`

Examples

```
y1 = as.matrix(iris[1:25, 1:4])
y2 = as.matrix(iris[26:50, 1:4])
eel.test2(y1, y2)
eel.test2(y1, y2 )
eel.test2( y1, y2 )
```

`Fast estimation of the value of alpha` *Fast estimation of the value of α*

Description

Fast estimation of the value of α.

Usage

```
alfa.tune(x, B = 1, ncores = 1)
```

Arguments

`x` A matrix with the compositional data. No zero vaues are allowed.

`B` If no (bootstrap based) confidence intervals should be returned this should be 1 and more than 1 otherwise.

`ncores` If ncores is greater than 1 parallel computing is performed. It is advisable to use it if you have many observations and or many variables, otherwise it will slow down th process.

Details

This is a faster function than `alfa.profile` for choosing the value of α.

Value

A vector with the best alpha, the maximised log-likelihood and the log-likelihood at $\alpha = 0$, when B = 1 (no bootstrap). If B>1 a list including:

`param` The best alpha and the value of the log-likelihod, along with the 95% bootstrap based confidence intervals.

`message` A message with some information about the histogram.

`runtime` The time (in seconds) of the process.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Giorgos Athineou <gioathineou@gmail.com>.

References

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

alfa.profile, alfa, alfainv

Examples

```
library(MASS)
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
alfa.tune(x)
alfa.profile(x)
```

Gaussian mixture models for compositional data

Gaussian mixture models for compositional data

Description

Gaussian mixture models for compositional data.

Usage

```
mix.compnorm(x, g, model, type = "alr", veo = FALSE)
```

Arguments

`x` A matrix with the compositional data.

`g` How many clusters to create.

`model` The type of model to be used.

1. "EII": All groups have the same diagonal covariance matrix, with the same variance for all variables.
2. "VII": Different diagonal covariance matrices, with the same variance for all variables within each group.
3. "EEI": All groups have the same diagonal covariance matrix.
4. "VEI": Different diagonal covariance matrices. If we make all covariance matrices have determinant 1, (divide the matrix with the $p$-th root of its determinant) then all covariance matrices will be the same.
5. "EVI": Different diagonal covariance matrices with the same determinant.
6. "VVI": Different diagonal covariance matrices, with nothing in common.
7. "EEE": All covariance matrices are the same.

8. "EEV": Different covariance matrices, but with the same determinant and in addition, if we make them have determinant 1, they will have the same trace.
9. "VEV": Different covariance matrices but if we make the matrices have determinant 1, then they will have the same trace.
10. "VVV": Different covariance matrices with nothing in common.
11. "EVE": Different covariance matrices, but with the same determinant. In addition, calculate the eigenvectors for each covariance matrix and you will see the extra similarities.
12. "VVE": Different covariance matrices, but they have something in common with their directions. Calculate the eigenvectors of each covariance matrix and you will see the similarities.
13. "VEE": Different covariance matrices, but if we make the matrices have determinant 1, then they will have the same trace. In addition, calculate the eigenvectors for each covariance matrix and you will see the extra similarities.
14. "EVV": Different covariance matrices, but with the same determinant.

`type` The type of trasformation to be used, either the additive log-ratio ("alr"), the isometric log-ratio ("ilr") or the pivot coordinate ("pivot") transformation.

`veo` Stands for "Variables exceed observations". If TRUE then if the number variablesin the model exceeds the number of observations, but the model is still fitted.

Details

A log-ratio transformation is applied and then a Gaussian mixture model is constructed.

Value

A list including:

`mu` A matrix where each row corresponds to the mean vector of each cluster.

`su` An array containing the covariance matrix of each cluster.

`prob` The estimated mixing probabilities.

`est` The estimated cluster membership values.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Ryan P. Browne, Aisha ElSherbiny and Paul D. McNicholas (2015). R package mixture: Mixture Models for Clustering and Classification.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`bic.mixcompnorm`, `rmixcomp`, `mix.compnorm.contour`, `alfa.mix.norm`,`alfa.knn`,`alfa.rda`, `comp.nb`

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x/ rowSums(x)
mod1 <- mix.compnorm(x, 3, model = "EII" )
mod2 <- mix.compnorm(x, 4, model = "VII")
```

`Gaussian mixture models for compositional data using the alpha-transformation`

Gaussian mixture models for compositional data using the α-transformation

---

Description

Gaussian mixture models for compositional data using the α-transformation.

Usage

```
alfa.mix.norm(x, g, a, model, veo = FALSE)
```

Arguments

`x` A matrix with the compositional data.

`g` How many clusters to create.

`a` The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.

`model` The type of model to be used.

1. "EII": All groups have the same diagonal covariance matrix, with the same variance for all variables.
2. "VII": Different diagonal covariance matrices, with the same variance for all variables within each group.
3. "EEI": All groups have the same diagonal covariance matrix.
4. "VEI": Different diagonal covariance matrices. If we make all covariance matrices have determinant 1, (divide the matrix with the p-th root of its determinant) then all covariance matrices will be the same.
5. "EVI": Different diagonal covariance matrices with the same determinant.
6. "VVI": Different diagonal covariance matrices, with nothing in common.
7. "EEE": All covariance matrices are the same.

8. "EEV": Different covariance matrices, but with the same determinant and in addition, if we make them have determinant 1, they will have the same trace.
9. "VEV": Different covariance matrices but if we make the matrices have determinant 1, then they will have the same trace.
10. "VVV": Different covariance matrices with nothing in common.
11. "EVE": Different covariance matrices, but with the same determinant. In addition, calculate the eigenvectors for each covariance matrix and you will see the extra similarities.
12. "VVE": Different covariance matrices, but they have something in common with their directions. Calculate the eigenvectors of each covariance matrix and you will see the similarities.
13. "VEE": Different covariance matrices, but if we make the matrices have determinant 1, then they will have the same trace. In addition, calculate the eigenvectors for each covariance matrix and you will see the extra similarities.
14. "EVV": Different covariance matrices, but with the same determinant.

`veo` Stands for "Variables exceed observations". If TRUE then if the number variablesin the model exceeds the number of observations, but the model is still fitted.

Details

A log-ratio transformation is applied and then a Gaussian mixture model is constructed.

Value

A list including:

`mu` A matrix where each row corresponds to the mean vector of each cluster.

`su` An array containing the covariance matrix of each cluster.

`prob` The estimated mixing probabilities.

`est` The estimated cluster membership values.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>`.

References

Ryan P. Browne, Aisha ElSherbiny and Paul D. McNicholas (2015). R package mixture: Mixture Models for Clustering and Classification.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

`bic.alfamixnorm`, `bic.mixcompnorm`, `rmixcomp`, `mix.compnorm.contour`, `mix.compnorm`,`alfa`, `alfa.knn`, `alfa.rda`, `comp.nb`

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x/ rowSums(x)
mod1 <- alfa.mix.norm(x, 3, 0.4, model = "EII" )
mod2 <- alfa.mix.norm(x, 4, 0.7, model = "VII")
```

`Generalised Dirichlet random values simulation`

Generalised Dirichlet random values simulation

Description

Generalised Dirichlet random values simulation.

Usage

```
rgendiri(n, a, b)
```

Arguments

n	The sample size, a numerical value.
a	A numerical vector with the shape parameter values of the Gamma distribution.
b	A numerical vector with the scale parameter values of the Gamma distribution.

Details

The algorithm is straightforward, for each vector, independent gamma values are generated and then divided by their total sum. The difference with `rdiri` is that here the Gamma distributed variables are not equally scaled.

Value

A matrix with the simulated data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Ng Kai Wang, Guo-Liang Tian and Man-Lai Tang (2011). Dirichlet and related distributions: Theory, methods and applications. John Wiley & Sons.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`rdiri`, `diri.est`, `diri.nr`, `diri.contour`

Examples

```
a <- c(1, 2, 3)
b <- c(2, 3, 4)
x <- rgendiri(100, a, b)
```

`Generate random folds for cross-validation`

Generate random folds for cross-validation

Description

Random folds for use in a cross validation are generated. There is the option for stratified splitting as well.

Usage

```
makefolds(ina, nfolds = 10, stratified = TRUE, seed = NULL)
```

Arguments

`ina`	A variable indicating the groupings.
`nfolds`	The number of folds to produce.
`stratified`	A boolean variable specifying whether stratified random (TRUE) or simple random (FALSE) sampling is to be used when producing the folds.
`seed`	You can specify your own seed number here or leave it NULL.

Details

I was inspired by the command in the package **TunePareto** in order to do the stratified version.

Value

A list with nfolds elements where each elements is a fold containing the indices of the data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

compknn.tune

Examples

```
a <- makefolds(iris[, 5], nfolds = 5, stratified = TRUE)
table(iris[a[[1]], 5])  ## 10 values from each group
```

Greenacre's power transformation

Greenacre's power transformation

Description

Greenacre's power transformation.

Usage

```
green(x, theta)
```

Arguments

x	A matrix with the compositional data.
theta	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\theta = 0$ the log transformation is applied.

Details

Greenacre's transformation is applied to the compositional data.

Value

A matrix with the power transformed data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Greenacre, M. (2009). Power transformations in correspondence analysis. Computational Statistics & Data Analysis, 53(8): 3107-3116. http://www.econ.upf.edu/~michael/work/PowerCA.pdf

See Also

alfa

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y1 <- green(x, 0.1)
y2 <- green(x, 0.2)
rbind( colMeans(y1), colMeans(y2) )
```

Helper Frechet mean for compositional data

Helper Frechet mean for compositional data

Description

Helper Frechet mean for compositional data.

Usage

```
frechet2(x, di, a, k)
```

Arguments

x	A matrix with the compositional data.
di	A matrix with indices as produced by the function "dista" of the package "Rfast"" or the function "nn" of the package "Rnanoflann". Better see the details section.
a	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied and the closed geometric mean is calculated.
k	The number of nearest neighbours used for the computation of the Frechet means.

Details

The power transformation is applied to the compositional data and the mean vector is calculated. Then the inverse of it is calculated and the inverse of the power transformation applied to the last vector is the Frechet mean.

What this helper function do is to speed up the Frechet mean when used in the α-k-NN regression. The α-k-NN regression computes the Frechet mean of the k nearest neighbours for a value of α and this function does exactly that. Suppose you want to predict the compositional value of some new predictors. For each predictor value you must use the Frechet mean computed at various nearest neighbours. This function performs these computations in a fast way. It is not the fastest way, yet it is a pretty fast way. This function is being called inside the function aknn.reg.

Value

A list where eqch element contains a matrix. Each matrix contains the Frechet means computed at various nearest neighbours.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

`alfa`, `alfainv`, `profile`

Examples

```
library(MASS)
library(Rfast)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
xnew <- x[1:10, ]
x <- x[-c(1:10), ]
k <- 2:5
di <- Rfast::dista( xnew, x, k = max(k), index = TRUE, square = TRUE )
est <- frechet2(x, di, 0.2, k)
```

Helper functions for the Kullback-Leibler regression

Helper functions for the Kullback-Leibler regression

Description

Helper functions for the Kullback-Leibler regression.

Usage

```
kl.compreg2(y, x, con = TRUE, xnew = NULL, tol = 1e-07, maxiters = 50)
klcompreg.boot(y, x, der, der2, id, b1, n, p, d, tol = 1e-07, maxiters = 50)
```

Arguments

`y`	A matrix with the compositional data (dependent variable). Zero values are allowed. For the klcompreg.boot the first column is removed.
`x`	The predictor variable(s), they can be either continuous or categorical or both. In the klcompreg.boot this is the design matrix.
`con`	If this is TRUE (default) then the constant term is estimated, otherwise the model includes no constant term.
`xnew`	If you have new data use it, otherwise leave it NULL.
`tol`	The tolerance value to terminate the Newton-Raphson procedure.
`maxiters`	The maximum number of Newton-Raphson iterations.
`der`	An vector to put the first derivative there.
`der2`	An empty matrix to put the second derivatives there, the Hessian matrix will be put here.
`id`	A help vector with indices.
`b1`	The matrix with the initial estimated coefficients.
`n`	The sample size
`p`	The number of columns of the design matrix.
`d`	The dimensionality of the simplex, that is the number of columns of the compositional data minus 1.

Details

These are help functions for the `kl.compreg` function. They are not to be called directly by the user.

Value

For kl.compreg2 a list including:

`iters`	The nubmer of iterations required by the Newton-Raphson.
`loglik`	The loglikelihood.
`be`	The beta coefficients.
`est`	The fitted or the predicted values (if xnew is not NULL).

For klcompreg.boot a list including:

`loglik`	The loglikelihood.
`be`	The beta coefficients.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Murteira, Jose MR, and Joaquim JS Ramalho 2016. Regression analysis of multivariate fractional data. Econometric Reviews 35(4): 515-552.

See Also

`diri.reg`, `js.compreg`, `ols.compreg`, `comp.reg`

Examples

```
library(MASS)
x <- as.vector(fgl[, 1])
y <- as.matrix(fgl[, 2:9])
y <- y / rowSums(y)
mod1<- kl.compreg(y, x, B = 1, ncores = 1)
mod2 <- js.compreg(y, x, B = 1, ncores = 1)
```

`Hotelling's multivariate version of the 2 sample t-test for Euclidean data`

Hotelling's multivariate version of the 2 sample t-test for Euclidean data

---

Description

Hotelling's test for testing the equality of two Euclidean population mean vectors.

Usage

```
hotel2T2(x1, x2, a = 0.05, R = 999, graph = FALSE)
```

Arguments

x1 A matrix containing the Euclidean data of the first group.

x2 A matrix containing the Euclidean data of the second group.

a The significance level, set to 0.05 by default.

R If R is 1 no bootstrap calibration is performed and the classical p-value via the F distribution is returned. If R is greater than 1, the bootstrap p-value is returned.

graph A boolean variable which is taken into consideration only when bootstrap calibration is performed. IF TRUE the histogram of the bootstrap test statistic values is plotted.

Details

The fist case scenario is when we assume equality of the two covariance matrices. This is called the two-sample Hotelling's T^2 test (Mardia, Kent and Bibby, 1979, pg. 131-140) and Everitt (2005, pg. 139). The test statistic is defined as

$$T^2 = \frac{n_1 n_2}{n_1 + n_2} \left(\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2\right)^T \mathbf{S}^{-1} \left(\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2\right),$$

where $\mathbf{S}$ is the pooled covariance matrix calculated under the assumption of equal covariance matrices $\mathbf{S} = \frac{(n_1-1)\mathbf{S}_1+(n_2-1)\mathbf{S}_2}{n_1+n_2-2}$. Under H_0 the statistic F given by

$$F = \frac{(n_1 + n_2 - p - 1)\, T^2}{(n_1 + n_2 - 2)\, p}$$

follows the F distribution with p and $n_1 + n_2 - p - 1$ degrees of freedom. Similar to the one-sample test, an extra argument (R) indicates whether bootstrap calibration should be used or not. If R=1, then the asymptotic theory applies, if R>1, then the bootstrap p-value will be applied and the number of re-samples is equal to R. The estimate of the common mean used in the bootstrap to transform the data under the null hypothesis the mean vector of the combined sample, of all the observations.

The built-in command `manova` does the same thing exactly. Try it, the asymptotic F test is what you have to see. In addition, this command allows for more mean vector hypothesis testing for more than two groups. I noticed this command after I had written my function and nevertheless as I mention in the introduction this document has an educational character as well.

Value

A list including:

mesoi The two mean vectors.

info The test statistic, the p-value, the critical value and the degrees of freedom of the F distribution (numerator and denominator). This is given if no bootstrap calibration is employed.

`pvalue` The bootstrap p-value is bootstrap is employed.

`note` A message informing the user that bootstrap calibration has been employed.

`runtime` The runtime of the bootstrap calibration.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Everitt B. (2005). An R and S-Plus Companion to Multivariate Analysis. Springer.

Mardia K.V., Kent J.T. and Bibby J.M. (1979). Multivariate Analysis. London: Academic Press.

Tsagris M., Preston S. and Wood A.T.A. (2017). Nonparametric hypothesis testing for equality of means on the simplex. Journal of Statistical Computation and Simulation, 87(2): 406–422.

See Also

`james`, `el.test2`, `eel.test2`

Examples

```
hotel2T2( as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]) )
hotel2T2( as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]), R = 1 )
```

`Hypothesis testing for two or more compositional mean vectors` *Hypothesis testing for two or more compositional mean vectors*

Description

Hypothesis testing for two or more compositional mean vectors.

Usage

```
comp.test(x, ina, test = "james", R = 0, ncores = 1, graph = FALSE)
```

Arguments

`x` A matrix containing compositional data.

`ina` A numerical or factor variable indicating the groups of the data.

`test` This can take the values of "james" for James’ test, "hotel" for Hotelling’s test, "maov" for multivariate analysis of variance assuming equality of the covariance matrices, "maovjames" for multivariate analysis of variance without assuming equality of the covariance matrices. "el" for empirical likelihood or "eel" for exponential empirical likelihood.

`R` This depends upon the value of the argument "test". If the test is "maov" or "maovjames", R is not taken into consideration. If test is "hotel", then R denotes the number of bootstrap resamples. If test is "james", then R can be 1 (chi-square distribution), 2 (F distribution), or more for bootstrap calibration. If test is "el", then R can be 0 (chi-square), 1 (corrected chi-sqaure), 2 (F distribution) or more for bootstrap calibration. See the help page of each test for more information.

`ncores` How many to cores to use. This is taken into consideration only if test is "el" and R is more than 2.

`graph` A boolean variable which is taken into consideration only when bootstrap calibration is performed. IF TRUE the histogram of the bootstrap test statistic values is plotted. This is taken into account only when R is greater than 2.

Details

The idea is to apply the α-transformation, with $\alpha = 1$, to the compositional data and then use a test to compare their mean vectors. See the help page of each test for more information. The function is visible so you can see exactly what is going on.

Value

A list including:

`result` The outcome of each test.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Giorgos Athineou <gioathineou@gmail.com>.

References

Tsagris M., Preston S. and Wood A.T.A. (2017). Nonparametric hypothesis testing for equality of means on the simplex. Journal of Statistical Computation and Simulation, 87(2): 406–422.

G.S. James (1954). Tests of Linear Hypothese in Univariate and Multivariate Analysis when the Ratios of the Population Variances are Unknown. Biometrika, 41(1/2): 19–43

Krishnamoorthy K. and Yanping Xia (2006). On Selecting Tests for Equality of Two Normal Mean Vectors. Multivariate Behavioral Research 41(4): 533–548.

Owen A. B. (2001). Empirical likelihood. Chapman and Hall/CRC Press.

Owen A.B. (1988). Empirical likelihood ratio confidence intervals for a single functional. Biometrika 75(2): 237–249.

Amaral G.J.A., Dryden I.L. and Wood A.T.A. (2007). Pivotal bootstrap methods for k-sample problems in directional statistics and shape analysis. Journal of the American Statistical Association 102(478): 695–707.

Preston S.P. and Wood A.T.A. (2010). Two-Sample Bootstrap Hypothesis Tests for Three-Dimensional Labelled Landmark Data. Scandinavian Journal of Statistics 37(4): 568–587.

Jing Bing-Yi and Andrew TA Wood (1996). Exponential empirical likelihood is not Bartlett correctable. Annals of Statistics 24(1): 365–369.

See Also

hd.meantest2, dptest

Examples

```
ina <- rep(1:2, each = 50)
x <- as.matrix(iris[1:100, 1:4])
x <- x/ rowSums(x)
comp.test( x, ina, test = "james" )
comp.test( x, ina, test = "hotel" )
comp.test( x, ina, test = "el" )
comp.test( x, ina, test = "eel" )
```

ICE plot for projection pursuit regression with compositional predictor variables

ICE plot for projection pursuit regression with compositional predictor variables

---

Description

ICE plot for projection pursuit regression with compositional predictor variables.

Usage

```
ice.pprcomp(model, x, k = 1, frac = 0.1, type = "log")
```

Arguments

model	The ppr model, the outcome of the pprcomp function.
x	A matrix with the compositional data. No zero values are allowed.
k	Which variable to select?.
frac	Fraction of observations to use. The default value is 0.1.
type	Either "alr" or "log" corresponding to the additive log-ratio transformation or the simple logarithm applied to the compositional data.

Details

This function implements the Individual Conditional Expecation plots of Goldstein et al. (2015). See the references for more details.

Value

A graph with several curves. The horizontal axis contains the selected variable, whereas the vertical axis contains the centered predicted values. The black curves are the effects for each observation and the blue line is their average effect.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

https://christophm.github.io/interpretable-ml-book/ice.html

Goldstein, A., Kapelner, A., Bleich, J. and Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. Journal of Computational and Graphical Statistics 24(1): 44-65.

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression. Journal of the American Statistical Association, 76, 817-823. doi: 10.2307/2287576.

See Also

pprcomp, pprcomp.tune, ice.kernreg, alfa.pcr, lc.reg, comp.ppr

Examples

```
x <- as.matrix( iris[, 2:4] )
x <- x/ rowSums(x)
y <- iris[, 1]
model <- pprcomp(y, x)
ice <- ice.pprcomp(model, x, k = 1)
```

ICE plot for the alpha-k-NN regression

ICE plot for the $\alpha - k - NN$ regression

Description

ICE plot for the $\alpha - k - NN$ regression.

Usage

```
ice.aknnreg(y, x, a, k, apostasi = "euclidean", rann = FALSE,
ind = 1, frac = 0.2, qpos = 0.9)
```

Arguments

y A numerical vector with the response values.

x A numerical matrix with the predictor variables.

a The value α to consider.

k The number of nearest neighbours to consider.

apostasi The type of distance to use, either "euclidean" or "manhattan".

`rann` If you have large scale datasets and want a faster k-NN search, you can use kd-trees implemented in the R package "Rnanoflann". In this case you must set this argument equal to TRUE. Note however, that in this case, the only available distance is by default "euclidean".

`ind` Which variable to select?.

`frac` Fraction of observations to use. The default value is 0.1.

`qpos` A number between 0.8 and 1. This is used to place the legend of the figure better. You can play with it. In the worst case scenario the code is open and you tweak this argument as you prefer.

Details

This function implements the Individual Conditional Expecation plots of Goldstein et al. (2015). See the references for more details.

Value

A graph with several curves, one for each component. The horizontal axis contains the selected variable, whereas the vertical axis contains the locally smoothed predicted compositional lines.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

https://christophm.github.io/interpretable-ml-book/ice.html

Goldstein, A., Kapelner, A., Bleich, J. and Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. Journal of Computational and Graphical Statistics 24(1): 44-65.

See Also

`ice.akernreg`, `ice.pprcomp`

Examples

```
y <- as.matrix( iris[, 2:4] )
x <- iris[, 1]
ice <- ice.aknnreg(y, x, a = 0.6, k = 5, ind = 1)
```

```
ICE plot for the alpha-kernel regression
```

ICE plot for the α-kernel regression

Description

ICE plot for the α-kernel regression.

Usage

```
ice.akernreg(y, x, a, h, type = "gauss", ind = 1, frac = 0.1, qpos = 0.9)
```

Arguments

`y`	A numerical vector with the response values.
`x`	A numerical matrix with the predictor variables.
`a`	The value α to consider.
`h`	The bandwidth value to consider.
`type`	The type of kernel to use, "gauss" or "laplace".
`ind`	Which variable to select?.
`frac`	Fraction of observations to use. The default value is 0.1.
`qpos`	A number between 0.8 and 1. This is used to place the legend of the figure better. You can play with it. In the worst case scenario the code is open and you tweak this argument as you prefer.

Details

This function implements the Individual Conditional Expecation plots of Goldstein et al. (2015). See the references for more details.

Value

A graph with several curves, one for each component. The horizontal axis contains the selected variable, whereas the vertical axis contains the locally smoothed predicted compositional lines.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

https://christophm.github.io/interpretable-ml-book/ice.html

Goldstein, A., Kapelner, A., Bleich, J. and Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. Journal of Computational and Graphical Statistics 24(1): 44-65.

See Also

ice.aknnreg, ice.pprcomp

Examples

```
y <- as.matrix( iris[, 2:4] )
x <- iris[, 1]
ice <- ice.akernreg(y, x, a = 0.6, h = 0.1, ind = 1)
```

ICE plot for univariate kernel regression

ICE plot for univariate kernel regression

Description

ICE plot for univariate kernel regression.

Usage

```
ice.kernreg(y, x, h, type = "gauss", k = 1, frac = 0.1)
```

Arguments

y	A numerical vector with the response values.
x	A numerical matrix with the predictor variables.
h	The bandwidth value to consider.
type	The type of kernel to use, "gauss" or "laplace".
k	Which variable to select?.
frac	Fraction of observations to use. The default value is 0.1.

Details

This function implements the Individual Conditional Expecation plots of Goldstein et al. (2015). See the references for more details.

Value

A graph with several curves. The horizontal axis contains the selected variable, whereas the vertical axis contains the centered predicted values. The black curves are the effects for each observation and the blue line is their average effect.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

https://christophm.github.io/interpretable-ml-book/ice.html

Goldstein, A., Kapelner, A., Bleich, J. and Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. Journal of Computational and Graphical Statistics 24(1): 44-65.

See Also

`ice.pprcomp`, `kernreg.tune`, `alfa.pcr`, `lc.reg`

Examples

```
x <- as.matrix( iris[, 2:4] )
y <- iris[, 1]
ice <- ice.kernreg(y, x, h = 0.1, k = 1)
```

Inverse of the alpha-transformation
Inverse of the α-transformation

Description

The inverse of the α-transformation.

Usage

```
alfainv(x, a, h = TRUE)
```

Arguments

`x`	A matrix with Euclidean data. However, they must lie within the feasible, acceptable space. See references for more information.
`a`	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$, the inverse of the isometric log-ratio transformation is applied.
`h`	If h = TRUE this means that the multiplication with the Helmer sub-matrix will take place. It is set to TRUe by default.

Details

The inverse of the α-transformation is applied to the data. If the data lie outside the α-space, NAs will be returned for some values.

Value

A matrix with the pairwise distances.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>.

References

Tsagris M. and Stewart C. (2022). A Review of Flexible Transformations for Modeling Compositional Data. In Advances and Innovations in Statistics and Data Science, pp. 225–234. https://link.springer.com/chapter/10.10
3-031-08329-7_10

Tsagris M.T., Preston S. and Wood A.T.A. (2016). Improved classification for compositional data using the α-transformation. Journal of Classification 33(2): 243–261. https://arxiv.org/pdf/1506.04976v2.pdf

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

alfa, alfadist

Examples

```
library(MASS)
x <- as.matrix(fgl[1:10, 2:9])
x <- x / rowSums(x)
y <- alfa(x, 0.5)$aff
alfainv(y, 0.5)
```

James multivariate version of the t-test
James multivariate version of the t-test

Description

James test for testing the equality of two population mean vectors without assuming equality of the covariance matrices.

Usage

```
james(y1, y2, a = 0.05, R = 999, graph = FALSE)
```

Arguments

`y1` A matrix containing the Euclidean data of the first group.

`y2` A matrix containing the Euclidean data of the second group.

`a` The significance level, set to 0.05 by default.

`R` If R is 1 no bootstrap calibration is performed and the classical p-value via the F distribution is returned. If R is greater than 1, the bootstrap p-value is returned.

`graph` A boolean variable which is taken into consideration only when bootstrap calibration is performed. If TRUE the histogram of the bootstrap test statistic values is plotted.

Details

Here we show the modified version of the two-sample T^2 test (function `hotel2T2`) in the case where the two covariances matrices cannot be assumed to be equal.

James (1954) proposed a test for linear hypotheses of the population means when the variances (or the covariance matrices) are not known. Its form for two p-dimensional samples is:

$$T_u^2 = \left(\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2\right)^T \tilde{\mathbf{S}}^{-1} \left(\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2\right),$$

where $\tilde{\mathbf{S}} = \tilde{\mathbf{S}_1} + \tilde{\mathbf{S}_2} = \frac{\mathbf{S}_1}{n_1} + \frac{\mathbf{S}_2}{n_2}$.

James (1954) suggested that the test statistic is compared with $2h\left(\alpha\right)$, a corrected χ^2 distribution whose form is

$$2h\left(\alpha\right) = \chi^2\left(A + B\chi^2\right),$$

where $A = 1 + \frac{1}{2p}\sum_{i=1}^2 \frac{\left(tr\tilde{\mathbf{S}}^{-1}\tilde{\mathbf{S}_i}\right)^2}{n_i-1}$ and $B = \frac{1}{p(p+2)}\left[\sum_{i=1}^2 \frac{tr\left(\tilde{\mathbf{S}}^{-1}\tilde{\mathbf{S}_i}\right)^2}{n_i-1} + \frac{1}{2}\sum_{i=1}^2 \frac{\left(\mathrm{tr}\tilde{\mathbf{S}}^{-1}\tilde{\mathbf{S}_i}\right)^2}{n_i-1}\right]$.

If you want to do bootstrap to get the p-value, then you must transform the data under the null hypothesis. The estimate of the common mean is given by Aitchison (1986)

$$\hat{\boldsymbol{\mu}}_c = \left(n_1\mathbf{S}_1^{-1} + n_2\mathbf{S}_2^{-1}\right)^{-1}\left(n_1\mathbf{S}_1^{-1}\bar{\mathbf{X}}_1 + n_2\mathbf{S}_2^{-1}\bar{\mathbf{X}}_2\right) = \left(\tilde{\mathbf{S}}_1^{-1} + \tilde{\mathbf{S}}_2^{-1}\right)^{-1}\left(\tilde{\mathbf{S}}_1^{-1}\bar{\mathbf{X}}_1 + \tilde{\mathbf{S}}_2^{-1}\bar{\mathbf{X}}_2\right).$$

The modified Nel and van der Merwe (1986) test is based on the same quadratic form as that of James (1954) but the distribution used to compare the value of the test statistic is different. It is shown in Krishnamoorthy and Yanping (2006) that $T_u^2 \sim \frac{\nu p}{\nu-p+1}F_{p,\nu-p+1}$ approximately, where $\nu = \frac{p+p^2}{\frac{1}{n_1}\left\{\mathrm{tr}\left[\left(\mathbf{S}_1\tilde{\mathbf{S}}\right)^2\right] + \mathrm{tr}\left[\left(\mathbf{S}_1\tilde{\mathbf{S}}\right)\right]^2\right\} + \frac{1}{n_2}\left\{\mathrm{tr}\left[\left(\mathbf{S}_2\tilde{\mathbf{S}}\right)^2\right] + \mathrm{tr}\left[\left(\mathbf{S}_2\tilde{\mathbf{S}}\right)\right]^2\right\}}$.

The algorithm is taken by Krishnamoorthy and Yu (2004).

Value

A list including:

`note` A message informing the user about the test used.

`mesoi` The two mean vectors.

`info` The test statistic, the p-value, the correction factor and the corrected critical value of the chi-square distribution if the James test has been used or, the test statistic, the p-value, the critical value and the degrees of freedom (numerator and denominator) of the F distribution if the modified James test has been used.

`pvalue` The bootstrap p-value if bootstrap is employed.

`runtime` The runtime of the bootstrap calibration.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

James G.S. (1954). Tests of Linear Hypothese in Univariate and Multivariate Analysis when the Ratios of the Population Variances are Unknown. Biometrika, 41(1/2): 19–43.

Krishnamoorthy K. and Yu J. (2004). Modified Nel and Van der Merwe test for the multivariate Behrens-Fisher problem. Statistics & Probability Letters, 66(2): 161–169.

Krishnamoorthy K. and Yanping Xia (2006). On Selecting Tests for Equality of Two Normal Mean Vectors. Multivariate Behavioral Research, 41(4): 533–548.

Tsagris M., Preston S. and Wood A.T.A. (2017). Nonparametric hypothesis testing for equality of means on the simplex. Journal of Statistical Computation and Simulation, 87(2): 406–422.

See Also

`hotel2T2`, `maovjames`, `el.test2`, `eel.test2`

Examples

```
james( as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]), R = 1 )
james( as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]), R = 2 )
james( as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]) )
```

`Kernel regression with a numerical response vector or matrix`
Kernel regression with a numerical response vector or matrix

Description

Kernel regression (Nadaraya-Watson estimator) with a numerical response vector or matrix.

Usage

```
kern.reg(xnew, y, x, h = seq(0.1, 1, length = 10), type = "gauss" )
```

Arguments

`xnew` A matrix with the new predictor variables whose compositions are to be predicted.

`y` A numerical vector or a matrix with the response value.

`x` A matrix with the available predictor variables.

`h` The bandwidth value(s) to consider.

`type` The type of kernel to use, "gauss" or "laplace".

Details

The Nadaraya-Watson estimator regression is applied.

Value

The fitted values. If a single bandwidth is considered then this is a vector or a matrix, depeding on the nature of the response. If multiple bandwidth values are considered then this is a matrix, if the response is a vector, or a list, if the response is a matrix.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Wand M. P. and Jones M. C. (1994). Kernel smoothing. CRC press.

See Also

kernreg.tune, ice.kernreg, akern.reg, aknn.reg

Examples

```
y <- iris[, 1]
x <- iris[, 2:4]
est <- kern.reg(x, y, x, h = c(0.1, 0.2) )
```

Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distributions

Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distributions

Description

Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distributions.

Usage

```
kl.diri(a, b, type = "KL")
```

Arguments

a	A vector with the parameters of the first Dirichlet distribution.
b	A vector with the parameters of the second Dirichlet distribution.
type	A variable indicating whether the Kullback-Leibler divergence ("KL") or the Bhattacharyya distance ("bhatt") is to be computed.

Details

Note that the order is important in the Kullback-Leibler divergence, since this is asymmetric, but not in the Bhattacharyya distance, since it is a metric.

Value

The value of the Kullback-Leibler divergence or the Bhattacharyya distance.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Ng Kai Wang, Guo-Liang Tian and Man-Lai Tang (2011). Dirichlet and related distributions: Theory, methods and applications. John Wiley & Sons.

See Also

diri.est, diri.nr

Examples

```
library(MASS)
a <- runif(10, 0, 20)
b <- runif(10, 1, 10)
kl.diri(a, b)
kl.diri(b, a)
kl.diri(a, b, type = "bhatt")
kl.diri(b, a, type = "bhatt")
```

LASSO Kullback-Leibler divergence based regression

LASSO Kullback-Leibler divergence based regression

Description

LASSO Kullback-Leibler divergence based regression.

Usage

```
lasso.klcompreg(y, x, alpha = 1, lambda = NULL,
nlambda = 100, type = "grouped", xnew = NULL)
```

Arguments

`y`	A numerical matrix with compositional data. Zero values are allowed.
`x`	A numerical matrix containing the predictor variables.
`alpha`	The elastic net mixing parameter, with $0 \leq \alpha \leq 1$. The penalty is defined as a weighted combination of the ridge and of the Lasso regression. When $\alpha = 1$ LASSO is applied, while $\alpha = 0$ yields the ridge regression.
`lambda`	**This information is copied from the package glmnet.** A user supplied lambda sequence. Typical usage is to have the program compute its own lambda sequence based on nlambda and lambda.min.ratio. Supplying a value of lambda overrides this. WARNING: use with care. Avoid supplying a single value for lambda (for predictions after CV use predict() instead). Supply instead a decreasing sequence of lambda values. glmnet relies on its warms starts for speed, and its often faster to fit a whole path than compute a single fit.
`nlambda`	**This information is copied from the package glmnet.** The number of $lambda$ values, default is 100.
`type`	**This information is copied from the package glmnet.**. If "grouped" then a grouped lasso penalty is used on the multinomial coefficients for a variable. This ensures they are all in our out together. The default in our case is "grouped".
`xnew`	If you have new data use it, otherwise leave it NULL.

Details

The function uses the glmnet package to perform LASSO penalised regression. For more details see the function in that package.

Value

A list including:

`mod` We decided to keep the same list that is returned by glmnet. So, see the function in that package for more information.

`est` If you supply a matrix in the "xnew" argument this will return an array of many matrices with the fitted values, where each matrix corresponds to each value of λ.

Author(s)

Michail Tsagris and Abdulaziz Alenazi.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Abdulaziz Alenazi <`a.alenazi@nbu.edu.sa`>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Alenazi A. A. (2022). f-divergence regression models for compositional data. Pakistan Journal of Statistics and Operation Research, 18(4): 867–882.

Friedman J., Hastie T. and Tibshirani R. (2010) Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, Vol. 33(1), 1–22.

See Also

`lassocoef.plot`, `cv.lasso.klcompreg`, `kl.compreg`, `lasso.compreg`, `ols.compreg`, `alfa.pcr`, `alfa.knn.reg`

Examples

```
y <- as.matrix(iris[, 1:4])
y <- y / rowSums(y)
x <- matrix( rnorm(150 * 30), ncol = 30 )
a <- lasso.klcompreg(y, x)
```

`LASSO log-ratio regression with compositional response`

LASSO log-ratio regression with compositional response

Description

LASSO log-ratio regression with compositional response.

Usage

```
lasso.compreg(y, x, alpha = 1, lambda = NULL,
nlambda = 100, xnew = NULL)
```

Arguments

y
: A numerical matrix with compositional data. Zero values are not allowed as the additive log-ratio transformation (alr) is applied to the compositional response prior to implementing the LASSO algortihm.

x
: A numerical matrix containing the predictor variables.

alpha
: The elastic net mixing parameter, with $0 \leq \alpha \leq 1$. The penalty is defined as a weighted combination of the ridge and of the Lasso regression. When $\alpha = 1$ LASSO is applied, while $\alpha = 0$ yields the ridge regression.

lambda
: **This information is copied from the package glmnet.** A user supplied lambda sequence. Typical usage is to have the program compute its own lambda sequence based on nlambda and lambda.min.ratio. Supplying a value of lambda overrides this. WARNING: use with care. Avoid supplying a single value for lambda (for predictions after CV use predict() instead). Supply instead a decreasing sequence of lambda values. glmnet relies on its warms starts for speed, and its often faster to fit a whole path than compute a single fit.

nlambda
: **This information is copied from the package glmnet.** The number of $lambda$ values, default is 100.

xnew
: If you have new data use it, otherwise leave it NULL.

Details

The function uses the glmnet package to perform LASSO penalised regression. For more details see the function in that package.

Value

A list including:

mod
: We decided to keep the same list that is returned by glmnet. So, see the function in that package for more information.

est
: If you supply a matrix in the "xnew" argument this will return an array of many matrices with the fitted values, where each matrix corresponds to each value of λ.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Friedman, J., Hastie, T. and Tibshirani, R. (2010) Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, Vol. 33(1), 1-22.

See Also

cv.lasso.compreg, lassocoef.plot, lasso.klcompreg, cv.lasso.klcompreg,comp.reg

Examples

```
y <- as.matrix(iris[, 1:4])
y <- y / rowSums(y)
x <- matrix( rnorm(150 * 30), ncol = 30 )
a <- lasso.compreg(y, x)
```

`LASSO with compositional predictors using the alpha-transformation`

LASSO with compositional predictors using the $alpha$-transformation

---

Description

LASSO with compositional predictors using the $alpha$-transformation.

Usage

```
alfa.lasso(y, x, a = seq(-1, 1, by = 0.1), model = "gaussian", lambda = NULL,
xnew = NULL)
```

Arguments

`y`	A numerical vector or a matrix for multinomial logistic regression.
`x`	A numerical matrix containing the predictor variables, compositional data, where zero values are allowed..
`a`	A vector with a grid of values of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
`model`	The type of the regression model, "gaussian", "binomial", "poisson", "multinomial", or "mgaussian".
`lambda`	**This information is copied from the package glmnet.** A user supplied lambda sequence. Typical usage is to have the program compute its own lambda sequence based on nlambda and lambda.min.ratio. Supplying a value of lambda overrides this. WARNING: use with care. Avoid supplying a single value for lambda (for predictions after CV use predict() instead). Supply instead a decreasing sequence of lambda values. glmnet relies on its warms starts for speed, and its often faster to fit a whole path than compute a single fit.
`xnew`	If you have new data use it, otherwise leave it NULL.

Details

The function uses the glmnet package to perform LASSO penalised regression. For more details see the function in that package.

Value

A list including sublists for each value of α:

`mod` We decided to keep the same list that is returned by glmnet. So, see the function in that package for more information.

`est` If you supply a matrix in the "xnew" argument this will return an array of many matrices with the fitted values, where each matrix corresponds to each value of λ.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Friedman, J., Hastie, T. and Tibshirani, R. (2010) Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, Vol. 33(1), 1–22.

See Also

`alfalasso.tune`, `cv.lasso.klcompreg`, `lasso.compreg`, `alfa.knn.reg`

Examples

```
y <- as.matrix(iris[, 1])
x <- rdiri(150, runif(20, 2, 5) )
mod <- alfa.lasso(y, x, a = c(0, 0.5, 1))
```

`Log-contrast GLMs with compositional predictor variables` *Log-contrast GLMS with compositional predictor variables*

Description

Log-contrast GLMs with compositional predictor variables.

Usage

```
lc.glm(y, x, z = NULL, model = "logistic", xnew = NULL, znew = NULL)
```

Arguments

`y`	A numerical vector containing the response variable values. This is either a binary variable or a vector with counts.
`x`	A matrix with the predictor variables, the compositional data. No zero values are allowed.
`z`	A matrix, data.frame, factor or a vector with some other covariate(s).
`model`	For the ulc.glm(), this can be either "logistic" or "poisson".
`xnew`	A matrix containing the new compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.
`znew`	A matrix, data.frame, factor or a vector with the values of some other covariate(s). If you have no new data, leave this NULL as is by default.

Details

The function performs the log-contrast logistic or Poisson regression model. The logarithm of the compositional predictor variables is used (hence no zero values are allowed). The response variable is linked to the log-transformed data with the constraint that the sum of the regression coefficients equals 0. If you want the regression without the zum-to-zero contraints see `ulc.glm`. Extra predictors variables are allowed as well, for instance categorical or continuous.

Value

A list including:

`devi`	The residual deviance of the logistic or Poisson regression model.
`be`	The constrained regression coefficients. Their sum (excluding the constant) equals 0.
`est`	If the arguments "xnew" and znew were given these are the predicted or estimated values, otherwise it is NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Lu J., Shi P. and Li H. (2019). Generalized linear models with linear constraints for microbiome compositional data. Biometrics, 75(1): 235–244.

See Also

`ulc.glm`, `lc.glm2`, `ulc.glm2`, `lcglm.aov`

Examples

```
y <- rbinom(150, 1, 0.5)
x <- rdiri(150, runif(3, 1, 4) )
mod1 <- lc.glm(y, x)
```

Log-contrast logistic or Poisson regression with with multiple compositional predictors

Log-contrast logistic or Poisson regression with with multiple compositional predictors

---

Description

Log-contrast logistic or Poisson regression with with multiple compositional predictors.

Usage

```
lc.glm2(y, x, z = NULL, model = "logistic", xnew = NULL, znew = NULL)
```

Arguments

`y`	A numerical vector containing the response variable values. This is either a binary variable or a vector with counts.
`x`	A matrix with the predictor variables, the compositional data. No zero values are allowed.
`z`	A matrix, data.frame, factor or a vector with some other covariate(s).
`model`	This can be either "logistic" or "poisson".
`xnew`	A matrix containing the new compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.
`znew`	A matrix, data.frame, factor or a vector with the values of some other covariate(s). If you have no new data, leave this NULL as is by default.

Details

The function performs the log-contrast logistic or Poisson regression model. The logarithm of the compositional predictor variables is used (hence no zero values are allowed). The response variable is linked to the log-transformed data with the constraint that the sum of the regression coefficients equals 0. If you want the regression without the zum-to-zero contraints see `ulc.glm2`. Extra predictors variables are allowed as well, for instance categorical or continuous.

Value

A list including:

devi	The residual deviance of the logistic or Poisson regression model.
be	The constrained regression coefficients. Their sum (excluding the constant) equals 0.
est	If the arguments "xnew" and znew were given these are the predicted or estimated values, otherwise it is NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Lu J., Shi P. and Li H. (2019). Generalized linear models with linear constraints for microbiome compositional data. Biometrics, 75(1): 235–244.

See Also

ulc.glm2, ulc.glm, lc.glm

Examples

```
y <- rbinom(150, 1, 0.5)
x <- list()
x1 <- as.matrix(iris[, 2:4])
x1 <- x1 / rowSums(x1)
x[[ 1 ]] <- x1
x[[ 2 ]] <- rdiri(150, runif(4) )
x[[ 3 ]] <- rdiri(150, runif(5) )
mod <- lc.glm2(y, x)
```

Log-contrast quantile regression with compositional predictor variables

Log-contrast quantile regression with compositional predictor variables

---

Description

Log-contrast quantile regression with compositional predictor variables.

Usage

```
lc.rq(y, x, z = NULL, tau, xnew = NULL, znew = NULL)
```

Arguments

y	A numerical vector containing the response variable values.
x	A matrix with the predictor variables, the compositional data. No zero values are allowed.
z	A matrix, data.frame, factor or a vector with some other covariate(s).
tau	The quantile to be estimated, a number between 0 and 1.
xnew	A matrix containing the new compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.
znew	A matrix, data.frame, factor or a vector with the values of some other covariate(s). If you have no new data, leave this NULL as is by default.

Details

The function performs the quantile regression model. The logarithm of the compositional predictor variables is used (hence no zero values are allowed). The response variable is linked to the log-transformed data with the constraint that the sum of the regression coefficients equals 0. If you want the regression without the zum-to-zero contraints see `ulc.rq`. Extra predictor variables are allowed as well, for instance categorical or continuous.

Value

A list including:

mod	The object as returned by the function quantreg::rq(). This is useful for hypothesis testing purposes.
be	The constrained regression coefficients. Their sum (excluding the constant) equals 0.
est	If the arguments "xnew" and znew were given these are the predicted or estimated values, otherwise it is NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Koenker R. W. and Bassett G. W. (1978). Regression Quantiles, Econometrica, 46(1): 33–50.

Koenker R. W. and d'Orey V. (1987). Algorithm AS 229: Computing Regression Quantiles. Applied Statistics, 36(3): 383–393.

See Also

`lc.rq2`, `ulc.rq`

Examples

```
y <- rnorm(150)
x <- rdiri(150, runif(3, 1, 4) )
mod1 <- lc.rq(y, x)
```

`Log-contrast quantile regression with with multiple compositional predictors`

Log-contrast quantile regression with with multiple compositional predictors

---

Description

Log-contrast quantile regression with with multiple compositional predictors.

Usage

```
lc.rq2(y, x, z = NULL, tau = 0.5, xnew = NULL, znew = NULL)
```

Arguments

y	A numerical vector containing the response variable values.
x	A matrix with the predictor variables, the compositional data. No zero values are allowed.
z	A matrix, data.frame, factor or a vector with some other covariate(s).
tau	The quantile to be estimated, a number between 0 and 1.
xnew	A matrix containing the new compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.
znew	A matrix, data.frame, factor or a vector with the values of some other covariate(s). If you have no new data, leave this NULL as is by default.

Details

The function performs the log-contrast quantile regression model. The logarithm of the compositional predictor variables is used (hence no zero values are allowed). The response variable is linked to the log-transformed data with the constraint that the sum of the regression coefficients equals 0. If you want the regression without the zum-to-zero contraints see `ulc.rq2`. Extra predictor variables are allowed as well, for instance categorical or continuous.

Value

A list including:

mod	The object as returned by the function quantreg::rq(). This is useful for hypothesis testing purposes.

`be` The constrained regression coefficients. Their sum (excluding the constant) equals 0.

`est` If the arguments "xnew" and znew were given these are the predicted or estimated values, otherwise it is NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Koenker R. W. and Bassett G. W. (1978). Regression Quantiles, Econometrica, 46(1): 33–50.

Koenker R. W. and d'Orey V. (1987). Algorithm AS 229: Computing Regression Quantiles. Applied Statistics, 36(3): 383–393.

See Also

`lc.rq`, `ulc.rq`

Examples

```
y <- rnorm(150)
x <- list()
x1 <- as.matrix(iris[, 2:4])
x1 <- x1 / rowSums(x1)
x[[ 1 ]] <- x1
x[[ 2 ]] <- rdiri(150, runif(4) )
x[[ 3 ]] <- rdiri(150, runif(5) )
mod <- lc.rq2(y, x)
```

`Log-contrast regression with compositional predictor variables`
Log-contrast regression with compositional predictor variables

---

Description

Log-contrast regression with compositional predictor variables.

Usage

```
lc.reg(y, x, z = NULL, xnew = NULL, znew = NULL)
```

Arguments

`y`	A numerical vector containing the response variable values. This must be a continuous variable.
`x`	A matrix with the predictor variables, the compositional data. No zero values are allowed.
`z`	A matrix, data.frame, factor or a vector with some other covariate(s).
`xnew`	A matrix containing the new compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.
`znew`	A matrix, data.frame, factor or a vector with the values of some other covariate(s). If you have no new data, leave this NULL as is by default.

Details

The function performs the log-contrast regression model as described in Aitchison (2003), pg. 84-85. The logarithm of the compositional predictor variables is used (hence no zero values are allowed). The response variable is linked to the log-transformed data with the constraint that the sum of the regression coefficients equals 0. Hence, we apply constrained least squares, which has a closed form solution. The constrained least squares is described in Chapter 8.2 of Hansen (2019). The idea is to minimise the sum of squares of the residuals under the constraint $R^T\beta = c$, where $c = 0$ in our case. If you want the regression without the zum-to-zero contraints see `ulc.reg`. Extra predictors variables are allowed as well, for instance categorical or continuous.

Value

A list including:

`be`	The constrained regression coefficients. Their sum (excluding the constant) equals 0.
`covbe`	The covariance matrix of the constrained regression coefficients.
`va`	The estimated regression variance.
`residuals`	The vector of residuals.
`est`	If the arguments "xnew" and znew were given these are the predicted or estimated values, otherwise it is NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>`.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Hansen, B. E. (2022). Econometrics. Princeton University Press.

See Also

`ulc.reg`, `lcreg.aov`, `lc.reg2`, `alfa.pcr`, `alfa.knn.reg`

Examples

```
y <- iris[, 1]
x <- as.matrix(iris[, 2:4])
x <- x / rowSums(x)
mod1 <- lc.reg(y, x)
mod2 <- lc.reg(y, x, z = iris[, 5])
```

Log-contrast regression with multiple compositional predictors

Log-contrast regression with multiple compositional predictors

---

Description

Log-contrast regression with multiple compositional predictors.

Usage

```
lc.reg2(y, x, z = NULL, xnew = NULL, znew = NULL)
```

Arguments

y	A numerical vector containing the response variable values. This must be a continuous variable.
x	A list with multiple matrices with the predictor variables, the compositional data. No zero values are allowed.
z	A matrix, data.frame, factor or a vector with some other covariate(s).
xnew	A matrix containing a list with multiple matrices with compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.
znew	A matrix, data.frame, factor or a vector with the values of some other covariate(s). If you have no new data, leave this NULL as is by default.

Details

The function performs the log-contrast regression model as described in Aitchison (2003), pg. 84-85. The logarithm of the compositional predictor variables is used (hence no zero values are allowed). The response variable is linked to the log-transformed data with the constraint that the sum of the regression coefficients for each composition equals 0. Hence, we apply constrained least squares, which has a closed form solution. The constrained least squares is described in Chapter 8.2 of Hansen (2019). The idea is to minimise the sum of squares of the residuals under the constraint $R^T\beta = c$, where $c = 0$ in our case. If you want the regression without the zum-to-zero contraints see ulc.reg2. Extra predictors variables are allowed as well, for instance categorical or continuous. The difference with lc.reg is that instead of one, there are multiple compositions treated as predictor variables.

Value

A list including:

`be`	The constrained regression coefficients. The sum of the sets of coefficients (excluding the constant) corresponding to each predictor composition sums to 0.
`covbe`	If covariance matrix of the constrained regression coefficients.
`va`	The variance of the estimated regression coefficients.
`residuals`	The vector of residuals.
`est`	If the arguments "xnew" and "znew" were given these are the predicted or estimated values, otherwise it is NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Hansen, B. E. (2022). Econometrics. Princeton University Press.

Xiaokang Liu, Xiaomei Cong, Gen Li, Kendra Maas and Kun Chen (2020). Multivariate Log-Contrast Regression with Sub-Compositional Predictors: Testing the Association Between Preterm Infants' Gut Microbiome and Neurobehavioral Outcome.

See Also

`ulc.reg2`, `lc.reg`, `ulc.reg`, `lcreg.aov`, `alfa.pcr`, `alfa.knn.reg`

Examples

```
y <- iris[, 1]
x <- list()
x1 <- as.matrix(iris[, 2:4])
x1 <- x1 / rowSums(x1)
x[[ 1 ]] <- x1
x[[ 2 ]] <- rdiri(150, runif(4) )
x[[ 3 ]] <- rdiri(150, runif(5) )
mod <- lc.reg2(y, x)
be <- mod$be
sum(be[2:4])
sum(be[5:8])
sum(be[9:13])
```

Log-likelihood ratio test for a Dirichlet mean vector

Log-likelihood ratio test for a Dirichlet mean vector

Description

Log-likelihood ratio test for a Dirichlet mean vector.

Usage

```
dirimean.test(x, a)
```

Arguments

`x`	A matrix with the compositional data. No zero values are allowed.
`a`	A compositional mean vector. The concentration parameter is estimated at first. If the elements do not sum to 1, it is assumed that the Dirichlet parameters are supplied.

Details

Log-likelihood ratio test is performed for the hypothesis the given vector of parameters "a" describes the compositional data well.

Value

If there are no zeros in the data, a list including:

`param`	A matrix with the estimated parameters under the null and the alternative hypothesis.
`loglik`	The log-likelihood under the alternative and the null hypothesis.
`info`	The value of the test statistic and its relevant p-value.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Giorgos Athineou <gioathineou@gmail.com>.

References

Ng Kai Wang, Guo-Liang Tian and Man-Lai Tang (2011). Dirichlet and related distributions: Theory, methods and applications. John Wiley & Sons.

See Also

`sym.test`, `diri.nr`, `diri.est`, `rdiri`, `ddiri`

Examples

```
x <- rdiri( 100, c(1, 2, 3) )
dirimean.test(x, c(1, 2, 3) )
dirimean.test( x, c(1, 2, 3)/6 )
```

`Log-likelihood ratio test for a symmetric Dirichlet distribution`

Log-likelihood ratio test for a symmetric Dirichlet distribution

Description

Log-likelihood ratio test for a symmetric Dirichlet distribution.

Usage

```
sym.test(x)
```

Arguments

`x` A matrix with the compositional data. No zero values are allowed.

Details

Log-likelihood ratio test is performed for the hypothesis that all Dirichelt parameters are equal.

Value

A list including:

`est.par` The estimated parameters under the alternative hypothesis.

`one.par` The value of the estimated parameter under the null hypothesis.

`res` The loglikelihood under the alternative and the null hypothesis, the value of the test statistic, its relevant p-value and the associated degrees of freedom, which are actually the dimensionality of the simplex, $D-1$, where D is the number of components.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Ng Kai Wang, Guo-Liang Tian and Man-Lai Tang (2011). Dirichlet and related distributions: Theory, methods and applications. John Wiley & Sons.

See Also

diri.nr, diri.est, rdiri, dirimean.test

Examples

```
x <- rdiri( 100, c(5, 7, 1, 3, 10, 2, 4) )
sym.test(x)
x <- rdiri( 100, c(5, 5, 5, 5, 5) )
sym.test(x)
```

Minimized Kullback-Leibler divergence between Dirichlet and logistic normal

Minimized Kullback-Leibler divergence between Dirichlet and logistic normal

Description

Minimized Kullback-Leibler divergence between Dirichlet and logistic normal distributions.

Usage

```
kl.diri.normal(a)
```

Arguments

a	A vector with the parameters of the Dirichlet parameters.

Details

The function computes the minimized Kullback-Leibler divergence from the Dirichlet distribution to the logistic normal distribution.

Value

The minimized Kullback-Leibler divergence from the Dirichlet distribution to the logistic normal distribution.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data, p. 127. Chapman & Hall.

See Also

diri.nr, diri.contour, rdiri, ddiri, dda, diri.reg

Examples

```
a <- runif(5, 1, 5)
kl.diri.normal(a)
```

`Mixture model selection via BIC`

Mixture model selection via BIC

Description

Mixture model selection via BIC.

Usage

```
bic.mixcompnorm(x, G, type = "alr", veo = FALSE, graph = TRUE)
```

Arguments

`x`	A matrix with compositional data.
`G`	A numeric vector with the number of components, clusters, to be considered, e.g. 1:3.
`type`	The type of trasformation to be used, either the additive log-ratio ("alr"), the isometric log-ratio ("ilr") or the pivot coordinate ("pivot") transformation.
`veo`	Stands for "Variables exceed observations". If TRUE then if the number variablesin the model exceeds the number of observations, but the model is still fitted.
`graph`	A boolean variable, TRUE or FALSE specifying whether a graph should be drawn or not.

Details

The alr or the ilr-transformation is applied to the compositional data first and then mixtures of multivariate Gaussian distributions are fitted. BIC is used to decide on the optimal model and number of components.

Value

A plot with the BIC of the best model for each number of components versus the number of components. A list including:

`mod`	A message informing the user about the best model.
`BIC`	The BIC values for every possible model and number of components.
`optG`	The number of components with the highest BIC.
`optmodel`	The type of model corresponding to the highest BIC.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Ryan P. Browne, Aisha ElSherbiny and Paul D. McNicholas (2018). mixture: Mixture Models for Clustering and Classification. R package version 1.5.

Ryan P. Browne and Paul D. McNicholas (2014). Estimating Common Principal Components in High Dimensions. Advances in Data Analysis and Classification, 8(2), 217-226.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

mix.compnorm, mix.compnorm.contour, rmixcomp, bic.alfamixnorm

Examples

```
x <- as.matrix( iris[, 1:4] )
x <- x/ rowSums(x)
bic.mixcompnorm(x, 1:3, type = "alr", graph = FALSE)
bic.mixcompnorm(x, 1:3, type = "ilr", graph = FALSE)
```

Mixture model selection with the alpha-transformation using BIC

Mixture model selection with the α-transformation using BIC

Description

Mixture model selection with the α-transformation using BIC.

Usage

```
bic.alfamixnorm(x, G, a = seq(-1, 1, by = 0.1), veo = FALSE, graph = TRUE)
```

Arguments

x	A matrix with compositional data.
G	A numeric vector with the number of components, clusters, to be considered, e.g. 1:3.
a	A vector with a grid of values of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.

`veo` Stands for "Variables exceed observations". If TRUE then if the number variablesin the model exceeds the number of observations, but the model is still fitted.

`graph` A boolean variable, TRUE or FALSE specifying whether a graph should be drawn or not.

Details

The α-transformation is applied to the compositional data first and then mixtures of multivariate Gaussian distributions are fitted. BIC is used to decide on the optimal model and number of components.

Value

A list including:

`abic` A list that contains the matrices of all BIC values for all values of α.

`optalpha` The value of α that leads to the highest BIC.

`optG` The number of components with the highest BIC.

`optmodel` The type of model corresponding to the highest BIC.

If graph is set equal to TRUE a plot with the BIC of the best model for each number of components versus the number of components and a list with the results of the Gaussian mixture model for each value of α.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Ryan P. Browne, Aisha ElSherbiny and Paul D. McNicholas (2018). mixture: Mixture Models for Clustering and Classification. R package version 1.5.

Ryan P. Browne and Paul D. McNicholas (2014). Estimating Common Principal Components in High Dimensions. Advances in Data Analysis and Classification, 8(2), 217-226.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

`alfa.mix.norm`, `mix.compnorm`, `mix.compnorm.contour`, `rmixcomp`, `alfa`, `alfa.knn`,`alfa.rda`, `comp.nb`

Examples

```
x <- as.matrix( iris[, 1:4] )
x <- x/ rowSums(x)
bic.alfamixnorm(x, 1:3, a = c(0.4, 0.5, 0.6), graph = FALSE)
```

MLE for the multivariate t distribution

MLE for the multivariate t distribution

Description

MLE of the parameters of a multivariate t distribution.

Usage

```
multivt(y, plot = FALSE)
```

Arguments

`y`	A matrix with continuous data.
`plot`	If plot is TRUE the value of the maximum log-likelihood as a function of the degres of freedom is presented.

Details

The parameters of a multivariate t distribution are estimated. This is used by the functions `comp.den` and `bivt.contour`.

Value

A list including:

`center`	The location estimate.
`scatter`	The scatter matrix estimate.
`df`	The estimated degrees of freedom.
`loglik`	The log-likelihood value.
`mesos`	The classical mean vector.
`covariance`	The classical covariance matrix.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Giorgos Athineou <gioathineou@gmail.com>.

References

Nadarajah, S. and Kotz, S. (2008). Estimation methods for the multivariate t distribution. Acta Applicandae Mathematicae, 102(1):99-118.

See Also

bivt.contour, comp.den

Examples

```
x <- as.matrix(iris[, 1:4])
multivt(x)
```

MLE of distributions defined in the (0, 1) interval

MLE of distributions defined in the (0, 1) interval

Description

MLE of distributions defined in the (0, 1) interval.

Usage

```
beta.est(x, tol = 1e-07)
logitnorm.est(x)
hsecant01.est(x, tol = 1e-07)
kumar.est(x, tol = 1e-07)
unitweibull.est(x, tol = 1e-07, maxiters = 100)
ibeta.est(x, tol = 1e-07)
zilogitnorm.est(x)
```

Arguments

`x`	A numerical vector with proportions, i.e. numbers in (0, 1) (zeros and ones are not allowed).
`tol`	The tolerance level up to which the maximisation stops.
`maxiters`	The maximum number of iterations the Newton-Raphson algorithm will perform.

Details

Maximum likelihood estimation of the parameters of some distributions are performed, some of which use the Newton-Raphson. Some distributions and hence the functions do not accept zeros. "logitnorm.mle" fits the logistic normal, hence no Newton-Raphson is required and the "hypersecant01.mle" use the golden ratio search as is it faster than the Newton-Raphson (less computations). The "zilogitnorm.est" stands for the zero inflated logistic normal distribution. The "ibeta.est" fits the zero or the one inflated beta distribution.

Value

A list including:

`iters` The number of iterations required by the Newton-Raphson.

`loglik` The value of the log-likelihood.

`param` The estimated parameters. In the case of "hypersecant01.est" this is called "theta" as there is only one parameter.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. Journal of Hydrology. 46(1-2): 79-88.

Jones, M.C. (2009). Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. Statistical Methodology. 6(1): 70-81.

You can also check the relevant wikipedia pages.

See Also

`diri.est`

Examples

```
x <- rbeta(1000, 1, 4)
beta.est(x)
ibeta.est(x)

x <- runif(1000)
hsecant01.est(x)
logitnorm.est(x)
ibeta.est(x)

x <- rbeta(1000, 2, 5)
x[sample(1:1000, 50)] <- 0
ibeta.est(x)
```

`MLE of the Dirichlet distribution`

MLE of the a Dirichlet distribution

Description

MLE of the parameters of a Dirichlet distribution.

Usage

```
diri.est(x, type = "mle")
```

Arguments

`x`	A matrix containing compositional data.
`type`	If you want to estimate the parameters use type="mle". If you want to estimate the mean vector along with the precision parameter, the second parametrisation of the Dirichlet, use type="prec".

Details

Maximum likelihood estimation of the parameters of a Dirichlet distribution is performed.

Value

A list including:

`loglik`	The value of the log-likelihood.
`param`	The estimated parameters.
`phi`	The estimated precision parameter, if type = "prec".
`mu`	The estimated mean vector, if type = "prec".
`runtime`	The run time of the maximisation procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Giorgos Athineou <gioathineou@gmail.com>.

References

Ng Kai Wang, Guo-Liang Tian and Man-Lai Tang (2011). Dirichlet and related distributions: Theory, methods and applications. John Wiley & Sons.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

diri.nr, diri.contour, rdiri, ddiri, dda, diri.reg

Examples

```
x <- rdiri( 100, c(5, 7, 1, 3, 10, 2, 4) )
diri.est(x)
diri.est(x, type = "prec")
```

MLE of the Dirichlet distribution via Newton-Rapshon

MLE of the Dirichlet distribution via Newton-Rapshon

Description

MLE of the Dirichlet distribution via Newton-Rapshon.

Usage

```
diri.nr(x, type = 1, tol = 1e-07)
```

Arguments

x	A matrix containing compositional data. Zeros are not allowed.
type	Type can either be 1, so that the Newton-Rapshon is used for the maximisation of the log-likelihood, as Minka (2012) suggested or it can be 1. In the latter case the Newton-Raphson algorithm is implemented involving matrix inversions. In addition an even faster implementation has been implemented (in C++) in the package **Rfast** and is used here.
tol	The tolerance level indicating no further increase in the log-likelihood.

Details

Maximum likelihood estimation of the parameters of a Dirichlet distribution is performed via Newton-Raphson. Initial values suggested by Minka (2003) are used. The estimation is super faster than "diri.est" and the difference becomes really apparent when the sample size and or the dimensions increase. In fact this will work with millions of observations. So in general, I trust this one more than "diri.est".

The only problem I have seen with this method is that if the data are concentrated around a point, say the center of the simplex, it will be hard for this and the previous methods to give estimates of the parameters. In this extremely difficult scenario I would suggest the use of the previous function with the precision parametrization "diri.est(x, type = "prec")". It will be extremely fast and accurate.

Value

A list including:

iter	The number of iterations required. If the argument "type" is set to 2 this is not returned.
loglik	The value of the log-likelihood.
param	The estimated parameters.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Thomas P. Minka (2003). Estimating a Dirichlet distribution. http://research.microsoft.com/en-us/um/people/minka/papers/dirichlet/minka-dirichlet.pdf

See Also

diri.est, diri.contour rdiri, ddiri, dda

Examples

```
x <- rdiri( 100, c(5, 7, 5, 8, 10, 6, 4) )
diri.nr(x)
diri.nr(x, type = 2)
diri.est(x)
```

MLE of the folded model for a given value of alpha

MLE of the folded model for a given value of α

Description

MLE of the folded model for a given value of α.

Usage

```
alpha.mle(x, a)
a.mle(a, x)
```

Arguments

x	A matrix with the compositional data. No zero vaues are allowed.
a	A value of α.

Details

This is a function for choosing or estimating the value of α in the α-folded model (Tsagris and Stewart, 2020). It is called by `a.est`.

Value

If "alpha.mle" is called, a list including:

`iters` The number of iterations the EM algorithm required.

`loglik` The maximimized log-likelihood of the folded model.

`p` The estimated probability inside the simplex of the α-folded model.

`mu` The estimated mean vector of the α-folded model.

`su` The estimated covariance matrix of the α-folded model.

If "a.mle" is called, the log-likelihood is returned only.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris M. and Stewart C. (2022). A Review of Flexible Transformations for Modeling Compositional Data. In Advances and Innovations in Statistics and Data Science, pp. 225–234. https://link.springer.com/chapter/10.10
3-031-08329-7_10

Tsagris M. and Stewart C. (2020). A folded model for compositional data analysis. Australian and New Zealand Journal of Statistics, 62(2): 249-277. https://arxiv.org/pdf/1802.07330.pdf

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

`alfa.profile`, `alfa`, `alfainv`, `a.est`

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
mod <- alfa.tune(x)
mod
alpha.mle(x, mod[1])
```

`MLE of the zero adjusted Dirichlet distribution` *MLE of the zero adjusted Dirichlet distribution*

Description

MLE of the zero adjusted Dirichlet distribution.

Usage

```
zad.est(y)
```

Arguments

`y`	A matrix with the compositional data.

Details

A zero adjusted Dirichlet distribution is being fitted and its parameters are estimated.

Value

A list including:

`loglik`	The value of the log-likelihood.
`phi`	The precision parameter. If covariates are linked with it (function "diri.reg2"), this will be a vector.
`mu`	The mean vector of the distribution.
`runtime`	The time required by the model..

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris M. and Stewart C. (2018). A Dirichlet regression model for compositional data with zeros. Lobachevskii Journal of Mathematics, 39(3): 398–412.

Preprint available from https://arxiv.org/pdf/1410.5011.pdf

See Also

`zadr`, `diri.nr`, `zilogitnorm.est`, `zeroreplace`

Examples

```
y <- as.matrix(iris[, 1:3])
y <- y / rowSums(y)
mod1 <- diri.nr(y)
y[sample(1:450, 15) ] <- 0
mod2 <- zad.est(y)
```

Multivariate analysis of variance (James test)

Multivariate analysis of variance (James test)

Description

Multivariate analysis of variance without assuming equality of the covariance matrices.

Usage

```
maovjames(x, ina, a = 0.05)
```

Arguments

x	A matrix containing Euclidean data.
ina	A numerical or factor variable indicating the groups of the data.
a	The significance level, set to 0.005 by default.

Details

James (1954) also proposed an alternative to MANOVA when the covariance matrices are not assumed equal. The test statistic for k samples is

$$J = \sum_{i=1}^{k} \left(\bar{\mathbf{x}}_i - \bar{\mathbf{X}}\right)^T \mathbf{W}_i \left(\bar{\mathbf{x}}_i - \bar{\mathbf{X}}\right),$$

where $\bar{\mathbf{x}}_i$ and n_i are the sample mean vector and sample size of the i-th sample respectively and $\mathbf{W}_i = \left(\frac{\mathbf{S}_i}{n_i}\right)^{-1}$, where $\mathbf{S}_i$ is the covariance matrix of the i-sample mean vector and $\bar{\mathbf{X}}$ is the estimate of the common mean $\bar{\mathbf{X}} = \left(\sum_{i=1}^k \mathbf{W}_i\right)^{-1} \sum_{i=1}^k \mathbf{W}_i \bar{\mathbf{x}}_i$.

Normally one would compare the test statistic with a $\chi^2_{r,1-\alpha}$, where $r = p\left(k-1\right)$ are the degrees of freedom with k denoting the number of groups and p the dimensionality of the data. There are r constraints (how many univariate means must be equal, so that the null hypothesis, that all the mean vectors are equal, holds true), that is where these degrees of freedom come from. James (1954) compared the test statistic with a corrected χ^2 distribution instead. Let A and B be $A = 1 + \frac{1}{2r} \sum_{i=1}^k \frac{\left[\text{tr}\left(\mathbf{I}_p - \mathbf{W}^{-1}\mathbf{W}_i\right)\right]^2}{n_i - 1}$ and $B = \frac{1}{r(r+2)} \sum_{i=1}^k \left\{ \frac{\text{tr}\left[\left(\mathbf{I}_p - \mathbf{W}^{-1}\mathbf{W}_i\right)^2\right]}{n_i - 1} + \frac{\left[\text{tr}\left(\mathbf{I}_p - \mathbf{W}^{-1}\mathbf{W}_i\right)\right]^2}{2(n_i - 1)} \right\}$.

The corrected quantile of the χ^2 distribution is given as before by $2h\left(\alpha\right) = \chi^2\left(A + B\chi^2\right)$.

Value

A vector with the next 4 elements:

`test`	The test statistic.
`correction`	The value of the correction factor.
`corr.critical`	The corrected critical value of the chi-square distribution.
`p-value`	The p-value of the corrected test statistic.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

James G.S. (1954). Tests of Linear Hypotheses in Univariate and Multivariate Analysis when the Ratios of the Population Variances are Unknown. Biometrika, 41(1/2): 19–43.

See Also

`hotel2T2`, `james`

Examples

```
maovjames( as.matrix(iris[,1:4]), iris[,5] )
```

`Multivariate analysis of variance assuming equality of the covariance matrices`

Multivariate analysis of variance assuming equality of the covariance matrices

Description

Multivariate analysis of variance assuming equality of the covariance matrices.

Usage

```
maov(x, ina)
```

Arguments

`x`	A matrix containing Euclidean data.
`ina`	A numerical or factor variable indicating the groups of the data.

Details

Multivariate analysis of variance assuming equality of the covariance matrices.

Value

A list including:

`note`	A message stating whether the F or the chi^2 approximation has been used.
`result`	The test statistic and the p-value.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Johnson R.A. and Wichern D.W. (2007, 6th Edition). Applied Multivariate Statistical Analysis, pg. 302–303.

Todorov V. and Filzmoser P. (2010). Robust Statistic for the One-way MANOVA. Computational Statistics & Data Analysis, 54(1): 37–48.

See Also

`maovjames`, `hotel2T2`, `james`

Examples

```
maov( as.matrix(iris[,1:4]), iris[,5] )
maovjames( as.matrix(iris[,1:4]), iris[,5] )
```

`Multivariate kernel density estimation`
Multivariate kernel density estimation

Description

Multivariate kernel density estimation.

Usage

```
mkde(x, h = NULL, thumb = "silverman")
```

Arguments

`x`	A matrix with Euclidean (continuous) data.
`h`	The bandwidh value. It can be a single value, which is turned into a vector and then into a diagonal matrix, or a vector which is turned into a diagonal matrix. If you put this NULL then you need to specify the "thumb" argument below.
`thumb`	Do you want to use a rule of thumb for the bandwidth parameter? If no, set h equal to NULL and put "estim" for maximum likelihood cross-validation, "scott" or "silverman" for Scott's and Silverman's rules of thumb respectively.

Details

The multivariate kernel density estimate is calculated with a (not necssarily given) bandwidth value.

Value

A vector with the density estimates calculated for every vector.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>` and Giorgos Athineou <gioathineou@gmail.com>.

References

Arsalane Chouaib Guidoum (2015). Kernel Estimator and Bandwidth Selection for Density and its Derivatives. The kedd R package.

M.P. Wand and M.C. Jones (1995). Kernel smoothing, pages 91-92.

B.W. Silverman (1986). Density estimation for statistics and data analysis, pages 76-78.

See Also

`mkde.tune`, `comp.kerncontour`

Examples

```
mkde( as.matrix(iris[, 1:4]), thumb = "scott" )
mkde( as.matrix(iris[, 1:4]), thumb = "silverman" )
```

Multivariate kernel density estimation for compositional data

Multivariate kernel density estimation for compositional data

Description

Multivariate kernel density estimation for compositional data.

Usage

```
comp.kern(x, type= "alr", h = NULL, thumb = "silverman")
```

Arguments

x	A matrix with Euclidean (continuous) data.
type	The type of trasformation used, either the additive log-ratio ("alr"), the isometric log-ratio ("ilr") or the pivot coordinate ("pivot") transformation.
h	The bandwidh value. It can be a single value, which is turned into a vector and then into a diagonal matrix, or a vector which is turned into a diagonal matrix. If it is NULL, then you need to specify the "thumb" argument below.
thumb	Do you want to use a rule of thumb for the bandwidth parameter? If no, leave the "h" NULL and put "estim" for maximum likelihood cross-validation, "scott" or "silverman" for Scott's and Silverman's rules of thumb respectively.

Details

The multivariate kernel density estimate is calculated with a (not necssarily given) bandwidth value.

Value

A vector with the density estimates calculated for every vector.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Arsalane Chouaib Guidoum (2015). Kernel Estimator and Bandwidth Selection for Density and its Derivatives.

The kedd R package.

M.P. Wand and M.C. Jones (1995). Kernel smoothing, pages 91-92.

B.W. Silverman (1986). Density estimation for statistics and data analysis, pages 76-78.

See Also

comp.kerncontour, mkde

Examples

```
x <- as.matrix(iris[, 1:3])
x <- x / rowSums(x)
f <- comp.kern(x)
```

Multivariate linear regression

Multivariate linear regression

Description

Multivariate linear regression.

Usage

```
multivreg(y, x, plot = TRUE, xnew = NULL)
```

Arguments

y	A matrix with the Eucldidean (continuous) data.
x	A matrix with the predictor variable(s), they have to be continuous.
plot	Should a plot appear or not?
xnew	If you have new data use it, otherwise leave it NULL.

Details

The classical multivariate linear regression model is obtained.

Value

A list including:

suma	A summary as produced by lm, which includes the coefficients, their standard error, t-values, p-values.
r.squared	The value of the R^2 for each univariate regression.
resid.out	A vector with number indicating which vectors are potential residual outliers.
x.leverage	A vector with number indicating which vectors are potential outliers in the predictor variables space.
out	A vector with number indicating which vectors are potential outliers in the residuals and in the predictor variables space.
est	The predicted values if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>.

References

K.V. Mardia, J.T. Kent and J.M. Bibby (1979). Multivariate Analysis. Academic Press.

See Also

diri.reg, js.compreg, kl.compreg, ols.compreg, comp.reg

Examples

```
library(MASS)
x <- as.matrix(iris[, 1:2])
y <- as.matrix(iris[, 3:4])
multivreg(y, x, plot = TRUE)
```

Multivariate normal random values simulation on the simplex

Multivariate normal random values simulation on the simplex

Description

Multivariate normal random values simulation on the simplex.

Usage

```
rcompnorm(n, m, s, type = "alr")
```

Arguments

n	The sample size, a numerical value.
m	The mean vector in R^d.
s	The covariance matrix in R^d.
type	The alr (type = "alr") or the ilr (type = "ilr") is to be used for closing the Euclidean data onto the simplex.

Details

The algorithm is straightforward, generate random values from a multivariate normal distribution in R^d and brings the values to the simplex S^d using the inverse of a log-ratio transformation.

Value

A matrix with the simulated data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

comp.den, rdiri, rcompt, rcompsn

Examples

```
x <- as.matrix(iris[, 1:2])
m <- colMeans(x)
s <- var(x)
y <- rcompnorm(100, m, s)
comp.den(y)
ternary(y)
```

Multivariate or univariate regression with compositional data in the covariates side using the alpha-transformation

Multivariate or univariate regression with compositional data in the covariates side using the α-transformation

---

Description

Multivariate or univariate regression with compositional data in the covariates side using the α-transformation.

Usage

```
alfa.pcr(y, x, a, k, model = "gaussian", xnew = NULL)
```

Arguments

y	A numerical vector containing the response variable values. They can be continuous, binary, discrete (counts). This can also be a vector with discrete values or a factor for the multinomial regression (model = "multinomial").
x	A matrix with the predictor variables, the compositional data.

`a` The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.

`k` How many principal components to use. You may also specify a vector and in this case the results produced will refer to each number of principal components.

`model` The type of regression model to fit. The possible values are "gaussian", "multinomial", "binomial" and "poisson".

`xnew` A matrix containing the new compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.

Details

The α-transformation is applied to the compositional data first ,the first k principal component scores are calcualted and used as predictor variables for a regression model. The family of distributions can be either, "normal" for continuous response and hence normal distribution, "binomial" corresponding to binary response and hence logistic regression or "poisson" for count response and poisson regression.

Value

A list tincluding:

`be` If linear regression was fitted, the regression coefficients of the k principal component scores on the response variable y.

`mod` If another regression model was fitted its outcome as produced in the package **Rfast**.

`per` The percentage of variance explained by the first k principal components.

`vec` The first k principal components, loadings or eigenvectors. These are useful for future prediction in the sense that one needs not fit the whole model again.

`est` If the argument "xnew" was given these are the predicted or estimated values (if xnew is not NULL). If the argument `k` is a vector then this is a matrix with the estimated values for each number of components.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris M. (2015). Regression analysis with compositional data containing zero values. Chilean Journal of Statistics, 6(2): 47-57. https://arxiv.org/pdf/1508.01913v1.pdf

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

glm.pcr, alfapcr.tune

Examples

```
library(MASS)
y <- as.vector(fgl[, 1])
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- alfa.pcr(y = y, x = x, 0.7, 1)
mod
```

Multivariate regression with compositional data

Multivariate regression with compositional data

Description

Multivariate regression with compositional data.

Usage

```
comp.reg(y, x, type = "classical", xnew = NULL, yb = NULL)
```

Arguments

y	A matrix with compsitional data. Zero values are not allowed.
x	The predictor variable(s), they have to be continuous.
type	The type of regression to be used, "classical" for standard multivariate regression, or "spatial" for the robust spatial median regression. Alternatively you can type "lmfit" for the fast classical multivariate regression that does not return standard errors whatsoever.
xnew	This is by default set to NULL. If you have new data whose compositional data values you want to predict, put them here.
yb	If you have already transformed the data using the additive log-ratio transformation, plut it here. Othewrise leave it NULL. This is intended to be used in the function alfareg.tune in order to speed up the process.

Details

The additive log-ratio transformation is applied and then the chosen multivariate regression is implemented. The alr is easier to explain than the ilr and that is why the latter is avoided here.

Value

A list including:

`runtime` The time required by the regression.

`be` The beta coefficients.

`seb` The standard error of the beta coefficients.

`est` The fitted values of xnew if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Giorgos Athineou <gioathineou@gmail.com>.

References

Mardia K.V., Kent J.T., and Bibby J.M. (1979). Multivariate analysis. Academic press.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`multivreg`, `spatmed.reg`, `js.compreg`, `diri.reg`

Examples

```
library(MASS)
y <- as.matrix(iris[, 1:3])
y <- y / rowSums(y)
x <- as.vector(iris[, 4])
mod1 <- comp.reg(y, x)
mod2 <- comp.reg(y, x, type = ”spatial”)
```

`Multivariate skew normal random values simulation on the simplex` *Multivariate skew normal random values simulation on the simplex*

Description

Multivariate skew normal random values simulation on the simplex.

Usage

```
rcompsn(n, xi, Omega, alpha, dp = NULL, type = ”alr”)
```

Arguments

n	The sample size, a numerical value.
xi	A numeric vector of length d representing the location parameter of the distribution.
Omega	A $d \times d$ symmetric positive-definite matrix of dimension.
alpha	A numeric vector which regulates the slant of the density.
dp	A list with three elements, corresponding to xi, Omega and alpha described above. The default value is FALSE. If dp is assigned, individual parameters must not be specified.
type	The alr (type = "alr") or the ilr (type = "ilr") is to be used for closing the Euclidean data onto the simplex.

Details

The algorithm is straightforward, generate random values from a multivariate t distribution in R^d and brings the values to the simplex S^d using the inverse of a log-ratio transformation.

Value

A matrix with the simulated data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Azzalini, A. and Dalla Valle, A. (1996). The multivariate skew-normal distribution. Biometrika, 83(4): 715–726.

Azzalini, A. and Capitanio, A. (1999). Statistical applications of the multivariate skew normal distribution. Journal of the Royal Statistical Society Series B, 61(3):579-602. Full-length version available from http://arXiv.org/abs/0911.2093

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

comp.den, rdiri, rcompnorm

Examples

```
x <- as.matrix(iris[, 1:2])
par <- sn::msn.mle(y = x)$dp
y <- rcompsn(100, dp = par)
comp.den(y, dist = "skewnorm")
ternary(y)
```

Multivariate t random values simulation on the simplex

Multivariate t random values simulation on the simplex

Description

Multivariate t random values simulation on the simplex.

Usage

```
rcompt(n, m, s, dof, type = "alr")
```

Arguments

n	The sample size, a numerical value.
m	The mean vector in R^d.
s	The covariance matrix in R^d.
dof	The degrees of freedom.
type	The alr (type = "alr") or the ilr (type = "ilr") is to be used for closing the Euclidean data onto the simplex.

Details

The algorithm is straightforward, generate random values from a multivariate t distribution in R^d and brings the values to the simplex S^d using the inverse of a log-ratio transformation.

Value

A matrix with the simulated data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

comp.den, rdiri, rcompnorm

Examples

```
x <- as.matrix(iris[, 1:2])
m <- Rfast::colmeans(x)
s <- var(x)
y <- rcompt(100, m, s, 10)
comp.den(y, dist = "t")
ternary(y)
```

`Naive Bayes classifiers for compositional data`

Naive Bayes classifiers for compositional data

Description

Naive Bayes classifiers for compositional data.

Usage

```
comp.nb(xnew = NULL, x, ina, type = "beta")
```

Arguments

`xnew`	A matrix with the new compositional predictor data whose class you want to predict. Zeros are not allowed
`x`	A matrix with the available compositional predictor data. Zeros are not allowed
`ina`	A vector of data. The response variable, which is categorical (factor is acceptable).
`type`	The type of naive Bayes, "beta", "logitnorm", "cauchy", "laplace", "gamma", "normlog" or "weibull". For the last 4 distributions, the negative of the logarithm of the compositional data is applied first.

Value

Depending on the classifier a list including (the ni and est are common for all classifiers):

`shape`	A matrix with the shape parameters.
`scale`	A matrix with the scale parameters.
`expmu`	A matrix with the mean parameters.
`sigma`	A matrix with the (MLE, hence biased) variance parameters.
`location`	A matrix with the location parameters (medians).
`scale`	A matrix with the scale parameters.
`mean`	A matrix with the scale parameters.
`var`	A matrix with the variance parameters.
`a`	A matrix with the "alpha" parameters.

b	A matrix with the "beta" parameters.
ni	The sample size of each group in the dataset.
est	The estimated group of the xnew observations. It returns a numerical value back regardless of the target variable being numerical as well or factor. Hence, it is suggested that you do \"as.numeric(ina)\" in order to see what is the predicted class of the new data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Friedman J., Hastie T. and Tibshirani R. (2017). The elements of statistical learning. New York: Springer.

See Also

cv.compnb, alfa.rda, alfa.knn, comp.knn, mix.compnorm, dda

Examples

```
x <- Compositional::rdiri(100, runif(5) )
ina <- rbinom(100, 1, 0.5) + 1
a <- comp.nb(x, x, ina, type = "beta")
```

Naive Bayes classifiers for compositional data using the alpha-transformation

Naive Bayes classifiers for compositional data using the α-transformation

---

Description

Naive Bayes classifiers for compositional data using the α-transformation.

Usage

```
alfa.nb(xnew, x, ina, a, type = "gaussian")
```

Arguments

`xnew`	A matrix with the new compositional predictor data whose class you want to predict. Zeros are allowed.
`x`	A matrix with the available compositional predictor data. Zeros are allowed.
`ina`	A vector of data. The response variable, which is categorical (factor is acceptable).
`a`	This can be a vector of values or a single number.
`type`	The type of naive Bayes, "gaussian", "cauchy" or "laplace".

Details

The α-transformation is applied to the compositional and a naive Bayes classifier is employed.

Value

A matrix with the estimated groups. One column for each value of α.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

Friedman J., Hastie T. and Tibshirani R. (2017). The elements of statistical learning. New York: Springer.

See Also

`comp.nb`, `alfa.rda`, `alfa.knn`, `comp.knn`, `mix.compnorm`

Examples

```
x <- Compositional::rdiri(100, runif(5) )
ina <- rbinom(100, 1, 0.5) + 1
mod <- alfa.nb(x, x, a = c(0, 0.1, 0.2), ina )
```

Non linear least squares regression for compositional data
Non linear least squares regression for compositional data

Description

Non linear least squares regression for compositional data.

Usage

```
ols.compreg(y, x, con = TRUE, B = 1, ncores = 1, xnew = NULL)
```

Arguments

y	A matrix with the compositional data (dependent variable). Zero values are allowed.
x	A matrix or a data frame with the predictor variable(s).
con	If this is TRUE (default) then the constant term is estimated, otherwise the model includes no constant term.
B	If B is greater than 1 bootstrap estimates of the standard error are returned. If B=1, no standard errors are returned.
ncores	If ncores is 2 or more parallel computing is performed. This is to be used for the case of bootstrap. If B=1, this is not taken into consideration.
xnew	If you have new data use it, otherwise leave it NULL.

Details

The ordinary least squares between the observed and the fitted compositional data is adopted as the objective function. This involves numerical optimization since the relationship is non linear. There is no log-likelihood.

Value

A list including:

runtime	The time required by the regression.
beta	The beta coefficients.
covbe	The covariance matrix of the beta coefficients. If B=1, this is based on the observed information (Hessian matrix), otherwise if B> this is the bootstrap estimate.
est	The fitted of xnew if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>`.

References

Murteira, Jose MR, and Joaquim JS Ramalho 2016. Regression analysis of multivariate fractional data. Econometric Reviews 35(4): 515-552.

See Also

diri.reg, js.compreg, kl.compreg, comp.reg, comp.reg, alfa.reg

Examples

```
library(MASS)
x <- as.vector(fgl[, 1])
y <- as.matrix(fgl[, 2:9])
y <- y / rowSums(y)
mod1 <- ols.compreg(y, x, B = 1, ncores = 1)
mod2 <- js.compreg(y, x, B = 1, ncores = 1)
```

Non-parametric zero replacement strategies
Non-parametric zero replacement strategies

Description

Non-parametric zero replacement strategies.

Usage

```
zeroreplace(x, a = 0.65, delta = NULL, type = "multiplicative")
```

Arguments

x	A matrix with the compositional data.
a	The replacement value (δ) will be "a" times the minimum value observed in the compositional data.
delta	Unless you specify the replacement value δ here.
type	This can be any of "multiplicative", "additive" or "simple". See the references for more details.

Details

The "additive" is the zero replacement strategy suggested in Aitchison (1986, pg. 269). All of the three strategies can be found in Martin-Fernandez et al. (2003).

Value

A matrix with the zero replaced compositional data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Martin-Fernandez J. A., Barcelo-Vidal C. & Pawlowsky-Glahn, V. (2003). Dealing with zeros and missing values in compositional data sets using nonparametric imputation. Mathematical Geology, 35(3): 253-278.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

perturbation, alfa

Examples

```
x <- as.matrix(iris[1:20, 1:4])
x <- x/ rowSums(x)
x[ sample(1:20, 4),  sample(1:4, 1) ] <- 0
x <- x / rowSums(x)
zeroreplace(x)
```

Permutation linear independence test in the SCLS model

Permutation linear independence test in the SCLS model

Description

Permutation linear independence test in the SCLS model.

Usage

```
scls.indeptest(y, x, R = 999)
```

Arguments

y	A matrix with the compositional data (dependent variable). Zero values are allowed.
x	A matrix with the compositional predictors. Zero values are allowed.
R	The number of permutations to perform.

Details

Permutation independence test in the constrained linear least squares for compositional responses and predictors is performed. The observed test statistic is the MSE computed by scls. Then, the rows of X are permuted B times and each time the constrained OLS is performed and the MSE is computed. The p-value is then computed in the usual way.

Value

The p-value for the test of independence between Y and X.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

`scls`, `scls2`, `tflr`, `scls.betest`

Examples

```
library(MASS)
set.seed(1234)
y <- rdiri(214, runif(4, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
scls.indeptest(y, x, R = 99)
```

`Permutation linear independence test in the TFLR model` *Permutation linear independence test in the TFLR model*

Description

Permutation linear independence test in the TFLR model.

Usage

```
tflr.indeptest(y, x, R = 999, ncores = 1)
```

Arguments

`y`	A matrix with the compositional data (dependent variable). Zero values are allowed.
`x`	A matrix with the compositional predictors. Zero values are in general allowed, but there can be cases when these are problematic.
`R`	The number of permutations to perform.
`ncores`	The number of cores to use in case you are interested for parallel computations.

Details

Permutation independence test in the constrained linear least squares for compositional responses and predictors is performed. The observed test statistic is the Kullback-Leibler divergence computed by `tflr`. Then, the rows of X are permuted B times and each time the TFLR is performed and the Kullback-Leibler is computed. The p-value is then computed in the usual way.

Value

The p-value for the test of linear independence between the simplicial response Y and the simplicial predictor X.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Fiksel J., Zeger S. and Datta A. (2022). A transformation-free linear regression for compositional outcomes and predictors. Biometrics, 78(3): 974–987.

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

`tflr`, `scls`, `tflr.betest`

Examples

```
library(MASS)
set.seed(1234)
y <- rdiri(214, runif(4, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
tflr.indeptest(y, x, R = 9)
```

`Permutation test for the matrix of coefficients in the SCLS model` *Permutation test for the matrix of coefficients in the SCLS model*

Description

Permutation test for the matrix of coefficients in the SCLS model.

Usage

```
scls.betest(y, x, B, R = 999)
```

Arguments

`y`	A matrix with the compositional data (dependent variable). Zero values are allowed.
`x`	A matrix with the compositional predictors. Zero values are allowed.
`B`	A specific matrix of coefficients to test. Under the null hypothesis, the matrix of coefficients is equal to this matrix.
`R`	The number of permutations to perform.

Details

Permutation independence test in the constrained linear least squares for compositional responses and predictors is performed. The observed test statistic is the MSE computed by `scls`. Then, the rows of X are permuted B times and each time the constrained OLS is performed and the MSE is computed. The p-value is then computed in the usual way.

Value

The p-value for the test of independence between Y and X.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

`scls`, `scls2`, `tflr`, `scls.indeptest`,`tflr.indeptest`

Examples

```
y <- rdiri(100, runif(3, 1, 3) )
x <- rdiri(100, runif(3, 1, 3) )
B <- diag(3)
scls.betest(y, x, B = B, R = 99)
```

`Permutation test for the matrix of coefficients in the TFLR model` *Permutation test for the matrix of coefficients in the TFLR model*

Description

Permutation test for the matrix of coefficients in the TFLR model.

Usage

```
tflr.betest(y, x, B, R = 999, ncores = 1)
```

Arguments

`y`	A matrix with the compositional data (dependent variable). Zero values are allowed.
`x`	A matrix with the compositional predictors. Zero values are in general allowed, but there can be cases when these are problematic.
`B`	A specific matrix of coefficients to test. Under the null hypothesis, the matrix of coefficients is equal to this matrix.
`R`	The number of permutations to perform.
`ncores`	The number of cores to use in case you are interested for parallel computations.

Details

Permutation independence test in the constrained linear least squares for compositional responses and predictors is performed. The observed test statistic is the Kullback-Leibler divergence computed by `tflr`. Then, the rows of X are permuted B times and each time the TFLR is performed and the Kullback-Leibler is computed. The p-value is then computed in the usual way.

Value

The p-value for the test of linear independence between the simplicial response Y and the simplicial predictor X.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Fiksel J., Zeger S. and Datta A. (2022). A transformation-free linear regression for compositional outcomes and predictors. Biometrics, 78(3): 974–987.

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).s

See Also

tflr, tflr.indeptest, scls, scls.indeptest

Examples

```
y <- rdiri(100, runif(3, 1, 3) )
x <- rdiri(100, runif(3, 1, 3) )
B <- diag(3)
tflr.betest(y, x, B = B, R = 99)
```

Perturbation operation

Perturbation operation

Description

Perturbation operation.

Usage

```
perturbation(x, y, oper = "+")
```

Arguments

x	A matrix with the compositional data.
y	Either a matrix with compositional data or a vector with compositional data. In either case, the data may not be compositional data, as long as they non negative.
oper	For the summation this must be "*" and for the negation it must be "/". According to Aitchison (1986), multiplication is equal to summation in the log-space, and division is equal to negation.

Details

This is the perturbation operation defined by Aitchison (1986).

Value

A matrix with the perturbed compositional data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

power

Examples

```
x <- as.matrix(iris[1:15, 1:4])
y <- as.matrix(iris[21:35, 1:4])
perturbation(x, y)
perturbation(x, y[1, ])
```

Plot of the LASSO coefficients

Plot of the LASSO coefficients

Description

Plot of the LASSO coefficients.

Usage

```
lassocoef.plot(lasso, lambda = TRUE)
```

Arguments

lasso	An object where you have saved the result of the LASSO regression. See the examples for more details.
lambda	If you want the x-axis to contain the logarithm of the penalty parameter $\log(\lambda)$ set this to TRUE. Otherwise the x-axis will contain the L_1-norm of the coefficients.

Details

This function plots the L_2-norm of the coefficients of each predictor variable versus the $\log(\lambda)$ or the L_1-norm of the coefficients. This is the same plot as the one produced by the glmnet package with type.coef = "2norm".

Value

A plot of the L_2-norm of the coefficients of each predictor variable (y-axis) versus the L_1-norm of all the coefficients (x-axis).

Author(s)

Michail Tsagris and Abdulaziz Alenazi.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Abdulaziz Alenazi <a.alenazi@nbu.edu.sa>. <a.alenazi@nbu.edu.sa>.

References

Alenazi, A. A. (2022). f-divergence regression models for compositional data. Pakistan Journal of Statistics and Operation Research, 18(4): 867–882.

Friedman, J., Hastie, T. and Tibshirani, R. (2010) Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, Vol. 33(1), 1–22.

See Also

`lasso.klcompreg`, `cv.lasso.klcompreg`, `lasso.compreg`, `cv.lasso.compreg`,`kl.compreg`, `comp.reg`

Examples

```
y <- as.matrix(iris[, 1:4])
y <- y / rowSums(y)
x <- matrix( rnorm(150 * 30), ncol = 30 )
a <- lasso.klcompreg(y, x)
lassocoef.plot(a)
b <- lasso.compreg(y, x)
lassocoef.plot(b)
```

`Power operation` *Power operation*

Description

Power operation.

Usage

```
pow(x, a)
```

Arguments

`x`	A matrix with the compositional data.
`a`	Either a vector with numbers of a single number.

Details

This is the power operation defined by Aitchison (1986). It is also the starting point of the α-transformation.

Value

A matrix with the power transformed compositional data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. http://arxiv.org/pdf/1106.1451.pdf

See Also

perturbation, alfa

Examples

```
x <- as.matrix(iris[1:15, 1:4])
a <- runif(1)
pow(x, a)
```

Principal component analysis

Principal component analysis

Description

Principal component analysis.

Usage

```
logpca(x, center = TRUE, scale = TRUE, k = NULL, vectors = FALSE)
```

Arguments

x	A matrix with the compositional data. Zero values are not allowed.
center	Do you want your data centered? TRUE or FALSE.
scale	Do you want each of your variables scaled, i.e. to have unit variance? TRUE or FALSE.
k	If you want a specific number of eigenvalues and eigenvectors set it here, otherwise all eigenvalues (and eigenvectors if requested) will be returned.
vectors	Do you want the eigenvectors be returned? By dafault this is FALSE.

Details

The logarithm is applied to the compositional data and PCA is performed.

Value

A list including:

`values`	The eigenvalues.
`vectors`	The eigenvectors.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`alfa.pca`, `alfa.pcr`, `kl.alfapcr`

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x/ rowSums(x)
a <- logpca(x)
```

`Principal component analysis using the alpha-transformation`

Principal component analysis using the α-transformation

Description

Principal component analysis using the α-transformation.

Usage

```
alfa.pca(x, a, center = TRUE, scale = TRUE, k = NULL, vectors = FALSE)
```

Arguments

`x`	A matrix with the compositional data. Zero values are allowed. In that case "a" should be positive.
`a`	The value of α to use in the α-transformation.
`center`	Do you want your data centered? TRUE or FALSE.
`scale`	Do you want each of your variables scaled, i.e. to have unit variance? TRUE or FALSE.
`k`	If you want a specific number of eigenvalues and eigenvectors set it here, otherwise all eigenvalues (and eigenvectors if requested) will be returned.
`vectors`	Do you want the eigenvectors be returned? By dafault this is FALSE.

Details

The α-transformation is applied to the compositional data and then PCA is performed. Note however, that the right multiplication by the Helmert sub-matrix is not applied in order to be in accordance with Aitchison (1983). When $\alpha = 0$, this results to the PCA proposed by Aitchison (1983).

Value

A list including:

`values`	The eigenvalues.
`vectors`	The eigenvectors.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Aitchison, J. (1983). Principal component analysis of compositional data. Biometrika, 70(1), 57–65.

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. http://arxiv.org/pdf/1106.1451.pdf

See Also

`logpca`, `alfa.pcr`, `kl.alfapcr`

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x/ rowSums(x)
a <- alfa.pca(x, 0.5)
```

Principal component generalised linear models

Principal component generalised linear models

Description

Principal component generalised linear models.

Usage

```
glm.pcr(y, x, k = 1, xnew = NULL)
```

Arguments

`y`	A numerical vector with 0 and 1 (binary) or a vector with discrete (count) data.
`x`	A matrix with the predictor variable(s), they have to be continuous.
`k`	A number greater than or equal to 1. How many principal components to use. You may get results for the sequence of principal components.
`xnew`	If you have new data use it, otherwise leave it NULL.

Details

Principal component regression is performed with binary logistic or Poisson regression, depending on the nature of the response variable. The principal components of the cross product of the independent variables are obtained and classical regression is performed. This is used in the function `alfa.pcr`.

Value

A list including:

`model`	The summary of the logistic or Poisson regression model as returned by the package Rfast.
`per`	The percentage of variance of the predictor variables retained by the k principal components.
`vec`	The principal components, the loadings.
`est`	The fitted or the predicted values (if xnew is not NULL). If the argument k is a vector then this is a matrix with the estimated values for each number of components.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aguilera A.M., Escabias M. and Valderrama M.J. (2006). Using principal components for estimating logistic regression with high-dimensional multicollinear data. Computational Statistics & Data Analysis 50(8): 1905-1924.

Jolliffe I.T. (2002). Principal Component Analysis.

See Also

`alfa.pcr`, `alfapcr.tune`

Examples

```
x <- as.matrix(iris[, 1:4])
y <- rbinom(150, 1, 0.6)
mod <- glm.pcr(y, x, k = 1)
```

`Principal coordinate analysis using the alpha-distance`

Principal coordinate analysis using the α-distance

Description

Principal coordinate analysis using the α-distance.

Usage

```
alfa.mds(x, a, k = 2, eig = TRUE)
```

Arguments

`x`	A matrix with the compositional data. Zero values are allowed.
`a`	The value of a. In case of zero values in the data it has to be greater than 1.
`k`	The maximum dimension of the space which the data are to be represented in. This can be a number between 1 and $D - 1$, where D denotes the number of dimensions.
`eig`	Should eigenvalues be returned? The default value is TRUE.

Details

The function computes the α-distance matrix and then plugs it into the classical multidimensional scaling function in the "cmdscale" function.

Value

A list with the results of "cmdscale" function.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Cox, T. F. and Cox, M. A. A. (2001). Multidimensional Scaling. Second edition. Chapman and Hall.

Mardia, K. V., Kent, J. T. and Bibby, J. M. (1979). Chapter 14 of Multivariate Analysis, London: Academic Press.

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

esov.mds, alfa.pca,

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x/ rowSums(x)
a <- esov.mds(x)
```

Principal coordinate analysis using the Jensen-Shannon divergence
Principal coordinate analysis using the Jensen-Shannon divergence

---

Description

Principal coordinate analysis using the Jensen-Shannon divergence.

Usage

```
esov.mds(x, k = 2, eig = TRUE)
```

Arguments

x A matrix with the compositional data. Zero values are allowed.

k The maximum dimension of the space which the data are to be represented in. This can be a number between 1 and $D-1$, where D denotes the number of dimensions.

eig Should eigenvalues be returned? The default value is TRUE.

Details

The function computes the Jensen-Shannon divergence matrix and then plugs it into the classical multidimensional scaling function in the "cmdscale" function.

Value

A list with the results of "cmdscale" function.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Cox, T. F. and Cox, M. A. A. (2001). Multidimensional Scaling. Second edition. Chapman and Hall.

Mardia, K. V., Kent, J. T. and Bibby, J. M. (1979). Chapter 14 of Multivariate Analysis, London: Academic Press.

Tsagris, Michail (2015). A novel, divergence based, regression for compositional data. Proceedings of the 28th Panhellenic Statistics Conference, 15-18/4/2015, Athens, Greece. https://arxiv.org/pdf/1511.07600.pdf

See Also

`alfa.mds`, `alfa.pca`,

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x/ rowSums(x)
a <- esov.mds(x)
```

`Projection pursuit regression for compositional data`

Projection pursuit regression for compositional data

Description

Projection pursuit regression for compositional data.

Usage

```
comp.ppr(y, x, nterms = 3, type = "alr", xnew = NULL, yb = NULL )
```

Arguments

`y`	A matrix with the compositional data.
`x`	A matrix with the continuous predictor variables or a data frame including categorical predictor variables.
`nterms`	The number of terms to include in the final model.
`type`	Either "alr" or "ilr" corresponding to the additive or the isometric log-ratio transformation respectively.
`xnew`	If you have new data use it, otherwise leave it NULL.
`yb`	If you have already transformed the data using a log-ratio transformation put it here. Othewrise leave it NULL.

Details

This is the standard projection pursuit. See the built-in function "ppr" for more details.

Value

A list includign:

runtime The runtime of the regression.

mod The produced model as returned by the function "ppr".

est The fitted values of xnew if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression. Journal of the American Statistical Association, 76, 817-823. doi: 10.2307/2287576.

See Also

compppr.tune, aknn.reg, akern.reg, comp.reg, kl.compreg, alfa.reg

Examples

```
y <- as.matrix(iris[, 1:3])
y <- y/ rowSums(y)
x <- iris[, 4]
mod <- comp.ppr(y, x)
```

Projection pursuit regression with compositional predictor variables

Projection pursuit regression with compositional predictor variables

---

Description

Projection pursuit regression with compositional predictor variables.

Usage

```
pprcomp(y, x, nterms = 3, type = "log", xnew = NULL)
```

Arguments

y A numerical vector with the continuous variable.

x A matrix with the compositional data. No zero values are allowed.

nterms The number of terms to include in the final model.

type Either "alr" or "log" corresponding to the additive log-ratio transformation or the simple logarithm applied to the compositional data.

xnew If you have new data use it, otherwise leave it NULL.

Details

This is the standard projection pursuit. See the built-in function "ppr" for more details. When the data are transformed with the additive log-ratio transformation this is close in spirit to the log-contrast regression.

Value

A list including:

`runtime` The runtime of the regression.

`mod` The produced model as returned by the function "ppr".

`est` The fitted values of xnew if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression. Journal of the American Statistical Association, 76, 817-823. doi: 10.2307/2287576.

See Also

`pprcomp.tune`, `ice.pprcomp`, `alfa.pcr`, `lc.reg`, `comp.ppr`

Examples

```
x <- as.matrix( iris[, 2:4] )
x <- x/ rowSums(x)
y <- iris[, 1]
pprcomp(y, x)
```

`Projection pursuit regression with compositional predictor variables using the alpha-transformation`

Projection pursuit regression with compositional predictor variables using the α-transformation

---

Description

Projection pursuit regression with compositional predictor variables using the α-transformation.

Usage

```
alfa.pprcomp(y, x, nterms = 3, a, xnew = NULL)
```

Arguments

`y`	A numerical vector with the continuous variable.
`x`	A matrix with the compositional data. Zero values are allowed.
`nterms`	The number of terms to include in the final model.
`a`	The value of α for the α-transformation.
`xnew`	If you have new data use it, otherwise leave it NULL.

Details

This is the standard projection pursuit. See the built-in function "ppr" for more details. The compositional data are transformed with the α-transformation

Value

A list including:

`runtime`	The runtime of the regression.
`mod`	The produced model as returned by the function "ppr".
`est`	The fitted values of xnew if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression. Journal of the American Statistical Association, 76, 817-823. doi: 10.2307/2287576.

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

`alfapprcomp.tune`, `pprcomp`, `comp.ppr`

Examples

```
x <- as.matrix( iris[, 2:4] )
x <- x / rowSums(x)
y <- iris[, 1]
alfa.pprcomp(y, x, a = 0.5)
```

`Projections based test for distributional equality of two groups` *Projections based test for distributional equality of two groups*

Description

Projections based test for distributional equality of two groups.

Usage

```
dptest(x1, x2, B = 100)
```

Arguments

`x1`	A matrix containing compositional data of the first group.
`x2`	A matrix containing compositional data of the second group.
`B`	The number of random uniform projections to use.

Details

The test compares the distributions of two compositional datasets using random projections. For more details see Cuesta-Albertos, Cuevas and Fraiman (2009).

Value

A vector including:

`pvalues`	The p-values of the Kolmogorov-Smirnov tests.
`pvalue`	The p-value of the test based on the Benjamini and Heller (2008) procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>`.

References

Cuesta-Albertos J. A., Cuevas A. and Fraiman, R. (2009). On projection-based tests for directional and compositional data. Statistics and Computing, 19: 367–380.

Benjamini Y. and Heller R. (2008). Screening for partial conjunction hypotheses. Biometrics, 64(4): 1215–1222.

See Also

`comp.test`

Examples

```
x1 <- rdiri(50, c(3, 4, 5)) ## Fisher distribution with low concentration
x2 <- rdiri(50, c(3, 4, 5))
dptest(x1, x2)
```

Proportionality correlation coefficient matrix

Proportionality correlation coefficient matrix

Description

Proportionality correlation coefficient matrix.

Usage

```
pcc(x)
```

Arguments

x	A numerical matrix with the compositional data. Zeros are not allowed as the logarithm is applied.

Details

The function returns the proportionality correlation coefficient matrix. See Lovell et al. (2015) for more information.

Value

A matrix with the alr transformed data (if alr is used) or with the compositional data (if the alrinv is used).

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Zheng, B. (2000). Summarizing the goodness of fit of generalized linear models for longitudinal data. Statistics in medicine, 19(10), 1265-1275.

Lovell D., Pawlowsky-Glahn V., Egozcue J. J., Marguerat S. and Bahler, J. (2015). Proportionality: a valid alternative to correlation for relative data. PLoS Computational Biology, 11(3), e1004075.

See Also

acor, alr

Examples

```
x <- Compositional::rdiri(100, runif(4) )
a <- Compositional::pcc(x)
```

Quasi binomial regression for proportions

Quasi binomial regression for proportions

Description

Quasi binomial regression for proportions.

Usage

```
propreg(y, x, varb = "quasi", tol = 1e-07, maxiters = 100)
propregs(y, x, varb = "quasi", tol = 1e-07, logged = FALSE, maxiters = 100)
```

Arguments

y	A numerical vector proportions. 0s and 1s are allowed.
x	For the "propreg" a matrix with data, the predictor variables. This can be a matrix or a data frame. For the "propregs" this must be a numerical matrix, where each columns denotes a variable.
tol	The tolerance value to terminate the Newton-Raphson algorithm. This is set to 10^{-9} by default.
varb	The type of estimate to be used in order to estimate the covariance matrix of the regression coefficients. There are two options, either "quasi" (default value) or "glm". See the references for more information.
logged	Should the p-values be returned (FALSE) or their logarithm (TRUE)?
maxiters	The maximum number of iterations before the Newton-Raphson is terminated automatically.

Details

We are using the Newton-Raphson, but unlike R's built-in function "glm" we do no checks and no extra calculations, or whatever. Simply the model. The "propregs" is to be used for very many univariate regressions. The "x" is a matrix in this case and the significance of each variable (column of the matrix) is tested. The function accepts binary responses as well (0 or 1).

Value

For the "propreg" function a list including:

iters	The number of iterations required by the Newton-Raphson.
varb	The covariance matrix of the regression coefficients.

`phi` The phi parameter is returned if the input argument "varb" was set to "glm", othwerise this is NULL.

`info` A table similar to the one produced by "glm" with the estimated regression coefficients, their standard error, Wald test statistic and p-values.

For the "propregs" a two-column matrix with the test statistics (Wald statistic) and the associated p-values (or their loggarithm).

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Papke L. E. & Wooldridge J. (1996). Econometric methods for fractional response variables with an application to 401(K) plan participation rates. Journal of Applied Econometrics, 11(6): 619–632.

McCullagh, Peter, and John A. Nelder. Generalized linear models. CRC press, USA, 2nd edition, 1989.

See Also

`ols.compreg` `beta.reg`

Examples

```
y <- rbeta(100, 1, 4)
x <- matrix(rnorm(100 * 3), ncol = 3)
a <- propreg(y, x)
y <- rbeta(100, 1, 4)
x <- matrix(rnorm(400 * 100), ncol = 400)
b <- propregs(y, x)
mean(b[, 2] < 0.05)
```

Random values generation from some univariate distributions defined on the (0,1) interval

Random values generation from some univariate distributions defined on the $(0,1)$ interval

---

Description

Random values generation from some univariate distributions defined on the $(0,1)$ interval.

Usage

```
rbeta1(n, a)
runitweibull(n, a, b)
rlogitnorm(n, m, s, fast = FALSE)
```

Arguments

`n`	The sample size, a numerical value.
`a`	The shape parameter of the beta distribution. In the case of the unit Weibull, this is the shape parameter.
`b`	This is the scale parameter for the unit Weibull distribution.
`m`	The mean of the univariate normal in R.
`s`	The standard deviation of the univariate normal in R.
`fast`	If you want a faster generation set this equal to TRUE. This will use the Rnorm() function from the Rfast package. However, the speed is only observable if you want to simulate at least 500 (this number may vary among computers) observations. The larger the sample size the higher the speed-up.

Details

The function genrates random values from the Be(a, 1), the unit Weibull or the univariate logistic normal distribution.

Value

A vector with the simulated data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

See Also

`beta.est`, `colbeta.est`, `rdiri`

Examples

```
x <- rbeta1(100, 3)
```

`Read a file as a Filebacked Big Matrix`

Read a file as a Filebacked Big Matrix

Description

Read a file as a Filebacked Big Matrix.

Usage

```
read.fbm(file, select)
```

Arguments

`file`	The File to read.
`select`	Indices of columns to read (sorted). The length of select will be the number of columns of the resulting FBM.

Details

The functions read a file as a Filebacked Big Matrix object. For more information see the "bigstatsr" package.

Value

A Filebacked Big Matrix object.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

See Also

`scls`

Examples

```
x <- matrix( runif(50 * 20, 0, 2*pi), ncol = 20 )
```

`Regression with compositional data using the alpha-transformation`
Regression with compositional data using the α-transformation

---

Description

Regression with compositional data using the α-transformation.

Usage

```
alfa.reg(y, x, a, seb = NULL, xnew = NULL, yb = NULL)
alfa.reg2(y, x, a, xnew = NULL)
alfa.reg3(y, x, a = c(-1, 1), xnew = NULL)
```

Arguments

`y` A matrix with the compositional data.

`x` A matrix with the continuous predictor variables or a data frame including categorical predictor variables.

`a` The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied and the solution exists in a closed form, since it the classical mutivariate regression. For the alfa.reg2() this should be a vector of α values and the function call repeatedly the alfa.reg() function. For the alfa.reg3() function it should be a vector with two values, the endpoints of the interval of α. This function searches for the optimal vaue of α that minimizes the sum of squares of the errors. Using the `optimize` function it searches for the optimal value of α. Instead of choosing the value of α using `alfareg.tune` (that uses cross-validation) one can select it this way.

`seb` If this is NULL, the standard errors of the coefficients will not be returned. For reasons of possible numerical purposes or errors, you may want to leave this NULL.

`xnew` If you have new data use it, otherwise leave it NULL.

`yb` If you have already transformed the data using the α-transformation with the same α as given in the argument "a", put it here. Othewrise leave it NULL.

This is intended to be used in the function `alfareg.tune` in order to speed up the process. The time difference in that function is small for small samples. But, if you have a few thousands and or a few more components, there will be bigger differences.

Details

The α-transformation is applied to the compositional data first and then multivariate regression is applied. This involves numerical optimisation. The alfa.reg2() function accepts a vector with many values of α, while the the alfa.reg3() function searches for the value of α that minimizes the Kulback-Leibler divergence between the observed and the fitted compositional values. The functions are highly optimized.

Value

For the alfa.reg() function a list including:

`runtime` The time required by the regression.

`be` The beta coefficients.

`seb` The standard error of the beta coefficients.

`est` The fitted values for xnew if xnew is not NULL.

For the alfa.reg2() function a list with as many sublists as the number of values of α. Each element (sublist) of the list contains the above outcomes of the alfa.reg() function.

For the alfa.reg3() function a list with all previous elements plus an output "alfa", the optimal value of α.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris M. (2015). Regression analysis with compositional data containing zero values. Chilean Journal of Statistics, 6(2): 47-57. https://arxiv.org/pdf/1508.01913v1.pdf

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

Mardia K.V., Kent J.T., and Bibby J.M. (1979). Multivariate analysis. Academic press.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

alfareg.tune, diri.reg, js.compreg, kl.compreg,ols.compreg, comp.reg

Examples

```
library(MASS)
x <- as.vector(fgl[1:40, 1])
y <- as.matrix(fgl[1:40, 2:9])
y <- y / rowSums(y)
mod <- alfa.reg(y, x, 0.2)
```

Regularised and flexible discriminant analysis for compositional data using the alpha-transformation

Regularised and flexible discriminant analysis for compositional data using the α-transformation

---

Description

Regularised and flexible discriminant analysis for compositional data using the α-transformation.

Usage

```
alfa.rda(xnew, x, ina, a, gam = 1, del = 0)
alfa.fda(xnew, x, ina, a)
```

Arguments

`xnew` A matrix with the new compositional data whose group is to be predicted. Zeros are allowed, but you must be careful to choose strictly positive vcalues of α.

`x` A matrix with the available compositional data. Zeros are allowed, but you must be careful to choose strictly positive vcalues of α.

`ina` A group indicator variable for the available data.

`a` The value of α for the α-transformation.

`gam` This is a number between 0 and 1. It is the weight of the pooled covariance and the diagonal matrix.

`del` This is a number between 0 and 1. It is the weight of the LDA and QDA.

Details

For the alfa.rda, the covariance matrix of each group is calcualted and then the pooled covariance matrix. The spherical covariance matrix consists of the average of the pooled variances in its diagonal and zeros in the off-diagonal elements. gam is the weight of the pooled covariance matrix and 1-gam is the weight of the spherical covariance matrix, Sa = gam * Sp + (1-gam) * sp. Then it is a compromise between LDA and QDA. del is the weight of Sa and 1-del the weight of each group covariance group.

For the alfa.fda a flexible discriminant analysis is performed. See the R package **fda** for more details.

Value

For the alfa.rda a list including:

`prob` The estimated probabilities of the new data of belonging to each group.

`scores` The estimated socres of the new data of each group.

`est` The estimated group membership of the new data.

For the alfa.fda a list including:

`mod` An fda object as returned by the command fda of the R package mda.

`est` The estimated group membership of the new data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Giorgos Athineou <gioathineou@gmail.com>.

References

Friedman Jerome, Trevor Hastie and Robert Tibshirani (2009). The elements of statistical learning, 2nd edition. Springer, Berlin.

Tsagris Michail, Simon Preston and Andrew T.A. Wood (2016). Improved classification for compositional data using the α-transformation. Journal of classification, 33(2): 243-261. https://arxiv.org/pdf/1106.1451.pdf

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

Hastie, Tibshirani and Buja (1994). Flexible Disriminant Analysis by Optimal Scoring. Journal of the American Statistical Association, 89(428):1255-1270.

See Also

alfa, alfarda.tune, alfa.knn, alfa.nb, comp.nb, mix.compnorm

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
ina <- iris[, 5]
mod <- alfa.rda(x, x, ina, 0)
table(ina, mod$est)
mod2 <- alfa.fda(x, x, ina, 0)
table(ina, mod2$est)
```

Regularised discriminant analysis for Euclidean data

Regularised discriminant analysis for Euclidean data

Description

Regularised discriminant analysis for Euclidean data.

Usage

```
rda(xnew, x, ina, gam = 1, del = 0)
```

Arguments

xnew	A matrix with the new data whose group is to be predicted. They have to be continuous.
x	A matrix with the available data. They have to be continuous.
ina	A group indicator variable for the avaiable data.
gam	This is a number between 0 and 1. It is the weight of the pooled covariance and the diagonal matrix.
del	This is a number between 0 and 1. It is the weight of the LDA and QDA.

Details

The covariance matrix of each group is calculated and then the pooled covariance matrix. The spherical covariance matrix consists of the average of the pooled variances in its diagonal and zeros in the off-diagonal elements. gam is the weight of the pooled covariance matrix and 1-gam is the weight of the spherical covariance matrix, Sa = gam * Sp + (1-gam) * sp. Then it is a compromise between LDA and QDA. del is the weight of Sa and 1-del the weight of each group covariance group.

Value

A list including:

`prob`	The estimated probabilities of the new data of belonging to each group.
`scores`	The estimated socres of the new data of each group.
`est`	The estimated group membership of the new data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Friedman J.H. (1989): Regularized Discriminant Analysis. Journal of the American Statistical Association 84(405): 165–175.

Friedman Jerome, Trevor Hastie and Robert Tibshirani (2009). The elements of statistical learning, 2nd edition. Springer, Berlin.

Tsagris M., Preston S. and Wood A.T.A. (2016). Improved classification for compositional data using the α-transformation. Journal of Classification, 33(2): 243–261.

See Also

rda.tune

Examples

```
x <- as.matrix(iris[, 1:4])
ina <- iris[, 5]
mod <- rda(x, x, ina)
table(ina, mod$est)
```

Ridge regression *Ridge regression*

Description

Ridge regression.

Usage

```
ridge.reg(y, x, lambda, B = 1, xnew = NULL)
```

Arguments

y	A real valued vector. If it contains percentages, the logit transformation is applied.
x	A matrix with the predictor variable(s), they have to be continuous.
lambda	The value of the regularisation parameter λ.
B	If B = 1 (default value) no bootstrpa is performed. Otherwise bootstrap standard errors are returned.
xnew	If you have new data whose response value you want to predict put it here, otherwise leave it as is.

Details

This is used in the function `alfa.ridge`. There is also a built-in function available from the MASS library, called "lm.ridge".

Value

A list including:

beta	The beta coefficients.
seb	The standard eror of the coefficiens. If B > 1 the bootstrap standard errors will be returned.
est	The fitted or the predicted values (if xnew is not NULL).

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Giorgos Athineou <gioathineou@gmail.com>.

References

Hoerl A.E. and R.W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1): 55-67.

Brown P. J. (1994). Measurement, Regression and Calibration. Oxford Science Publications.

See Also

ridge.tune, alfa.ridge, ridge.plot

Examples

```
y <- as.vector(iris[, 1])
x <- as.matrix(iris[, 2:4])
mod1 <- ridge.reg(y, x, lambda = 0.1)
mod2 <- ridge.reg(y, x, lambda = 0)
```

Ridge regression plot *Ridge regression plot*

Description

A plot of the regularised regression coefficients is shown.

Usage

```
ridge.plot(y, x, lambda = seq(0, 5, by = 0.1) )
```

Arguments

y	A numeric vector containing the values of the target variable. If the values are proportions or percentages, i.e. strictly within 0 and 1 they are mapped into R using the logit transformation. In any case, they must be continuous only.
x	A numeric matrix containing the continuous variables. Rows are samples and columns are features.
lambda	A grid of values of the regularisation parameter λ.

Details

For every value of λ the coefficients are obtained. They are plotted versus the λ values.

Value

A plot with the values of the coefficients as a function of λ.

Author(s)

Michail Tsagris.

R implementation and documentation: Giorgos Athineou <gioathineou@gmail.com> and Michail Tsagris <mtsagris@uoc.gr>.

References

Hoerl A.E. and R.W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1): 55-67.

Brown P. J. (1994). Measurement, Regression and Calibration. Oxford Science Publications.

See Also

`ridge.reg`, `ridge.tune`, `alfa.ridge`, `alfaridge.plot`

Examples

```
y <- as.vector(iris[, 1])
x <- as.matrix(iris[, 2:4])
ridge.plot(y, x, lambda = seq(0, 2, by = 0.1) )
```

`Ridge regression with compositional data in the covariates side using the alpha-transformation`

Ridge regression with compositional data in the covariates side using the α-transformation

---

Description

Ridge regression with compositional data in the covariates side using the α-transformation.

Usage

```
alfa.ridge(y, x, a, lambda, B = 1, xnew = NULL)
```

Arguments

`y`	A numerical vector containing the response variable values. If they are percentages, they are mapped onto R using the logit transformation.
`x`	A matrix with the predictor variables, the compositional data. Zero values are allowed, but you must be careful to choose strictly positive vcalues of α.
`a`	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
`lambda`	The value of the regularisation parameter, λ.
`B`	If B > 1 bootstrap estimation of the standard errors is implemented.
`xnew`	A matrix containing the new compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.

Details

The α-transformation is applied to the compositional data first and then ridge components regression is performed.

Value

The output of the ridge.reg.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>.

References

Tsagris M. (2015). Regression analysis with compositional data containing zero values. Chilean Journal of Statistics, 6(2): 47-57. https://arxiv.org/pdf/1508.01913v1.pdf

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

ridge.reg, alfaridge.tune, alfaridge.plot

Examples

```
library(MASS)
y <- as.vector(fgl[, 1])
x <- as.matrix(fgl[, 2:9])
x <- x/ rowSums(x)
mod1 <- alfa.ridge(y, x, a = 0.5, lambda = 0.1, B = 1, xnew = NULL)
mod2 <- alfa.ridge(y, x, a = 0.5, lambda = 1, B = 1, xnew = NULL)
```

Ridge regression with the alpha-transformation plot

Ridge regression plot

Description

A plot of the regularised regression coefficients is shown.

Usage

```
alfaridge.plot(y, x, a, lambda = seq(0, 5, by = 0.1) )
```

Arguments

y	A numeric vector containing the values of the target variable. If the values are proportions or percentages, i.e. strictly within 0 and 1 they are mapped into R using the logit transformation. In any case, they must be continuous only.
x	A numeric matrix containing the continuous variables.
a	The value of the α-transformation. It has to be between -1 and 1. If there are zero values in the data, you must use a strictly positive value.
lambda	A grid of values of the regularisation parameter λ.

Details

For every value of λ the coefficients are obtained. They are plotted versus the λ values.

Value

A plot with the values of the coefficients as a function of λ.

Author(s)

Michail Tsagris.

R implementation and documentation: Giorgos Athineou <gioathineou@gmail.com> and Michail Tsagris <mtsagris@uoc.gr>.

References

Hoerl A.E. and R.W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1): 55-67.

Brown P. J. (1994). Measurement, Regression and Calibration. Oxford Science Publications.

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

ridge.plot, alfa.ridge

Examples

```
library(MASS)
y <- as.vector(fgl[, 1])
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
alfaridge.plot(y, x, a = 0.5, lambda = seq(0, 5, by = 0.1) )
```

Simplicial constrained median regression for compositional responses and predictors model

Simplicial constrained median regression for compositional responses and predictors

Description

Simplicial constrained median regression for compositional responses and predictors.

Usage

```
scrq(y, x, xnew = NULL)
```

Arguments

y	A matrix with the compositional data (dependent variable). Zero values are allowed.
x	A matrix with the compositional predictors. Zero values are allowed.
xnew	If you have new data use it, otherwise leave it NULL.

Details

The function performs median regression where the beta coefficients are constained to be positive and sum to 1.

Value

A list including:

mlad	The mean absolute deviation.
be	The beta coefficients.
est	The fitted of xnew if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

scls, tflr

Examples

```
library(MASS)
set.seed(1234)
y <- rdiri(214, runif(4, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- scrq(y, x)
mod
```

`Simulation of compositional data from Gaussian mixture models`
Simulation of compositional data from Gaussian mixture models

---

Description

Simulation of compositional data from Gaussian mixture models.

Usage

```
rmixcomp(n, prob, mu, sigma, type = "alr")
```

Arguments

`n`	The sample size.
`prob`	A vector with mixing probabilities. Its length is equal to the number of clusters.
`mu`	A matrix where each row corresponds to the mean vector of each cluster.
`sigma`	An array consisting of the covariance matrix of each cluster.
`type`	Should the additive ("type=alr") or the isometric (type="ilr") log-ration be used? The default value is for the additive log-ratio transformation.

Details

A sample from a multivariate Gaussian mixture model is generated.

Value

A list including:

`id`	A numeric variable indicating the cluster of simulated vector.
`x`	A matrix containing the simulated compositional data. The number of dimensions will be + 1.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Ryan P. Browne, Aisha ElSherbiny and Paul D. McNicholas (2015). R package mixture: Mixture Models for Clustering and Classification.

See Also

mix.compnorm, bic.mixcompnorm

Examples

```
p <- c(1/3, 1/3, 1/3)
mu <- matrix(nrow = 3, ncol = 4)
s <- array( dim = c(4, 4, 3) )
x <- as.matrix(iris[, 1:4])
ina <- as.numeric(iris[, 5])
mu <- rowsum(x, ina) / 50
s[, , 1] <- cov(x[ina == 1, ])
s[, , 2] <- cov(x[ina == 2, ])
s[, , 3] <- cov(x[ina == 3, ])
y <- rmixcomp(100, p, mu, s, type = "alr")
```

Simulation of compositional data from mixtures of Dirichlet distributions

Simulation of compositional data from mixtures of Dirichlet distributions

---

Description

Simulation of compositional data from mixtures of Dirichlet distributions.

Usage

```
rmixdiri(n, a, prob)
```

Arguments

n	The sample size.
a	A matrix where each row contains the parameters of each Dirichlet component.
prob	A vector with the mixing probabilities.

Details

A sample from a Dirichlet mixture model is generated.

Value

A list including:

id	A numeric variable indicating the cluster of simulated vector.
x	A matrix containing the simulated compositional data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Ye X., Yu Y. K. and Altschul S. F. (2011). On the inference of Dirichlet mixture priors for protein sequence comparison. Journal of Computational Biology, 18(8), 941-954.

See Also

rmixcomp, mixdiri.contour,

Examples

```
a <- matrix( c(12, 30, 45, 32, 50, 16), byrow = TRUE,ncol = 3)
prob <- c(0.5, 0.5)
x <- rmixdiri(100, a, prob)
```

Simulation of compositional data from the Flexible Dirichlet distribution

Simulation of compositional data from the Flexible Dirichlet distribution

Description

Simulation of compositional data from the Flexible Dirichlet distribution.

Usage

```
rfd(n, alpha, prob, tau)
```

Arguments

n	The sample size.
alpha	A vector of the non-negative α parameters.
prob	A vector of the clusters' probabilities that must sum to one.
tau	The positive scalar tau parameter.

Details

For more information see the references and the package FlxeDir.

Value

A matrix with compositional data.

Author(s)

Michail Tsagris ported from the R package FlexDir. <mtsagris@uoc.gr>.

References

Ongaro A. and Migliorati S. (2013). A generalization of the Dirichlet distribution. Journal of Multivariate Analysis, 114, 412–426.

Migliorati S., Ongaro A. and Monti G. S. (2017). A structured Dirichlet mixture model for compositional data: inferential and applicative issues. Statistics and Computing, 27, 963–983.

See Also

dfd

Examples

```
alpha <- c(12, 11, 10)
prob <- c(0.25, 0.25, 0.5)
x <- rfd(100, alpha, prob, 7)
```

Simulation of compositional data from the folded normal distribution

Simulation of compositional data from the folded model normal distribution

---

Description

Simulation of compositional data from the folded model normal distribution.

Usage

```
rfolded(n, mu, su, a)
```

Arguments

n	The sample size.
mu	The mean vector.
su	The covariance matrix.
a	The value of α.

Details

A sample from the folded model is generated.

Value

A matrix with compositional data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris M. and Stewart C. (2020). A folded model for compositional data analysis. Australian and New Zealand Journal of Statistics, 62(2): 249-277. https://arxiv.org/pdf/1802.07330.pdf

See Also

alfa, alpha.mle, a.est

Examples

```
s <-  c(0.1490676523, -0.4580818209,  0.0020395316, -0.0047446076, -0.4580818209,
1.5227259250,  0.0002596411,  0.0074836251,  0.0020395316,  0.0002596411,
0.0365384838, -0.0471448849, -0.0047446076,  0.0074836251, -0.0471448849,
0.0611442781)
s <- matrix(s, ncol = 4)
m <- c(1.715, 0.914, 0.115, 0.167)
x <- rfolded(100, m, s, 0.5)
a.est(x)
```

`Spatial median regression`

Spatial median regression

Description

Spatial median regression with Euclidean data.

Usage

```
spatmed.reg(y, x, xnew = NULL, tol = 1e-07, ses = FALSE)
```

Arguments

`y`	A matrix with the compositional data. Zero values are not allowed.
`x`	The predictor variable(s), they have to be continuous.
`xnew`	If you have new data use it, otherwise leave it NULL.
`tol`	The threshold upon which to stop the iterations of the Newton-Rapshon algorithm.
`ses`	If you want to extract the standard errors of the parameters, set this to TRUE. Be careful though as this can slow down the algorithm dramatically. In a run example with 10,000 observations and 10 variables for y and 30 for x, when ses = FALSE the algorithm can take 0.20 seconds, but when ses = TRUE it can go up to 140 seconds.

Details

The objective function is the minimization of the sum of the absolute residuals. It is the multivariate generalization of the median regression. This function is used by `comp.reg`.

Value

A list including:

`iter`	The number of iterations that were required.
`runtime`	The time required by the regression.
`be`	The beta coefficients.
`seb`	The standard error of the beta coefficients is returned if ses=TRUE and NULL otherwise.
`est`	The fitted of xnew if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Biman Chakraborty (2003). On multivariate quantile regression. Journal of Statistical Planning and Inference, 110(1-2), 109-132. http://www.stat.nus.edu.sg/export/sites/dsap/research/documents/tr01_2000.pdf

See Also

`multivreg`, `comp.reg`, `alfa.reg`, `js.compreg`, `diri.reg`

Examples

```
library(MASS)
x <- as.matrix(iris[, 3:4])
y <- as.matrix(iris[, 1:2])
mod1 <- spatmed.reg(y, x)
mod2 <- multivreg(y, x, plot = FALSE)
```

`Ternary diagram` *Ternary diagram*

Description

Ternary diagram.

Usage

```
ternary(x, dg = FALSE, hg = FALSE, means = TRUE, pca = FALSE, colour = NULL)
```

Arguments

`x`	A matrix with the compositional data.
`dg`	Do you want diagonal grid lines to appear? If yes, set this TRUE.
`hg`	Do you want horizontal grid lines to appear? If yes, set this TRUE.
`means`	A boolean variable. Should the closed geometric mean and the arithmetic mean appear (TRUE) or not (FALSE)?.
`pca`	Should the first PCA calculated Aitchison (1983) described appear? If yes, then this should be TRUE, or FALSE otherwise.
`colour`	If you want the points to appear in different colour put a vector with the colour numbers or colours.

Details

There are two ways to create a ternary graph. We used here that one where each edge is equal to 1 and it is what Aitchison (1986) uses. For every given point, the sum of the distances from the edges is equal to 1. Horizontal and or diagonal grid lines can appear, so as the closed geometric and the simple arithmetic mean. The first PCA is calculated using the centred log-ratio transformation as Aitchison (1983, 1986) suggested. If the data contain zero values, the first PCA will not be plotted. Zeros in the data appear with green circles in the triangle and you will also see NaN in the closed geometric mean.

Value

The ternary plot and a 2-row matrix with the means. The closed geometric and the simple arithmetic mean vector and or the first principal component will appear as well if the user has asked for them. Additionally, horizontal or diagonal grid lines can appear as well.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>.

References

Aitchison, J. (1983). Principal component analysis of compositional data. Biometrika 70(1): 57–65.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

ternary.mcr, ternary.reg, diri.contour

Examples

```
x <- as.matrix(iris[, 1:3])
x <- x / rowSums(x)
ternary(x, means = TRUE, pca = TRUE)
```

Ternary diagram of regression models

Ternary diagram of regression models

Description

Ternary diagram of regression models.

Usage

```
ternary.reg(y, est, id, labs)
```

Arguments

y	A matrix with the compositional data.
est	A matrix with all fitted compositional data for all regression models, one under the other.
id	A vector indicating the regression model of each fitted compositional data set.
labs	The names of the regression models to appea in the legend.

Details

The points first appear on the ternary plot. Then, the fitted compositional data appear with different lines for each regression model.

Value

The ternary plot and lines for the fitted values of each regression model.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

ternary, ternary.mcr, diri.contour

Examples

```
x <- cbind(1, rnorm(50) )
a <- exp( x %*% matrix( rnorm(6,0, 0.4), ncol = 3) )
y <- matrix(NA, 50, 3)
for (i in 1:50) y[i, ] <- rdiri(1, a[i, ])
est <- comp.reg(y, x[, -1], xnew = x[, -1])$est
ternary.reg(y, est, id = rep(1, 50), labs = "ALR regression")
```

Ternary diagram with confidence region for the matrix of coefficients of the SCLS or the TFLR model

Ternary diagram with confidence region for the matrix of coefficients of the SCLS or the TFLR model

---

Description

Ternary diagram with confidence region for the matrix of coefficients of the SCLS or the TFLR model.

Usage

```
ternary.coefcr(y, x, type = "scls", conf = 0.95, R = 1000, dg = FALSE, hg = FALSE)
```

Arguments

y	A matrix with the response compositional data.
x	A matrix with the predictor compositional data.
type	The type of model to use, "scls" or "tflr". Depending on the model selected, the function will construct the confidence regions of the estimated matrix of coefficients of that model.

`conf`	The confidence level, by default this is set to 0.95.
`R`	Number of bootstrap replicates to run.
`dg`	Do you want diagonal grid lines to appear? If yes, set this TRUE.
`hg`	Do you want horizontal grid lines to appear? If yes, set this TRUE.

Details

This function runs the SCLS or the TFLR model and constructs confidence regions for the estimated matrix of regression coefficients using non-parametric bootstrap.

Value

A ternary plot of the estimated matrix of coefficients of the SCLS or of the TFLR model, and their associated confidence regions.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Fiksel J., Zeger S. and Datta A. (2022). A transformation-free linear regression for compositional outcomes and predictors. Biometrics, 78(3): 974–987.

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

`ternary`, `scls`, `tflr`, `ternary.mcr`

Examples

```
y <- rdiri(50, runif(3))
x <- rdiri(50, runif(4))
ternary.coefcr(y, x, R = 500, dg = TRUE, hg = TRUE)
```

`Ternary diagram with confidence region for the mean`

Ternary diagram with confidence region for the mean

Description

Ternary diagram with confidence region for the mean.

Usage

```
ternary.mcr(x, type = "alr", conf = 0.95, dg = FALSE, hg = FALSE, colour = NULL)
```

Arguments

`x`	A matrix with the compositional data.
`dg`	Do you want diagonal grid lines to appear? If yes, set this TRUE.
`type`	The type of log-ratio transformation to aply, the "alr" or the "ilr".
`conf`	The confidence level, by default this is set to 0.95.
`hg`	Do you want horizontal grid lines to appear? If yes, set this TRUE.
`colour`	If you want the points to appear in different colour put a vector with the colour numbers or colours.

Details

Ternary plot of compositional data including the log-ratio mean and its confidence region. The confidence region is based on the Hotelling T^2 test statistic of the log-ratio transformed data.

Value

A ternary plot of compositional data including the log-ratio mean and its confidence region.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aitchison, J. (1983). Principal component analysis of compositional data. Biometrika 70(1): 57–65.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`ternary`, `ternary.reg`, `diri.contour`

Examples

```
x <- as.matrix(iris[, 1:3])
x <- x / rowSums(x)
ternary.mcr(x, type = "alr", dg = TRUE, hg = TRUE)
```

`Ternary diagram with the coefficients of the simplicial-simplicial regression models`

Ternary diagram with the coefficients of the simplicial-simplicial regression models

Description

Ternary diagram with the coefficients of the simplicial-simplicial regression models.

Usage

```
ternary.coef(B, dg = FALSE, hg = FALSE, colour = NULL)
```

Arguments

`B`	A matrix with the coefficients of the `tflr` or the `scls` functions. See examples for this.
`dg`	Do you want diagonal grid lines to appear? If yes, set this TRUE.
`hg`	Do you want horizontal grid lines to appear? If yes, set this TRUE.
`colour`	If you want the points to appear in different colour put a vector with the colour numbers or colours.

Details

Ternary plot of the coefficients of the `tflr` or the `scls` functions.

Value

A ternary plot of the coefficients of the `tflr` or the `scls` functions.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aitchison, J. (1983). Principal component analysis of compositional data. Biometrika 70(1): 57–65.

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`ternary`, `ternary.reg`, `scls`

Examples

```
y <- as.matrix(iris[, 1:3])
y <- y / rowSums(y)
x <- rdiri(150, runif(5, 1,4) )
mod <- scls(y, x)
ternary.coef(mod$be)
```

`The additive log-ratio transformation and its inverse` *The additive log-ratio transformation and its inverse*

Description

The additive log-ratio transformation and its inverse.

Usage

```
alr(x)
alrinv(y)
```

Arguments

`x`	A numerical matrix with the compositional data.
`y`	A numerical matrix with data to be closed into the simplex.

Details

The additive log-ratio transformation with the first component being the common divisor is applied. The inverse of this transformation is also available. This means that no zeros are allowed.

Value

A matrix with the alr transformed data (if alr is used) or with the compositional data (if the alrinv is used).

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`bc`, `pivot`, `fp`, `green`, `alfa`, `alfainv`

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y <- alr(x)
x1 <- alrinv(y)
```

The alpha-distance *The α-distance*

Description

This is the Euclidean (or Manhattan) distance after the α-transformation has been applied.

Usage

```
alfadist(x, a, type = "euclidean", square = FALSE)
alfadista(xnew, x, a, type = "euclidean", square = FALSE)
```

Arguments

xnew	A matrix or a vector with new compositional data.
x	A matrix with the compositional data.
a	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$, the isometric log-ratio transformation is applied.
type	Which type distance do you want to calculate after the α-transformation, "euclidean", or "manhattan".
square	In the case of the Euclidean distance, you can choose to return the squared distance by setting this TRUE.

Details

The α-transformation is applied to the compositional data first and then the Euclidean or the Manhattan distance is calculated.

Value

For "alfadist" a matrix including the pairwise distances of all observations or the distances between xnew and x. For "alfadista" a matrix including the pairwise distances of all observations or the distances between xnew and x.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris M.T., Preston S. and Wood A.T.A. (2016). Improved classification for compositional data using the α-transformation. Journal of Classification. 33(2): 243–261. https://arxiv.org/pdf/1506.04976v2.pdf

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

`alfa`, `alfainv`, `alfa.reg`, `esov`

Examples

```
library(MASS)
x <- as.matrix(fgl[1:20, 2:9])
x <- x / rowSums(x)
alfadist(x, 0.1)
alfadist(x, 1)
```

`The alpha-IT transformation`

The α-IT transformation

Description

The α-IT transformation.

Usage

```
ait(x, a, h = TRUE)
```

Arguments

`x`	A matrix with the compositional data.
`a`	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
`h`	A boolean variable. If is TRUE (default value) the multiplication with the Helmert sub-matrix will take place. When $\alpha = 0$ and h = FALSE, the result is the centred log-ratio transformation (Aitchison, 1986). In general, when h = FALSE the resulting transformation maps the data onto a singualr space. The sum of the vectors is equal to 0. Hence, from the simplex constraint the data go to another constraint.

Details

The α-IT transformation is applied to the compositional data.

Value

A matrix with the α-IT transformed data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Clarotto L., Allard D. and Menafoglio A. (2022). A new class of α-transformations for the spatial analysis of Compositional Data. Spatial Statistics, 47.

See Also

aitdist, ait.knn, alfa, green, alr

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y1 <- ait(x, 0.2)
y2 <- ait(x, 1)
rbind( colMeans(y1), colMeans(y2) )
```

The alpha-IT-distance *The α-IT-distance*

Description

This is the Euclidean (or Manhattan) distance after the α-IT-transformation has been applied.

Usage

```
aitdist(x, a, type = "euclidean", square = FALSE)
aitdista(xnew, x, a, type = "euclidean", square = FALSE)
```

Arguments

xnew	A matrix or a vector with new compositional data.
x	A matrix with the compositional data.
a	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$, the isometric log-ratio transformation is applied.

`type`	Which type distance do you want to calculate after the α-transformation, "euclidean", or "manhattan".
`square`	In the case of the Euclidean distance, you can choose to return the squared distance by setting this TRUE.

Details

The α-IT-transformation is applied to the compositional data first and then the Euclidean or the Manhattan distance is calculated.

Value

For "alfadist" a matrix including the pairwise distances of all observations or the distances between xnew and x. For "alfadista" a matrix including the pairwise distances of all observations or the distances between xnew and x.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Clarotto L., Allard D. and Menafoglio A. (2021). A new class of α-transformations for the spatial analysis of Compositional Data. https://arxiv.org/abs/2110.07967

See Also

ait, alfadist, alfa

Examples

```
library(MASS)
x <- as.matrix(fgl[1:20, 2:9])
x <- x / rowSums(x)
aitdist(x, 0.1)
aitdist(x, 1)
```

The alpha-k-NN regression for compositional response data
The α-k-NN regression for compositional response data

Description

The α-k-NN regression for compositional response data.

Usage

```
aknn.reg(xnew, y, x, a = seq(0.1, 1, by = 0.1), k = 2:10,
apostasi = "euclidean", rann = FALSE)
```

Arguments

`xnew` A matrix with the new predictor variables whose compositions are to be predicted.

`y` A matrix with the compositional response data. Zeros are allowed.

`x` A matrix with the available predictor variables.

`a` The value(s) of α. Either a single value or a vector of values. As zero values in the compositional data are allowed, you must be careful to choose strictly positive vcalues of α. However, if negative values are passed, the positive ones are used only.

`k` The number of nearest neighbours to consider. It can be a single number or a vector.

`apostasi` The type of distance to use, either "euclidean" or "manhattan".

`rann` If you have large scale datasets and want a faster k-NN search, you can use kd-trees implemented in the R package "Rnanoflann". In this case you must set this argument equal to TRUE. Note however, that in this case, the only available distance is by default "euclidean".

Details

The α-k-NN regression for compositional response variables is applied.

Value

A list with the estimated compositional response data for each value of α and k.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris M., Alenazi A. and Stewart C. (2023). Flexible non-parametric regression models for compositional response data with zeros. Statistics and Computing, 33(106).

https://link.springer.com/article/10.1007/s11222-023-10277-5

See Also

`aknnreg.tune`, `akern.reg`, `alfa.reg`, `comp.ppr`, `comp.reg`, `kl.compreg`

Examples

```
y <- as.matrix( iris[, 1:3] )
y <- y / rowSums(y)
x <- iris[, 4]
mod <- aknn.reg(x, y, x, a = c(0.4, 0.5), k = 2:3, apostasi = "euclidean")
```

`The alpha-k-NN regression with compositional predictor variables`

The α-k-NN regression with compositional predictor variables

Description

The α-k-NN regression with compositional predictor variables.

Usage

```
alfa.knn.reg(xnew, y, x, a = 1, k = 2:10, apostasi = "euclidean", method = "average")
```

Arguments

`xnew` A matrix with the new compositional predictor variables whose response is to be predicted. Zeros are allowed.

`y` The response variable, a numerical vector.

`x` A matrix with the available compositional predictor variables. Zeros are allowed.

`a` A single value of α. As zero values in the compositional data are allowed, you must be careful to choose strictly positive vcalues of α. If negative values are passed, the positive ones are used only. If the data are already alpha-transformed, you can make this NULL.

`k` The number of nearest neighbours to consider. It can be a single number or a vector.

`apostasi` The type of distance to use, either "euclidean" or "manhattan".

`method` If you want to take the average of the reponses of the k closest observations, type "average". For the median, type "median" and for the harmonic mean, type "harmonic".

Details

The α-k-NN regression with compositional predictor variables is applied.

Value

A matrix with the estimated response data for each value of k.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris M., Alenazi A. and Stewart C. (2023). Flexible non-parametric regression models for compositional response data with zeros. Statistics and Computing, 33(106).

https://link.springer.com/article/10.1007/s11222-023-10277-5

See Also

`aknn.reg`, `alfa.knn`, `alfa.pcr`, `alfa.ridge`

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y <- fgl[, 1]
mod <- alfa.knn.reg(x, y, x, a = 0.5, k = 2:4)
```

`The alpha-kernel regression with compositional response data`

The α-kernel regression with compositional response data

Description

The α-kernel regression with compositional response data.

Usage

```
akern.reg( xnew, y, x, a = seq(0.1, 1, by = 0.1),
h = seq(0.1, 1, length = 10), type = "gauss" )
```

Arguments

`xnew` A matrix with the new predictor variables whose compositions are to be predicted.

`y` A matrix with the compositional response data. Zeros are allowed.

`x` A matrix with the available predictor variables.

`a` The value(s) of α. Either a single value or a vector of values. As zero values in the compositional data are allowed, you must be careful to choose strictly positive vcalues of α. However, if negative values are passed, the positive ones are used only.

`h` The bandwidth value(s) to consider.

`type` The type of kernel to use, "gauss" or "laplace".

Details

The α-kernel regression for compositional response variables is applied.

Value

A list with the estimated compositional response data for each value of α and h.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris M., Alenazi A. and Stewart C. (2023). Flexible non-parametric regression models for compositional response data with zeros. Statistics and Computing, 33(106).

https://link.springer.com/article/10.1007/s11222-023-10277-5

See Also

akernreg.tune, aknn.reg, aknnreg.tune,alfa.reg, comp.ppr, comp.reg, kl.compreg

Examples

```
y <- as.matrix( iris[, 1:3] )
y <- y / rowSums(y)
x <- iris[, 4]
mod <- akern.reg( x, y, x, a = c(0.4, 0.5), h = c(0.1, 0.2) )
```

The alpha-SCLS model *The α-SCLS model for compositional responses and predictors*

Description

The α-SCLS model for compositional responses and predictors.

Usage

```
ascls(y, x, a = seq(0.1, 1, by = 0.1), xnew)
```

Arguments

y	A matrix with the compositional data (dependent variable). Zero values are allowed.
x	A matrix with the compositional predictors. Zero values are allowed.
a	A vector or a single number of values of the α-parameter. This has to be different from zero, and it can take negative values if there are no zeros in the simplicial response (y).
xnew	The new data for which predictions will be made.

Details

This is an extension of the SCLS model that includes the α-transformation and is intended solely for prediction purposes.

Value

A list with matrices containing the predicted simplicial response values, one matrix for each value of α.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

scls, cv.ascls, atflr

Examples

```
library(MASS)
set.seed(1234)
y <- rdiri(214, runif(4, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- ascls(y, x, xnew = x)
mod
```

The alpha-TFLR model *The α-TFLR model for compositional responses and predictors*

Description

The α-TFLR model for compositional responses and predictors.

Usage

```
atflr(y, x, a = seq(0.1, 1, by = 0.1), xnew)
```

Arguments

y A matrix with the compositional data (dependent variable). Zero values are allowed.

x A matrix with the compositional predictors. Zero values are allowed.

a A vector or a single number of values of the α-parameter. This has to be different from zero, and it can take negative values if there are no zeros in the simplicial response (y).

xnew The new data for which predictions will be made.

Details

This is an extension of the TFLR model that includes the α-transformation and is intended solely for prediction purposes.

Value

A list with matrices containing the predicted simplicial response values, one matrix for each value of α.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Fiksel J., Zeger S. and Datta A. (2022). A transformation-free linear regression for compositional outcomes and predictors. Biometrics, 78(3): 974–987.

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

tflr, cv.atflr, ascls

Examples

```
library(MASS)
set.seed(1234)
y <- rdiri(214, runif(4, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- ascls(y, x, a = c(0.5, 1), xnew = x)
mod
```

The alpha-transformation

The α-transformation

Description

The α-transformation.

Usage

```
alfa(x, a, h = TRUE)
alef(x, a)
```

Arguments

x	A matrix with the compositional data.
a	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
h	A boolean variable. If is TRUE (default value) the multiplication with the Helmert sub-matrix will take place. When $\alpha = 0$ and h = FALSE, the result is the centred log-ratio transformation (Aitchison, 1986). In general, when h = FALSE the resulting transformation maps the data onto a singualr space. The sum of the vectors is equal to 0. Hence, from the simplex constraint the data go to another constraint.

Details

The α-transformation is applied to the compositional data. The command "alef" is the same as "alfa(x, a, h = FALSE)", but reurns a different element as well and is necessary for the functions a.est, a.mle and alpha.mle.

Value

A list including:

sa	The logarithm of the Jacobian determinant of the α-transformation. This is used in the "profile" function to speed up the computations.
sk	If the "alef" was called, this will return the sum of the α-power transformed data, prior to being normalised to sum to 1. If $\alpha = 0$, this will not be returned.
aff	The α-transformed data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>.

References

Tsagris M. and Stewart C. (2022). A Review of Flexible Transformations for Modeling Compositional Data. In Advances and Innovations in Statistics and Data Science, pp. 225–234. https://link.springer.com/chapter/10.10
3-031-08329-7_10

Tsagris Michail and Stewart Connie (2020). A folded model for compositional data analysis. Australian and New Zealand Journal of Statistics, 62(2): 249-277. https://arxiv.org/pdf/1802.07330.pdf

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`alfainv`, `pivot`, `alfa.profile`, `alfa.tune a.est`, `alpha.mle`, `alr`, `bc`, `fp`, `green`

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y1 <- alfa(x, 0.2)$aff
y2 <- alfa(x, 1)$aff
rbind( colMeans(y1), colMeans(y2) )
y3 <- alfa(x, 0.2)$aff
dim(y1)  ;  dim(y3)
rowSums(y1)
rowSums(y3)
```

`The Box-Cox transformation applied to ratios of components`

The Box-Cox transformation applied to ratios of components

Description

The Box-Cox transformation applied to ratios of components.

Usage

```
bc(x, lambda)
```

Arguments

`x` A matrix with the compositional data. The first component must be zero values free.

`lambda` The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\lambda = 0$ the additive log-ratio transformation (`alr`) is applied.

Details

The Box-Cox transformation applied to ratios of components, as described in Aitchison (1986) is applied.

Value

A matrix with the transformed data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

alr, fp, green, alfa

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y1 <- bc(x, 0.2)
y2 <- bc(x, 0)
rbind( colMeans(y1), colMeans(y2) )
rowSums(y1)
rowSums(y2)
```

The ESOV-distance *The ESOV-distance*

Description

The ESOV-distance.

Usage

```
esov(x)
esova(xnew, x)
es(x1, x2)
```

Arguments

`x`	A matrix with compositional data.
`xnew`	A matrix or a vector with new compositional data.
`x1`	A vector with compositional data.
`x2`	A vector with compositional data.

Details

The ESOV distance is calculated.

Value

For "esov()" a matrix including the pairwise distances of all observations or the distances between xnew and x.

For "esova()" a matrix including the pairwise distances of all observations or the distances between xnew and x.

For "es()" a number, the ESOV distance between x1 and x2.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris, Michail (2014). The k-NN algorithm for compositional data: a revised approach with and without zero values present. Journal of Data Science, 12(3): 519-534.

Endres, D. M. and Schindelin, J. E. (2003). A new metric for probability distributions. Information Theory, IEEE Transactions on 49, 1858-1860.

Osterreicher, F. and Vajda, I. (2003). A new class of metric divergences on probability spaces and its applicability in statistics. Annals of the Institute of Statistical Mathematics 55, 639-653.

See Also

`alfadist`, `comp.knn`, `js.compreg`

Examples

```
library(MASS)
x <- as.matrix(fgl[1:20, 2:9])
x <- x / rowSums(x)
esov(x)
```

`The folded power transformation` *The folded power transformation*

Description

The folded power transformation.

Usage

```
fp(x, lambda)
```

Arguments

`x`	A matrix with the compositional data. Zero values are allowed.
`lambda`	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\lambda = 0$ the additive log-ratio transformation (`alr`) is applied. If zero values are present λ must be strictly positive.

Details

The folded power transformation is applied to the compositional data.

Value

A matrix with the transformed data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Atkinson, A. C. (1985). Plots, transformations and regression; an introduction to graphical methods of diagnostic regression analysis Oxford University Press.

See Also

`alr`, `bc`, `green`, `alfa`

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y1 <- fp(x, 0.2)
y2 <- fp(x, 0)
rbind( colMeans(y1), colMeans(y2) )
rowSums(y1)
rowSums(y2)
```

`The Frechet mean for compositional data`

The Frechet mean for compositional data

Description

Mean vector or matrix with mean vectors of compositional data using the α-transformation.

Usage

```
frechet(x, a)
```

Arguments

`x`	A matrix with the compositional data.
`a`	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied and the closed geometric mean is calculated. You can also provide a sequence of values of alpha and in this case a matrix of Frechet means will be returned.

Details

The power transformation is applied to the compositional data and the mean vector is calculated. Then the inverse of it is calculated and the inverse of the power transformation applied to the last vector is the Frechet mean.

Value

If α is a single value, the function will return a vector with the Frechet mean for the given value of α. Otherwise the function will return a matrix with the Frechet means for each value of α.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Giorgos Athineou <gioathineou@gmail.com>.

References

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

alfa, alfainv, profile

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
frechet(x, 0.2)
frechet(x, 1)
```

The Helmert sub-matrix

The Helmert sub-matrix

Description

The Helmert sub-matrix.

Usage

```
helm(n)
```

Arguments

n	A number grater than or equal to 2.

Details

The Helmert sub-matrix is returned. It is an orthogonal matrix without the first row.

Value

A $(n-1) \times n$ matrix.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>.

References

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

John Aitchison (2003). The Statistical Analysis of Compositional Data, p. 99. Blackburn Press.

Lancaster H. O. (1965). The Helmert matrices. The American Mathematical Monthly 72(1): 4-12.

See Also

alfa, alfainv

Examples

```
helm(3)
helm(5)
```

`The k-nearest neighbours using the alpha-distance`

The k-nearest neighbours using the α-distance

Description

The k-nearest neighbours using the α-distance.

Usage

```
alfann(xnew, x, a, k = 10, rann = FALSE)
```

Arguments

`xnew`	A matrix or a vector with new compositional data.
`x`	A matrix with the compositional data.
`a`	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$, the isometric log-ratio transformation is applied.
`k`	The number of nearest neighbours to search for.
`rann`	If you have large scale datasets and want a faster k-NN search, you can use kd-trees implemented in the R package "Rnanoflann". In this case you must set this argument equal to TRUE. Note however, that in this case, the only available distance is by default "euclidean".

Details

The α-transformation is applied to the compositional data first and the indices of the k-nearest neighbours using the Euclidean distance are returned.

Value

A matrix including the indices of the nearest neighbours of each xnew from x.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

MTsagris M., Alenazi A. and Stewart C. (2023). Flexible non-parametric regression models for compositional response data with zeros. Statistics and Computing, 33(106).

https://link.springer.com/article/10.1007/s11222-023-10277-5

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain.

https://arxiv.org/pdf/1106.1451.pdf

See Also

alfa.knn, comp.nb, alfa.rda, alfa.nb,link{aknn.reg}, alfa, alfainv

Examples

```
library(MASS)
xnew <- as.matrix(fgl[1:20, 2:9])
xnew <- xnew / rowSums(xnew)
x <- as.matrix(fgl[-c(1:20), 2:9])
x <- x / rowSums(x)
b <- alfann(xnew, x, a = 0.1, k = 10)
```

The k-NN algorithm for compositional data

The k-NN algorithm for compositional data

Description

The k-NN algorithm for compositional data with and without using the power transformation.

Usage

```
comp.knn(xnew, x, ina, a = 1, k = 5,  apostasi = "ESOV", mesos = TRUE)

alfa.knn(xnew, x, ina, a = 1, k = 5, mesos = TRUE,
apostasi = "euclidean", rann = FALSE)

ait.knn(xnew, x, ina, a = 1, k = 5, mesos = TRUE,
apostasi = "euclidean", rann = FALSE)
```

Arguments

`xnew` A matrix with the new compositional data whose group is to be predicted. Zeros are allowed, but you must be careful to choose strictly positive values of α or not to set apostasi= "Ait".

`x` A matrix with the available compositional data. Zeros are allowed, but you must be careful to choose strictly positive values of α or not to set apostasi= "Ait".

`ina` A group indicator variable for the available data.

`a` The value of α. As zero values in the compositional data are allowed, you must be careful to choose strictly positive vcalues of α. You have the option to put a = NULL. In this case, the xnew and x are assumed to be the already α-transformed data.

`k` The number of nearest neighbours to consider. It can be a single number or a vector.

`apostasi` The type of distance to use. For the compk.knn this can be one of the following: "ESOV", "taxicab", "Ait", "Hellinger", "angular" or "CS". See the references for them. For the alfa.knn this can be either "euclidean" or "manhattan".

`mesos` This is used in the non standard algorithm. If TRUE, the arithmetic mean of the distances is calulated, otherwise the harmonic mean is used (see details).

`rann` If you have large scale datasets and want a faster k-NN search, you can use kd-trees implemented in the R package "Rnanoflann". In this case you must set this argument equal to TRUE. Note however, that in this case, the only available distance is by default "euclidean".

Details

The k-NN algorithm is applied for the compositional data. There are many metrics and possibilities to choose from. The algorithm finds the k nearest observations to a new observation and allocates it to the class which appears most times in the neighbours. It then computes the arithmetic or the harmonic mean of the distances. The new point is allocated to the class with the minimum distance.

Value

A vector with the estimated groups.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Giorgos Athineou <gioathineou@gmail.com>.

References

Tsagris, Michail (2014). The k-NN algorithm for compositional data: a revised approach with and without zero values present. Journal of Data Science, 12(3): 519–534.

Friedman Jerome, Trevor Hastie and Robert Tibshirani (2009). The elements of statistical learning, 2nd edition. Springer, Berlin

Tsagris Michail, Simon Preston and Andrew T.A. Wood (2016). Improved classification for compositional data using the α-transformation. Journal of Classification 33(2): 243–261.

Connie Stewart (2017). An approach to measure distance between compositional diet estimates containing essential zeros. Journal of Applied Statistics 44(7): 1137–1152.

Clarotto L., Allard D. and Menafoglio A. (2022). A new class of α-transformations for the spatial analysis of Compositional Data. Spatial Statistics, 47.

Endres, D. M. and Schindelin, J. E. (2003). A new metric for probability distributions. Information Theory, IEEE Transactions on 49, 1858–1860.

Osterreicher, F. and Vajda, I. (2003). A new class of metric divergences on probability spaces and its applicability in statistics. Annals of the Institute of Statistical Mathematics 55, 639–653.

See Also

compknn.tune, alfa.rda, comp.nb, alfa.nb, alfa,esov, mix.compnorm

Examples

```
x <- as.matrix( iris[, 1:4] )
x <- x/ rowSums(x)
ina <- iris[, 5]
mod <- comp.knn(x, x, ina, a = 1, k = 5)
table(ina, mod)
mod2 <- alfa.knn(x, x, ina, a = 1, k = 5)
table(ina, mod2)
```

The multiplicative log-ratio transformation and its inverse

The multiplicative log-ratio transformation and its inverse

Description

The multiplicative log-ratio transformation and its inverse.

Usage

```
mlr(x)
mlrinv(y)
```

Arguments

x	A numerical matrix with the compositional data.
y	A numerical matrix with data to be closed into the simplex.

Details

The multiplicative log-ratio transformation and its inverse are applied here. This means that no zeros are allowed.

Value

A matrix with the mlr transformed data (if mlr is used) or with the compositional data (if the mlrinv is used).

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

alr, pivot, green, alfa

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y <- mlr(x)
x1 <- mlrinv(y)
```

The pivot coordinate transformation and its inverse

The pivot coordinate transformation and its inverse

Description

The pivot coordinate transformation and its inverse.

Usage

```
pivot(x)
pivotinv(y)
```

Arguments

x	A numerical matrix with the compositional data.
y	A numerical matrix with data to be closed into the simplex.

Details

The pivot coordinate transformation and its inverse are computed. This means that no zeros are allowed.

Value

A matrix with the alr transformed data (if pivot is used) or with the compositional data (if the pivotinv is used).

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Peter Filzmoser, Karel Hron and Matthias Templ (2018). Applied Compositional Data Analysis With Worked Examples in R (pages 49 and 51). Springer.

See Also

alfa, alfainv, alr, green

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y <- pivot(x)
x1 <- alrinv(y)
```

The SCLS model   *Simplicial constrained linear least squares (SCLS) for compositional responses and predictors*

Description

Simplicial constrained linear least squares (SCLS) for compositional responses and predictors.

Usage

```
scls(y, x, xnew = NULL, nbcores = 4)
```

Arguments

y	A matrix with the compositional data (dependent variable). Zero values are allowed. It may also by a big matrix of the FBM class.
x	A matrix with the compositional predictors. Zero values are allowed. It may also by a big matrix of the FBM class.
xnew	If you have new data use it, otherwise leave it NULL.
nbcores	The number of cores to use in the case of an FBM class (big) matrix. If you do not know how many to cores to use, you may try the command **nb_cores()** from the **bigparallelr** package.

Details

The function performs least squares regression where the beta coefficients are constained to be positive and sum to 1. We were inspired by the transformation-free linear regression for compositional responses and predictors of Fiksel, Zeger and Datta (2022). Our implementation now uses quadratic programming instead of the function `optim`, and the solution is more accurate and extremely fast.

Big matrices, of FBM class, are now accepted.

Value

A list including:

`mse`	The mean squared error.
`be`	The beta coefficients.
`est`	The fitted of xnew if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

Fiksel J., Zeger S. and Datta A. (2022). A transformation-free linear regression for compositional outcomes and predictors. Biometrics, 78(3): 974–987.

See Also

`cv.scls`, `tflr`, `scls.indeptest`, `scrq`

Examples

```
library(MASS)
set.seed(1234)
y <- rdiri(214, runif(4, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- scls(y, x)
mod
```

`The SCLS model with multiple compositional predictors`

The SCLS model with multiple compositional predictors

Description

The SCLS model with multiple compositional predictors.

Usage

```
scls2(y, x, wei = FALSE, xnew = NULL)
```

Arguments

`y`	A matrix with the compositional data (dependent variable). Zero values are allowed.
`x`	A list of matrices with the compositional predictors. Zero values are allowed.
`wei`	Do you want weights among the different simplicial predictors? The default is FALSE.
`xnew`	If you have new data use it, otherwise leave it NULL.

Details

The function performs least squares regression where the beta coefficients are constained to be positive and sum to 1. We were inspired by the transformation-free linear regression for compositional responses and predictors of Fiksel, Zeger and Datta (2020). Our implementation now uses quadratic programming instead of the function `optim`, and the solution is more accurate and extremely fast. This function allows for more than one simplicial predictors and offers the possibility of assigning weights to each simplicial predictor.

Value

A list including:

`ini.mse`	The mean squared error when all simplicial predictors carry equal weight.
`ini.be`	The beta coefficients when all simplicial predictors carry equal weight.
`mse`	The mean squared error when the simplicial predictors carry unequal weights.
`weights`	The weights in a vector form. A vector of length equal to the number of rows of the matrix of coefficients.
`am`	The vector of weights, one for each simplicia predictor. The length of the vector is equal to the number of simplicial predictors.
`est`	The fitted of xnew if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

cv.scls, tflr, scls.indeptest

Examples

```
library(MASS)
set.seed(1234)
y <- rdiri(214, runif(4, 1, 3))
x1 <- as.matrix(fgl[, 2:9])
x <- list()
x[[ 1 ]] <- x1 / rowSums(x1)
x[[ 2 ]] <- Compositional::rdiri(214, runif(4))
mod <- scls2(y, x)
mod
```

`The TFLR model with multiple compositional predictors`

The TFLR model with multiple compositional predictors

Description

The TFLR model with multiple compositional predictors

Usage

```
tflr2(y, x, wei = FALSE, xnew = NULL)
```

Arguments

y	A matrix with the compositional data (dependent variable). Zero values are allowed.
x	A list of matrices with the compositional predictors. Zero values are allowed.
wei	Do you want weights among the different simplicial predictors? The default is FALSE.
xnew	If you have new data use it, otherwise leave it NULL.

Details

The transformation-free linear regression for compositional responses and predictors is implemented. The function to be minized is $-\sum_{i=1}^n y_i \log y_i/(X_i B)$. This is a self implementation of the function that can be found in the package codalm. This function allows for more than one simplicial predictors and offers the possibility of assigning weights to each simplicial predictor.

Value

A list including:

`ini.mse`	The mean squared error when all simplicial predictors carry equal weight.
`ini.be`	The beta coefficients when all simplicial predictors carry equal weight.
`mse`	The mean squared error when the simplicial predictors carry unequal weights.
`weights`	The weights in a vector form. A vector of length equal to the number of rows of the matrix of coefficients.
`am`	The vector of weights, one for each simplicia predictor. The length of the vector is equal to the number of simplicial predictors.
`est`	The fitted of xnew if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Fiksel J., Zeger S. and Datta A. (2022). A transformation-free linear regression for compositional outcomes and predictors. Biometrics, 78(3): 974–987.

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

cv.scls, tflr, scls.indeptest

Examples

```
library(MASS)
set.seed(1234)
y <- rdiri(214, runif(4, 1, 3))
x1 <- as.matrix(fgl[, 2:9])
x <- list()
x[[ 1 ]] <- x1 / rowSums(x1)
x[[ 2 ]] <- Compositional::rdiri(214, runif(4))
mod <- tflr2(y, x)
mod
```

`The transformation-free linear regression (TFLR) for compositional responses and predictors`

Transformation-free linear regression (TFLR) for compositional responses and predictors

Description

Transformation-free linear regression (TFLR) for compositional responses and predictors.

Usage

```
tflr(y, x, xnew = NULL)
```

Arguments

`y`	A matrix with the compositional response. Zero values are allowed.
`x`	A matrix with the compositional predictors. Zero values are in general allowed, but there can be cases when these are problematic.
`xnew`	If you have new data use it, otherwise leave it NULL.

Details

The transformation-free linear regression for compositional responses and predictors is implemented. The function to be minized is $-\sum_{i=1}^n y_i \log y_i/(X_iB)$. This is an efficient self implementation.

Value

A list including:

`kl`	The Kullback-Leibler divergence between the observed and the fitted response compositional data.
`be`	The beta coefficients.
`est`	The fitted values of xnew if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Fiksel J., Zeger S. and Datta A. (2022). A transformation-free linear regression for compositional outcomes and predictors. Biometrics, 78(3): 974–987.

Tsagris. M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(27).

See Also

cv.tflr, scls kl.alfapcr

Examples

```
library(MASS)
y <- rdiri(214, runif(3, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- tflr(y, x, x)
mod
```

Total variability   *Total variability*

Description

Total variability.

Usage

```
totvar(x, a = 0)
```

Arguments

x	A numerical matrix with the compositional data.
a	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the centred log-ratio transformation is used.

Details

The α-transformation is applied and the sum of the variances of the transformed variables is calculated. This is the total variability. Aitchison (1986) used the centred log-ratio transformation, but we have extended it to cover more geometries, via the α-transformation.

Value

The total variability of the data in a given geometry as dictated by the value of α.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

alfa, \ link{alfainv,} alfa.profile, alfa.tune

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
totvar(x)
```

Tuning of the alpha-generalised correlations between two compositional datasets

Tuning of the α-generalised correlations between two compositional datasets

---

Description

Tuning of the $alpha$-generalised correlations between two compositional datasets.

Usage

```
acor.tune(y, x, a = c(-1, 1), type = "dcor")
```

Arguments

y	A matrix with the compositional data.
x	A matrix with the compositional data.
a	The range of values of the power transformation to search for the optimal one. If zero values are present it has to be greater than 0.
type	the type of correlation to compute, the distance correlation ("dcor"), the canonical correlation type 1 ("cancor1") or the canonical correlation type 2 ("cancor2"). See details for more information.

Details

The α-transformation is applied to each composition and then, if type="dcor" the distance correlation or the canonical correlation is computed. If type = "cancor1" the function returns the value of α that maximizes the product of the eigenvalues. If type = "cancor2" the function returns the value of α that maximizes the the largest eigenvalue.

Value

A list including:

alfa	The optimal value of α.
acor	The maximum value of the acor.
runtime	The runtime of the optimization

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

Tsagris M. and Papadakis M. (2025). Fast and light-weight energy statistics using the R package Rfast. https://arxiv.org/abs/2501.02849v2

See Also

acor, alfa.profile, alfa, alfainv

Examples

```
y <- rdiri(30, runif(3) )
x <- rdiri(30, runif(3) )
acor.tune(y, x)
```

Tuning of the bandwidth h of the kernel using the maximum likelihood cross validation

Tuning of the bandwidth h of the kernel using the maximum likelihood cross validation

---

Description

Tuning of the bandwidth h of the kernel using the maximum likelihood cross validation.

Usage

```
mkde.tune( x, low = 0.1, up = 3, s = cov(x) )
```

Arguments

x	A matrix with Euclidean (continuous) data.
low	The minimum value to search for the optimal bandwidth value.
up	The maximum value to search for the optimal bandwidth value.
s	A covariance matrix. By default it is equal to the covariance matrix of the data, but can change to a robust covariance matrix, MCD for example.

Details

Maximum likelihood cross validation is applied in order to choose the optimal value of the bandwidth parameter. No plot is produced.

Value

A list including:

`hopt`	The optimal bandwidth value.
`maximum`	The value of the pseudo-log-likelihood at that given bandwidth value.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Giorgos Athineou <gioathineou@gmail.com>.

References

Arsalane Chouaib Guidoum (2015). Kernel Estimator and Bandwidth Selection for Density and its Derivatives. The kedd R package. http://cran.r-project.org/web/packages/kedd/vignettes/kedd.pdf

M.P. Wand and M.C. Jones (1995). Kernel smoothing, pages 91-92.

See Also

`mkde`, `comp.kerncontour`

Examples

```
library(MASS)
mkde.tune(as.matrix(iris[, 1:4]), c(0.1, 3) )
```

`Tuning of the divergence based regression for compositional data with compositional data in the covariates side using the alpha-transformation`

Tuning of the divergence based regression for compositional data with compositional data in the covariates side using the α-transformation

---

Description

Tuning of the divergence based regression for compositional data with compositional data in the covariates side using the α-transformation.

Usage

```
klalfapcr.tune(y, x, covar = NULL, nfolds = 10, maxk = 50, a = seq(-1, 1, by = 0.1),
folds = NULL, graph = FALSE, tol = 1e-07, maxiters = 50, seed = NULL)
```

Arguments

`y`	A numerical matrix with compositional data with or without zeros.
`x`	A matrix with the predictor variables, the compositional data. Zero values are allowed.
`covar`	If you have other continuous covariates put themn here.
`nfolds`	The number of folds for the K-fold cross validation, set to 10 by default.
`maxk`	The maximum number of principal components to check.
`a`	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
`folds`	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
`graph`	If graph is TRUE (default value) a plot will appear.
`tol`	The tolerance value to terminate the Newton-Raphson procedure.
`maxiters`	The maximum number of Newton-Raphson iterations.
`seed`	You can specify your own seed number here or leave it NULL.

Details

The M-fold cross validation is performed in order to select the optimal values for α and k, the number of principal components. The α-transformation is applied to the compositional data first, the first k principal component scores are calcualted and used as predictor variables for the Kullback-Leibler divergence based regression model. This procedure is performed M times during the M-fold cross validation.

Value

A list including:

`mspe`	A list with the KL divergence for each value of α and k in every fold.
`performance`	A matrix with the KL divergence for each value of α averaged over all folds. If graph is set to TRUE this matrix is plotted.
`best.perf`	The minimum KL divergence.
`params`	The values of α and k corresponding to the minimum KL divergence.

Author(s)

Initial code by Abdulaziz Alenazi. Modifications by Michail Tsagris.

R implementation and documentation: Abdulaziz Alenazi <`a.alenazi@nbu.edu.sa`> and Michail Tsagris <`mtsagris@uoc.gr`>.

References

Alenazi A. (2019). Regression for compositional data with compositional data as predictor variables with or without zero values. Journal of Data Science, 17(1): 219–238. https://jds-online.org/journal/JDS/article/136/file/pdf

Tsagris M. (2015). Regression analysis with compositional data containing zero values. Chilean Journal of Statistics, 6(2): 47–57. http://arxiv.org/pdf/1508.01913v1.pdf

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. http://arxiv.org/pdf/1106.1451.pdf

See Also

`kl.alfapcr`, `cv.tflr`, `glm.pcr`, `alfapcr.tune`

Examples

```
library(MASS)
y <- rdiri( 214, runif(4, 1, 3) )
x <- as.matrix( fgl[, 2:9] )
x <- x / rowSums(x)
mod <- klalfapcr.tune(y = y, x = x, a = c(0.7, 0.8) )
mod
```

`Tuning of the k-NN algorithm for compositional data`

Tuning of the k-NN algorithm for compositional data

Description

Tuning of the k-NN algorithm for compositional data with and without using the power or the α-transformation. In addition, estimation of the rate of correct classification via K-fold cross-validation.

Usage

```
compknn.tune(x, ina, nfolds = 10, k = 2:5, mesos = TRUE,
a = seq(-1, 1, by = 0.1), apostasi = "ESOV", folds = NULL,
stratified = TRUE, seed = NULL, graph = FALSE)

alfaknn.tune(x, ina, nfolds = 10, k = 2:5, mesos = TRUE,
a = seq(-1, 1, by = 0.1), apostasi = "euclidean", rann = FALSE,
folds = NULL, stratified = TRUE, seed = NULL, graph = FALSE)

aitknn.tune(x, ina, nfolds = 10, k = 2:5, mesos = TRUE,
a = seq(-1, 1, by = 0.1), apostasi = "euclidean", rann = FALSE,
folds = NULL, stratified = TRUE, seed = NULL, graph = FALSE)
```

Arguments

`x` A matrix with the available compositional data. Zeros are allowed, but you must be careful to choose strictly positive values of α or not to set apostasi= "Ait".

`ina` A group indicator variable for the available data.

`nfolds` The number of folds to be used. This is taken into consideration only if the folds argument is not supplied.

`k` A vector with the nearest neighbours to consider.

`mesos` This is used in the non standard algorithm. If TRUE, the arithmetic mean of the distances is calculated, otherwise the harmonic mean is used (see details).

`a` A grid of values of α to be used only if the distance chosen allows for it.

`apostasi` The type of distance to use. For the compk.knn this can be one of the following: "ESOV", "taxicab", "Ait", "Hellinger", "angular" or "CS". See the references for them. For the alfa.knn this can be either "euclidean" or "manhattan".

`rann` If you have large scale datasets and want a faster k-NN search, you can use kd-trees implemented in the R package "Rnanoflann". In this case you must set this argument equal to TRUE. Note however, that in this case, the only available distance is by default "euclidean".

`folds` If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.

`stratified` Do you want the folds to be created in a stratified way? TRUE or FALSE.

`seed` You can specify your own seed number here or leave it NULL.

`graph` If set to TRUE a graph with the results will appear.

Details

The k-NN algorithm is applied for the compositional data. There are many metrics and possibilities to choose from. The algorithm finds the k nearest observations to a new observation and allocates it to the class which appears most times in the neighbours.

Value

A list including:

`per` A matrix or a vector (depending on the distance chosen) with the averaged over all folds rates of correct classification for all hyper-parameters (α and k).

`performance` The estimated rate of correct classification.

`best_a` The best value of α. This is returned for "ESOV" and "taxicab" only.

`best_k` The best number of nearest neighbours.

`runtime` The run time of the cross-validation procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>.

References

Tsagris, Michail (2014). The k-NN algorithm for compositional data: a revised approach with and without zero values present. Journal of Data Science, 12(3): 519–534. https://arxiv.org/pdf/1506.05216.pdf

Friedman Jerome, Trevor Hastie and Robert Tibshirani (2009). The elements of statistical learning, 2nd edition. Springer, Berlin

Tsagris M., Preston S. and Wood A.T.A. (2016). Improved classification for compositional data using the α-transformation. Journal of Classification, 33(2): 243–261. http://arxiv.org/pdf/1106.1451.pdf

Connie Stewart (2017). An approach to measure distance between compositional diet estimates containing essential zeros. Journal of Applied Statistics 44(7): 1137–1152.

Clarotto L., Allard D. and Menafoglio A. (2022). A new class of α-transformations for the spatial analysis of Compositional Data. Spatial Statistics, 47.

Endres, D. M. and Schindelin, J. E. (2003). A new metric for probability distributions. Information Theory, IEEE Transactions on 49, 1858–1860.

Osterreicher, F. and Vajda, I. (2003). A new class of metric divergences on probability spaces and its applicability in statistics. Annals of the Institute of Statistical Mathematics 55, 639–653.

See Also

`comp.knn`, `alfarda.tune`, `cv.dda`, `cv.compnb`

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x/ rowSums(x)
ina <- iris[, 5]
mod1 <- compknn.tune(x, ina, a = seq(1, 1, by = 0.1) )
mod2 <- alfaknn.tune(x, ina, a = seq(-1, 1, by = 0.1) )
```

Tuning of the projection pursuit regression for compositional data

Tuning of the projection pursuit regression for compositional data

---

Description

Tuning of the projection pursuit regression for compositional data.

Usage

```
compppr.tune(y, x, nfolds = 10, folds = NULL, seed = NULL,
nterms = 1:10, type = "alr", yb = NULL )
```

Arguments

`y` A matrix with the available compositional data, but zeros are not allowed.

`x` A matrix with the continuous predictor variables.

`nfolds` The number of folds to use.

`folds` If you have the list with the folds supply it here.

`seed` You can specify your own seed number here or leave it NULL.

`nterms` The number of terms to try in the projection pursuit regression.

`type` Either "alr" or "ilr" corresponding to the additive or the isometric log-ratio transformation respectively.

`yb` If you have already transformed the data using a log-ratio transformation put it here. Othewrise leave it NULL.

Details

The function performs tuning of the projection pursuit regression algorithm.

Value

A list including:

`kl` The average Kullback-Leibler divergence.

`perf` The average Kullback-Leibler divergence.

`runtime` The run time of the cross-validation procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression. Journal of the American Statistical Association, 76, 817-823. doi: 10.2307/2287576.

See Also

`comp.ppr`, `aknnreg.tune`, `akernreg.tune`

Examples

```
y <- as.matrix(iris[, 1:3])
y <- y/ rowSums(y)
x <- iris[, 4]
mod <- compppr.tune(y, x)
```

`Tuning of the projection pursuit regression with compositional predictor variables`

Tuning of the projection pursuit regression with compositional predictor variables

Description

Tuning of the projection pursuit regression with compositional predictor variables.

Usage

```
pprcomp.tune(y, x, nfolds = 10, folds = NULL, seed = NULL,
nterms = 1:10, type = "log", graph = FALSE)
```

Arguments

`y`	A numerical vector with the continuous variable.
`x`	A matrix with the available compositional data, but zeros are not allowed.
`nfolds`	The number of folds to use.
`folds`	If you have the list with the folds supply it here.
`seed`	You can specify your own seed number here or leave it NULL.
`nterms`	The number of terms to try in the projection pursuit regression.
`type`	Either "alr" or "log" corresponding to the additive log-ratio transformation or the logarithm applied to the compositional predictor variables.
`graph`	If graph is TRUE (default value) a filled contour plot will appear.

Details

The function performs tuning of the projection pursuit regression algorithm with compositional predictor variables.

Value

A list including:

`runtime`	The run time of the cross-validation procedure.
`mse`	The mean squared error of prediction for each number of terms.
`opt.nterms`	The number of terms corresponding to the minimum mean squared error of prediction.
`opt.alpha`	The value of α corresponding to the minimum mean squared error of prediction.
`performance`	The minimum mean squared error of prediction.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression. Journal of the American Statistical Association, 76, 817-823. doi: 10.2307/2287576.

See Also

pprcomp, ice.pprcomp, alfapcr.tune, compppr.tune

Examples

```
x <- as.matrix(iris[, 2:4])
x <- x/ rowSums(x)
y <- iris[, 1]
mod <- pprcomp.tune(y, x)
```

Tuning of the projection pursuit regression with compositional predictor variables using the alpha-transformation

Tuning of the projection pursuit regression with compositional predictor variables using the α-transformation

---

Description

Tuning of the projection pursuit regression with compositional predictor variables using the α-transformation.

Usage

```
alfapprcomp.tune(y, x, nfolds = 10, folds = NULL, seed = NULL,
nterms = 1:10, a = seq(-1, 1, by = 0.1), graph = FALSE)
```

Arguments

y	A numerical vector with the continuous variable.
x	A matrix with the available compositional data. Zeros are allowed.
nfolds	The number of folds to use.
folds	If you have the list with the folds supply it here.
seed	You can specify your own seed number here or leave it NULL.
nterms	The number of terms to try in the projection pursuit regression.
a	A vector with the values of α for the α-transformation.
graph	If graph is TRUE (default value) a filled contour plot will appear.

Details

The function performs tuning of the projection pursuit regression algorithm with compositional predictor variables using the α-transformation.

Value

A list including:

`runtime` The run time of the cross-validation procedure.

`mse` The mean squared error of prediction for each number of terms.

`opt.nterms` The number of terms corresponding to the minimum mean squared error of prediction.

`opt.alpha` The value of α corresponding to the minimum mean squared error of prediction.

`performance` The minimum mean squared error of prediction.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression. Journal of the American Statistical Association, 76, 817-823. doi: 10.2307/2287576.

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

`alfa.pprcomp`, `pprcomp.tune`, `compppr.tune`

Examples

```
x <- as.matrix(iris[, 2:4])
x <- x / rowSums(x)
y <- iris[, 1]
mod <- alfapprcomp.tune( y, x, a = c(0, 0.5, 1) )
```

```
Tuning the number of PCs in the PCR with compositional data using
the alpha-transformation
```

Tuning the number of PCs in the PCR with compositional data using the α-transformation

Description

This is a cross-validation procedure to decide on the number of principal components when using regression with compositional data (as predictor variables) using the α-transformation.

Usage

```
alfapcr.tune(y, x, model = "gaussian", nfolds = 10, maxk = 50, a = seq(-1, 1, by = 0.1),
folds = NULL, ncores = 1, graph = TRUE, col.nu = 15, seed = NULL)
```

Arguments

`y` A vector with either continuous, binary or count data.

`x` A matrix with the predictor variables, the compositional data. Zero values are allowed.

`model` The type of regression model to fit. The possible values are "gaussian", "binomial" and "poisson".

`nfolds` The number of folds for the K-fold cross validation, set to 10 by default.

`maxk` The maximum number of principal components to check.

`a` A vector with a grid of values of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.

`folds` If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.

`ncores` How many cores to use. If you have heavy computations or do not want to wait for long time more than 1 core (if available) is suggested. It is advisable to use it if you have many observations and or many variables, otherwise it will slow down th process.

`graph` If graph is TRUE (default value) a filled contour plot will appear.

`col.nu` A number parameter for the filled contour plot, taken into account only if graph is TRUE.

`seed` You can specify your own seed number here or leave it NULL.

Details

The α-transformation is applied to the compositional data first and the function "pcr.tune" or "glmpcr.tune" is called.

Value

If graph is TRUE a filled contour will appear. A list including:

`mspe`	The MSPE where rows correspond to the α values and the columns to the number of principal components.
`best.par`	The best pair of α and number of principal components.
`performance`	The minimum mean squared error of prediction.
`runtime`	The time required by the cross-validation procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris M. (2015). Regression analysis with compositional data containing zero values. Chilean Journal of Statistics, 6(2): 47-57. https://arxiv.org/pdf/1508.01913v1.pdf

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

Jolliffe I.T. (2002). Principal Component Analysis.

See Also

`alfa`, `profile`, `alfa.pcr`, `pcr.tune`, `glmpcr.tune`, `glm`

Examples

```
library(MASS)
y <- as.vector(fgl[, 1])
x <- as.matrix(fgl[, 2:9])
x <- x/ rowSums(x)
mod <- alfapcr.tune(y, x, nfolds = 10, maxk = 50, a = seq(-1, 1, by = 0.1) )
```

`Tuning the parameters of the regularised discriminant analysis`
Tuning the parameters of the regularised discriminant analysis

---

Description

Tuning the parameters of the regularised discriminant analysis for Eucldiean data.

Usage

```
rda.tune(x, ina, nfolds = 10, gam = seq(0, 1, by = 0.1), del = seq(0, 1, by = 0.1),
ncores = 1, folds = NULL, stratified = TRUE, seed = NULL)
```

Arguments

`x`	A matrix with the data.
`ina`	A group indicator variable for the avaiable data.
`nfolds`	The number of folds in the cross validation.
`gam`	A grid of values for the γ parameter as defined in Tsagris et al. (2016).
`del`	A grid of values for the δ parameter as defined in Tsagris et al. (2016).
`ncores`	The number of cores to use. If more than 1, parallel computing will take place. It is advisable to use it if you have many observations and or many variables, otherwise it will slow down th process.
`folds`	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
`stratified`	Do you want the folds to be created in a stratified way? TRUE or FALSE.
`seed`	You can specify your own seed number here or leave it NULL.

Details

Cross validation is performed to select the optimal parameters for the regularisded discriminant analysis and also estimate the rate of accuracy.

The covariance matrix of each group is calcualted and then the pooled covariance matrix. The spherical covariance matrix consists of the average of the pooled variances in its diagonal and zeros in the off-diagonal elements. gam is the weight of the pooled covariance matrix and 1-gam is the weight of the spherical covariance matrix, Sa = gam * Sp + (1-gam) * sp. Then it is a compromise between LDA and QDA. del is the weight of Sa and 1-del the weight of each group covariance group.

Value

A list including: If graph is TRUE a plot of a heatmap of the performance s will appear.

`per`	An array with the estimate rate of correct classification for every fold. For each of the M matrices, the row values correspond to gam and the columns to the del parameter.
`percent`	A matrix with the mean estimated rates of correct classification. The row values correspond to gam and the columns to the del parameter.
`se`	A matrix with the standard error of the mean estimated rates of correct classification. The row values correspond to gam and the columns to the del parameter.
`result`	The estimated rate of correct classification along with the best gam and del parameters.
`runtime`	The time required by the cross-validation procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Friedman J.H. (1989): Regularized Discriminant Analysis. Journal of the American Statistical Association 84(405): 165–175.

Friedman Jerome, Trevor Hastie and Robert Tibshirani (2009). The elements of statistical learning, 2nd edition. Springer, Berlin.

Tsagris M., Preston S. and Wood A.T.A. (2016). Improved classification for compositional data using the α-transformation. Journal of Classification, 33(2): 243–261.

See Also

rda

Examples

```
mod <- rda.tune(as.matrix(iris[, 1:4]), iris[, 5], gam = seq(0, 1, by = 0.2),
del = seq(0, 1, by = 0.2) )
mod
```

`Tuning the principal components with GLMs`

Tuning the principal components with GLMs

Description

Tuning the number of principal components in the generalised linear models.

Usage

```
pcr.tune(y, x, nfolds = 10, maxk = 50, folds = NULL, ncores = 1,
seed = NULL, graph = TRUE)

glmpcr.tune(y, x, nfolds = 10, maxk = 10, folds = NULL, ncores = 1,
seed = NULL, graph = TRUE)

multinompcr.tune(y, x, nfolds = 10, maxk = 10, folds = NULL, ncores = 1,
seed = NULL, graph = TRUE)
```

Arguments

`y`	A real valued vector for "pcr.tune". A real valued vector for the "glmpcr.tune" with either two numbers, 0 and 1 for example, for the binomial regression or with positive discrete numbers for the poisson. For the "multinompcr.tune" a vector or a factor with more than just two values. This is a multinomial regression.
`x`	A matrix with the predictor variables, they have to be continuous.
`nfolds`	The number of folds in the cross validation.

`maxk` The maximum number of principal components to check.

`folds` If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.

`ncores` The number of cores to use. If more than 1, parallel computing will take place. It is advisable to use it if you have many observations and or many variables, otherwise it will slow down th process.

`seed` You can specify your own seed number here or leave it NULL.

`graph` If graph is TRUE a plot of the performance for each fold along the values of α will appear.

Details

Cross validation is performed to select the optimal number of principal components in the GLMs or the multinomial regression. This is used by `alfapcr.tune`.

Value

If graph is TRUE a plot of the performance versus the number of principal components will appear. A list including:

`msp` A matrix with the mean deviance of prediction or mean accuracy for every fold.

`mpd` A vector with the mean deviance of prediction or mean accuracy, each value corresponds to a number of principal components.

`k` The number of principal components which minimizes the deviance or maximises the accuracy.

`performance` The optimal performance, MSE for the linea regression, minimum deviance for the GLMs and maximum accuracy for the multinomial regression.

`runtime` The time required by the cross-validation procedure.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aguilera A.M., Escabias M. and Valderrama M.J. (2006). Using principal components for estimating logistic regression with high-dimensional multicollinear data. Computational Statistics & Data Analysis 50(8): 1905-1924.

Jolliffe I.T. (2002). Principal Component Analysis.

See Also

`pcr.tune`, `glm.pcr`, `alfa.pcr`, `alfapcr.tune`

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
y <- rpois(214, 10)
glmpcr.tune(y, x, nfolds = 10, maxk = 20, folds = NULL, ncores = 1)
```

Tuning the value of alpha in the alpha-regression

Tuning the value of α in the α-regression

Description

Tuning the value of α in the α-regression.

Usage

```
alfareg.tune(y, x, a = seq(0.1, 1, by = 0.1), nfolds = 10,
folds = NULL, nc = 1, seed = NULL, graph = FALSE)
```

Arguments

`y`	A matrix with compositional data. zero values are allowed.
`x`	A matrix with the continuous predictor variables or a data frame including categorical predictor variables.
`a`	The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
`nfolds`	The number of folds to split the data.
`folds`	If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
`nc`	The number of cores to use. IF you have a multicore computer it is advisable to use more than 1. It makes the procedure faster. It is advisable to use it if you have many observations and or many variables, otherwise it will slow down th process.
`seed`	You can specify your own seed number here or leave it NULL.
`graph`	If graph is TRUE a plot of the performance for each fold along the values of α will appear.

Details

The α-transformation is applied to the compositional data and the numerical optimisation is performed for the regression, unless $\alpha = 0$, where the coefficients are available in closed form.

Value

A plot of the estimated Kullback-Leibler divergences (multiplied by 2) along the values of α (if graph is set to TRUE). A list including:

`runtime`	The runtime required by the cross-validation.
`kula`	A matrix with twice the Kullback-Leibler divergence of the observed from the fitted values. Each row corresponds to a fold and each column to a value of α. The average over the columns equal the next argument, "kl".
`kl`	A vector with twice the Kullback-Leibler divergence of the observed from the fitted values. Every value corresponds to a value of α.
`opt`	The optimal value of α.
`value`	The minimum value of twice the Kullback-Leibler.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`> and Giorgos Athineou <gioathineou@gmail.com>.

References

Tsagris M. (2015). Regression analysis with compositional data containing zero values. Chilean Journal of Statistics, 6(2): 47-57. https://arxiv.org/pdf/1508.01913v1.pdf

Tsagris M.T., Preston S. and Wood A.T.A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain. https://arxiv.org/pdf/1106.1451.pdf

See Also

`alfa.reg`, `alfa`

Examples

```
library(MASS)
y <- as.matrix(fgl[1:40, 2:4])
y <- y /rowSums(y)
x <- as.vector(fgl[1:40, 1])
mod <- alfareg.tune(y, x, a = seq(0, 1, by = 0.1), nfolds = 5)
```

Two-sample test of high-dimensional means for compositional data

Two-sample test of high-dimensional means for compositional data

Description

Two-sample test of high-dimensional means for compositional data.

Usage

```
hd.meantest2(y1, y2, R = 1)
```

Arguments

y1	A matrix containing the compositional data of the first group.
y2	A matrix containing the compositional data of the second group.
R	If R is 1 no bootstrap calibration is performed and the asymptotic p-value is returned. If R is greater than 1, the bootstrap p-value is returned.

Details

A two sample for high dimensional mean vectors of compositional data is implemented. See references for more details.

Value

A vector with the test statistic value and its associated (bootstrap) p-value.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Cao Y., Lin W. and Li H. (2018). Two-sample tests of high-dimensional means for compositional data. Biometrika, 105(1): 115–132.

See Also

comp.test

Examples

```
m <- runif(200, 10, 15)
x1 <- rdiri(100, m)
x2 <- rdiri(100, m)
hd.meantest2(x1, x2)
```

Unconstrained GLMs with compositional predictor variables
Unconstrained GLMs with compositional predictor variables

Description

Unconstrained GLMs with compositional predictor variables.

Usage

```
ulc.glm(y, x, z = NULL, model = "logistic", xnew = NULL, znew = NULL)
```

Arguments

`y` A numerical vector containing the response variable values. This is either a binary variable or a vector with counts.

`x` A matrix with the predictor variables, the compositional data. No zero values are allowed.

`z` A matrix, data.frame, factor or a vector with some other covariate(s).

`model` For the ulc.glm(), this can be either "logistic" or "poisson".

`xnew` A matrix containing the new compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.

`znew` A matrix, data.frame, factor or a vector with the values of some other covariate(s). If you have no new data, leave this NULL as is by default.

Details

The function performs the unconstrained log-contrast logistic or Poisson regression model. The logarithm of the compositional predictor variables is used (hence no zero values are allowed). The response variable is linked to the log-transformed data **without** the constraint that the sum of the regression coefficients equals 0. If you want the regression without the zum-to-zero contraints see `lc.glm`. Extra predictors variables are allowed as well, for instance categorical or continuous.

Value

A list including:

`devi` The residual deviance of the logistic or Poisson regression model.

`be` The unconstrained regression coefficients. Their sum does not equal 0.

`est` If the arguments "xnew" and znew were given these are the predicted or estimated values, otherwise it is NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Lu J., Shi P., and Li H. (2019). Generalized linear models with linear constraints for microbiome compositional data. Biometrics, 75(1): 235–244.

See Also

lc.glm, lc.glm2, ulc.glm2, lcglm.aov

Examples

```
y <- rbinom(150, 1, 0.5)
x <- rdiri(150, runif(3, 1,3))
mod <- ulc.glm(y, x)
```

Unconstrained linear regression with compositional predictor variables

Unconstrained linear regression with compositional predictor variables

---

Description

Unconstrained linear regression with compositional predictor variables.

Usage

```
ulc.reg(y, x, z = NULL, xnew = NULL, znew = NULL)
```

Arguments

y	A numerical vector containing the response variable values. This must be a continuous variable.
x	A matrix with the predictor variables, the compositional data. No zero values are allowed.
z	A matrix, data.frame, factor or a vector with some other covariate(s).
xnew	A matrix containing the new compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.
znew	A matrix, data.frame, factor or a vector with the values of some other covariate(s). If you have no new data, leave this NULL as is by default.

Details

The function performs the unconstrained log-contrast regression model as opposed to the log-contrast regression described in Aitchison (2003), pg. 84-85. The logarithm of the compositional predictor variables is used (hence no zero values are allowed). The response variable is linked to the log-transformed data **without** the constraint that the sum of the regression coefficients equals 0. If you want the regression model with the zum-to-zero contraints see `lc.reg`. Extra predictors variables are allowed as well, for instance categorical or continuous.

Value

A list including:

`be`	The unconstrained regression coefficients. Their sum does not equal 0.
`covbe`	If covariance matrix of the constrained regression coefficients.
`va`	The estimated regression variance.
`residuals`	The vector of residuals.
`est`	If the arguments "xnew" and "znew" were given these are the predicted or estimated values, otherwise it is NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

See Also

`lc.reg`, `lcreg.aov`, `lc.reg2`, `ulc.reg2`, `alfa.pcr`, `alfa.knn.reg`

Examples

```
y <- iris[, 1]
x <- as.matrix(iris[, 2:4])
x <- x / rowSums(x)
mod1 <- ulc.reg(y, x)
mod2 <- ulc.reg(y, x, z = iris[, 5])
```

`Unconstrained linear regression with multiple compositional predictors`

Unconstrained linear regression with multiple compositional predictors

---

Description

Unconstrained linear regression with multiple compositional predictors.

Usage

```
ulc.reg2(y, x, z = NULL, xnew = NULL, znew = NULL)
```

Arguments

`y`	A numerical vector containing the response variable values. This must be a continuous variable.
`x`	A list with multiple matrices with the predictor variables, the compositional data. No zero values are allowed.
`z`	A matrix, data.frame, factor or a vector with some other covariate(s).
`xnew`	A matrix containing a list with multiple matrices with compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.
`znew`	A matrix, data.frame, factor or a vector with the values of some other covariate(s). If you have no new data, leave this NULL as is by default.

Details

The function performs the unconstrained log-contrast regression model as opposed to the log-contrast regression described in Aitchison (2003), pg. 84-85. The logarithm of the compositional predictor variables is used (hence no zero values are allowed). The response variable is linked to the log-transformed data **without** the constraint that the sum of the regression coefficients equals 0. If you want the regression model with the zum-to-zero contraints see `lc.reg2`. Extra predictors variables are allowed as well, for instance categorical or continuous. Similarly to `lc.reg2` there are multiple compositions treated as predictor variables.

Value

A list including:

`be`	The unconstrained regression coefficients. Their sum for each composition does not equal 0.
`covbe`	If covariance matrix of the constrained regression coefficients.
`va`	The estimated regression variance.
`residuals`	The vector of residuals.

est If the arguments "xnew" and "znew" were given these are the predicted or estimated values, otherwise it is NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Xiaokang Liu, Xiaomei Cong, Gen Li, Kendra Maas and Kun Chen (2020). Multivariate Log-Contrast Regression with Sub-Compositional Predictors: Testing the Association Between Preterm Infants' Gut Microbiome and Neurobehavioral Outcome.

See Also

lc.reg2, ulc.reg, lc.reg, alfa.pcr, alfa.knn.reg

Examples

```
y <- iris[, 1]
x <- list()
x1 <- as.matrix(iris[, 2:4])
x1 <- x1 / rowSums(x1)
x[[ 1 ]] <- x1
x[[ 2 ]] <- rdiri(150, runif(4) )
x[[ 3 ]] <- rdiri(150, runif(5) )
mod <- lc.reg2(y, x)
```

Unconstrained logistic or Poisson regression with multiple compositional predictors

Unconstrained logistic or Poisson regression with multiple compositional predictors

Description

Unconstrained logistic or Poisson regression with multiple compositional predictors.

Usage

```
ulc.glm2(y, x, z = NULL, model = "logistic", xnew = NULL, znew = NULL)
```

Arguments

`y`	A numerical vector containing the response variable values. This is either a binary variable or a vector with counts.
`x`	A list with multiple matrices with the predictor variables, the compositional data. No zero values are allowed.
`z`	A matrix, data.frame, factor or a vector with some other covariate(s).
`model`	This can be either "logistic" or "poisson".
`xnew`	A matrix containing a list with multiple matrices with compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.
`znew`	A matrix, data.frame, factor or a vector with the values of some other covariate(s). If you have no new data, leave this NULL as is by default.

Details

The function performs the unconstrained log-contrast logistic or Poisson regression model. The logarithm of the compositional predictor variables is used (hence no zero values are allowed). The response variable is linked to the log-transformed data **without** the constraint that the sum of the regression coefficients equals 0. If you want the regression without the zum-to-zero contraints see `lc.glm2`. Extra predictors variables are allowed as well, for instance categorical or continuous.

Value

A list including:

`devi`	The residual deviance of the logistic or Poisson regression model.
`be`	The unconstrained regression coefficients. Their sum does not equal 0.
`est`	If the arguments "xnew" and znew were given these are the predicted or estimated values, otherwise it is NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Lu J., Shi P., and Li H. (2019). Generalized linear models with linear constraints for microbiome compositional data. Biometrics, 75(1): 235–244.

See Also

`lc.glm2`, `ulc.glm`, `lc.glm`

Examples

```
y <- rbinom(150, 1, 0.5)
x <- list()
x1 <- as.matrix(iris[, 2:4])
x1 <- x1 / rowSums(x1)
x[[ 1 ]] <- x1
x[[ 2 ]] <- rdiri(150, runif(4) )
x[[ 3 ]] <- rdiri(150, runif(5) )
mod <- ulc.glm2(y, x)
```

`Unconstrained quantile regression with compositional predictor variables` *Unconstrained quantile regression with compositional predictor variables*

Description

Unconstrained quantile regression with compositional predictor variables.

Usage

```
ulc.rq(y, x, z = NULL, tau = 0.5, xnew = NULL, znew = NULL)
```

Arguments

`y`	A numerical vector containing the response variable values.
`x`	A matrix with the predictor variables, the compositional data. No zero values are allowed.
`z`	A matrix, data.frame, factor or a vector with some other covariate(s).
`tau`	The quantile to be estimated, a number between 0 and 1.
`xnew`	A matrix containing the new compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.
`znew`	A matrix, data.frame, factor or a vector with the values of some other covariate(s). If you have no new data, leave this NULL as is by default.

Details

The function performs the unconstrained log-contrast quantile regression model. The logarithm of the compositional predictor variables is used (hence no zero values are allowed). The response variable is linked to the log-transformed data **without** the constraint that the sum of the regression coefficients equals 0. If you want the regression without the zum-to-zero contraints see `lc.rq`. Extra predictors variables are allowed as well, for instance categorical or continuous.

Value

A list including:

`mod`	The object as returned by the function quantreg::rq(). This is useful for hypothesis testing purposes.
`be`	The unconstrained regression coefficients. Their sum does not equal 0.
`est`	If the arguments "xnew" and znew were given these are the predicted or estimated values, otherwise it is NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Koenker R. W. and Bassett G. W. (1978). Regression Quantiles, Econometrica, 46(1): 33–50.

Koenker R. W. and d'Orey V. (1987). Algorithm AS 229: Computing Regression Quantiles. Applied Statistics, 36(3): 383–393.

See Also

`lc.glm`, `lc.glm2`, `ulc.glm2`, `lcglm.aov`

Examples

```
y <- rnorm(150)
x <- rdiri(150, runif(3, 1,3))
mod <- ulc.rq(y, x)
```

`Unconstrained quantile regression with multiple compositional predictors`

Unconstrained quantile regression with multiple compositional predictors

---

Description

Unconstrained quantile regression with multiple compositional predictors.

Usage

```
ulc.rq2(y, x, z = NULL, tau = 0.5, xnew = NULL, znew = NULL)
```

Arguments

y	A numerical vector containing the response variable values.
x	A list with multiple matrices with the predictor variables, the compositional data. No zero values are allowed.
z	A matrix, data.frame, factor or a vector with some other covariate(s).
tau	The quantile to be estimated, a number between 0 and 1.
xnew	A matrix containing a list with multiple matrices with compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.
znew	A matrix, data.frame, factor or a vector with the values of some other covariate(s). If you have no new data, leave this NULL as is by default.

Details

The function performs the unconstrained log-contrast quantile regression model. The logarithm of the compositional predictor variables is used (hence no zero values are allowed). The response variable is linked to the log-transformed data **without** the constraint that the sum of the regression coefficients equals 0. If you want the regression without the zum-to-zero contraints see `lc.rq2`. Extra predictors variables are allowed as well, for instance categorical or continuous.

Value

A list including:

mod	The object as returned by the function quantreg::rq(). This is useful for hypothesis testing purposes.
be	The unconstrained regression coefficients. Their sum does not equal 0.
est	If the arguments "xnew" and znew were given these are the predicted or estimated values, otherwise it is NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Aitchison J. (1986). The statistical analysis of compositional data. Chapman & Hall.

Koenker R. W. and Bassett G. W. (1978). Regression Quantiles, Econometrica, 46(1): 33–50.

Koenker R. W. and d'Orey V. (1987). Algorithm AS 229: Computing Regression Quantiles. Applied Statistics, 36(3): 383–393.

See Also

`ulc.rq`, `lc.rq`

Examples

```
y <- rnorm(150)
x <- list()
x1 <- as.matrix(iris[, 2:4])
x1 <- x1 / rowSums(x1)
x[[ 1 ]] <- x1
x[[ 2 ]] <- rdiri(150, runif(4) )
x[[ 3 ]] <- rdiri(150, runif(5) )
mod <- ulc.rq2(y, x)
```

Unit-Weibull regression models for proportions

Unit-Weibull regression models for proportions

Description

Unit-Weibull regression models for proportions.

Usage

```
unitweib.reg(y, x, tau = 0.5)
```

Arguments

`y`	A numerical vector proportions. 0s and 1s are allowed.
`x`	A matrix or a data frame with the predictor variables.
`tau`	The quantile to be used for estimation. The default value is 0.5 yielding the median.

Details

See the reference paper.

Value

A list including:

`loglik`	The loglikelihood of the regression model.
`info`	A matrix with all estimated parameters, their standard error, their Wald-statistic and its associated p-value.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Mazucheli J., Menezes A. F. B., Fernandes L. B., de Oliveira R. P. and Ghitany M. E. (2020). The unit-Weibull distribution as an alternative to the Kumaraswamy distribution for the modeling of quantiles conditional on covariates. Journal of Applied Statistics, 47(6): 954–974.

See Also

propreg, beta.reg

Examples

```
y <- exp( - rweibull(100, 1, 1) )
x <- matrix( rnorm(100 * 2), ncol = 2 )
a <- unitweib.reg(y, x)
```

Zero adjusted Dirichlet regression

Zero adjusted Dirichlet regression

Description

Zero adjusted Dirichlet regression.

Usage

```
zadr(y, x, con = TRUE, B = 1, ncores = 2, xnew = NULL)
zadr2(y, x, con = TRUE, B = 1, ncores = 2, xnew = NULL)
```

Arguments

y	A matrix with the compositional data (dependent variable). The number of observations (vectors) with no zero values should be more than the columns of the predictor variables. Otherwise, the initial values will not be calculated.
x	The predictor variable(s), they can be either continnuous or categorical or both.
con	If this is TRUE (default) then the constant term is estimated, otherwise the model includes no constant term.
B	If B is greater than 1 bootstrap estimates of the standard error are returned. If you set this greater than 1, then you must define the number of clusters in order to run in parallel.
ncores	The number of cores to use when B>1. This is to be used for the case of bootstrap. If B = 1, this is not taken into consideration. If this does not work then you might need to load the doParallel yourselves.
xnew	If you have new data use it, otherwise leave it NULL.

Details

A zero adjusted Dirichlet regression is being fittd. The likelihood conists of two components. The contributions of the non zero compositional values and the contributions of the compositional vectors with at least one zero value. The second component may have many different sub-categories, one for each pattern of zeros. The function "zadr2()" links the covariates to the alpha parameters of the Dirichlet distribution, i.e. it uses the classical parametrization of the distribution. This means, that there is a set of regression parameters for each component.

Value

A list including:

`runtime` The time required by the regression.

`loglik` The value of the log-likelihood.

`phi` The precision parameter.

`be` The beta coefficients.

`seb` The standard error of the beta coefficients.

`sigma` Th covariance matrix of the regression parameters (for the mean vector and the phi parameter).

`est` The fitted or the predicted values (if xnew is not NULL).

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <`mtsagris@uoc.gr`>.

References

Tsagris M. and Stewart C. (2018). A Dirichlet regression model for compositional data with zeros. Lobachevskii Journal of Mathematics,39(3): 398–412.

Preprint available from https://arxiv.org/pdf/1410.5011.pdf

See Also

`zad.est`, `diri.reg`, `kl.compreg`, `ols.compreg`, `alfa.reg`

Examples

```
x <- as.vector(iris[, 4])
y <- as.matrix(iris[, 1:3])
y <- y / rowSums(y)
mod1 <- diri.reg(y, x)
y[sample(1:450, 15) ] <- 0
mod2 <- zadr(y, x)
```

Index