

Package ‘Compositional’

June 4, 2018

Type Package

Title Compositional Data Analysis

Version 3.1

URL

Date 2018-06-02

Author Michail Tsagris [aut, cre],
Giorgos Athineou [aut]

Maintainer Michail Tsagris <mtsagris@yahoo.gr>

Depends R (>= 3.2.2)

Imports sn, mixture, doParallel, stats, parallel, foreach, MASS,
fields, emplik, Rfast

Description Regression, classification, contour plots, hypothesis testing and fitting of distributions for compositional data are some of the functions included. The standard textbook for such data is John Aitchison's (1986). ``The statistical analysis of compositional data''. Chapman & Hall.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2018-06-04 10:27:53 UTC

R topics documented:

Compositional-package	3
Contour plot of a Dirichlet distribution in S^2	4
Contour plot of a Gaussian mixture model in S^2	5
Contour plot of the kernel density estimate in S^2	6
Contour plot of the normal distribution in S^2	7
Contour plot of the skew skew-normal distribution in S^2	8
Contour plot of the t distribution in S^2	9
Cross validation for the regularised discriminant analysis with compositional data using the alpha-transformation	11

Cross validation for the ridge regression	12
Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation	14
Density values of a Dirichlet distribution	15
Dirichlet random values simulation	17
Dirichlet regression	18
Divergence based regression for compositional data	19
Empirical likelihood for a one sample mean vector hypothesis testing	21
Empirical likelihood hypothesis testing for two mean vectors	22
Estimating location and scatter parameters for compositional data	24
Estimation of the value of alpha in the folded model	25
Estimation of the value of alpha via the profile log-likelihood	26
Exponential empirical likelihood for a one sample mean vector hypothesis testing	28
Exponential empirical likelihood hypothesis testing for two mean vectors	29
Fast estimation of the value of alpha	31
Fitting a Dirichlet distribution	32
Fitting a Dirichlet distribution via Newton-Raphson	33
Gaussian mixture models for compositional data	35
Helper functions for the Kullback-Leibler regression	37
Hotelling's multivariate version of the 1 sample t-test for Euclidean data	38
Hotelling's multivariate version of the 2 sample t-test for Euclidean data	39
Hypothesis testing for two or more compositional mean vectors	41
Inverse of the alpha-transformation	42
James multivariate version of the t-test	44
Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distri- butions	45
Log-likelihood ratio test for a Dirichlet mean vector	46
Log-likelihood ratio test for a symmetric Dirichlet distribution	47
Mixture model selection via BIC	49
MLE for the multivariate t distribution	50
MLE of the folded model for a given value of alpha	51
Multivariate analysis of variance	52
Multivariate analysis of variance (James test)	53
Multivariate kernel density estimation	55
Multivariate linear regression	56
Multivariate normal random values simulation on the simplex	57
Multivariate or univariate regression with compositional data in the covariates side using the alpha-transformation	58
Multivariate regression with compositional data	60
Multivariate skew normal random values simulation on the simplex	61
Multivariate t random values simulation on the simplex	62
Non linear least squares regression for compositional data	64
Principal component generalised linear models	65
Principal components regression	66
Regression with compositional data using the alpha-transformation	67
Regularised discriminant analysis for compositional data using the alpha-transformation	69
Regularised discriminant analysis for Euclidean data	70
Ridge regression	72

Ridge regression plot	73
Ridge regression with compositional data in the covariates side using the alpha-transformation	74
Ridge regression with the alpha-transformation plot	75
Simulation of compositional data from Gaussian mixture models	77
Simulation of compositional data from the folded model	78
Spatial median regression	79
Spatial sign covariance matrix	80
Ternary diagram	81
The additive log-ratio transformation and its inverse	83
The alpha-distance	84
The alpha-transformation	85
The Frechet mean for compositional data	87
The Helmert sub-matrix	88
The k-NN algorithm for compositional data	89
Total variability	91
Tuning of the bandwidth h of the kernel using the maximum likelihood cross validation .	92
Tuning of the k-NN algorithm for compositional data	93
Tuning of the principal components regression	95
Tuning the number of PCs in the PCR with compositional data using the alpha-transformation	96
Tuning the parameters of the regularised discriminant analysis	98
Tuning the principal components with GLMs	100
Tuning the value of alpha in the alpha-regression	101
Zero adjusted Dirichlet regression	103

Index	105
--------------	------------

Compositional-package *Compositional Data Analysis*

Description

A collection of functions for compositional data analysis.

Details

Package: Compositional
 Type: Package
 Version: 3.1
 Date: 2018-06-02
 License: GPL-2

Maintainers

Michail Tsagris <mtsagris@yahoo.gr>

Note

Acknowledgments:

Michail Tsagris would like to express his acknowledgments to Professor Andy Wood and Dr 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 mkde function. Van Thang Hoang from Hasselt university spotted a bug in the js.compreg function and is greatly acknowledged for that. Claudia Wehrhahn Cortes spotted a bug in the diri.reg function and she is greatly acknowledged for that.

Author(s)

Michail Tsagris <mtsagris@yahoo.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

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

Contour plot of a Dirichlet distribution in S^2

Contour plot of a Dirichlet distribution in S^2

Description

Contour plot of a Dirichlet distribution in S^2 .

Usage

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

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.

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

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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

[norm.contour](#), [bivt.contour](#), [comp.kerncontour](#), [mixnorm.contour](#)

Examples

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

Contour plot of a Gaussian mixture model in S^2

Contour plot of a Gaussian mixture model in S^2

Description

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

Usage

```
mixnorm.contour(x, mod)
```

Arguments

x	A matrix with the compositional data.
mod	An object containing the output of a mix.compnorm model.

Details

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

Value

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

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.gr> and Giorgos Athineou <gioathineou@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")
mixnorm.contour(x, mod)
```

Contour plot of the kernel density estimate in S^2

Contour plot of the kernel density estimate in S^2

Description

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

Usage

```
comp.kerncontour(x, type = "alr", n = 100)
```

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.

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

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.gr> and Giorgos Athineou <gioathineou@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](#), [mixnorm.contour](#), [bivt.contour](#), [norm.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^2

Description

Contour plot of the normal distribution in S^2 .

Usage

```
norm.contour(x, type = "alr", n = 100, appear = TRUE)
```

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)?

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

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

References

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

See Also

[diri.contour](#), [mixnorm.contour](#), [bivt.contour](#), [skewnorm.contour](#)

Examples

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

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

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

Description

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

Usage

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

Arguments

<code>x</code>	A matrix with the compositional data. It has to be a 3 column matrix.
<code>type</code>	This is either "alr" or "ilr", corresponding to the additive and the isometric log-ratio transformation respectively.
<code>n</code>	The number of grid points to consider over which the density is calculated.
<code>appear</code>	Should the available data appear on the ternary plot (TRUE) or not (FALSE)?

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

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.gr> and Giorgos Athineou <gioathineou@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](#), [mixnorm.contour](#), [bivt.contour](#), [norm.contour](#)

Examples

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

Contour plot of the t distribution in S^2

Contour plot of the t distribution in S^2

Description

Contour plot of the t distribution in S^2 .

Usage

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

Arguments

<code>x</code>	A matrix with compositional data. It has to be a 3 column matrix.
<code>type</code>	This is either "alr" or "ilr", corresponding to the additive and the isometric log-ratio transformation respectively.
<code>n</code>	The number of grid points to consider over which the density is calculated.
<code>appear</code>	Should the available data appear on the ternary plot (TRUE) or not (FALSE)?

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

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

References

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

See Also

[diri.contour](#), [mixnorm.contour](#), [norm.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 the regularised discriminant analysis with compositional data using the alpha-transformation
Cross validation for the regularised discriminant analysis with compositional data using the α -transformation

Description

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

Usage

```
alfarda.tune(x, ina, a = seq(-1, 1, by = 0.1), M = 10,  
gam = seq(0, 1, by = 0.1), del = seq(0, 1, by = 0.1),  
ncores = 1, mat = NULL)
```

Arguments

x	A matrix with the available compositional data. Zeros are allowed.
ina	A group indicator variable for the available 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.
M	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 the process.
mat	You can specify your own folds by giving a mat, where each column is a fold. Each column contains indices of the observations. You can also leave it NULL and it will create folds.

Details

A k-fold cross validation is performed.

Value

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.

Author(s)

Michail Tsagris

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

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.

See Also

[alfa.rda](#), [rda.tune](#)

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), M = 10,
gam = seq(0.1, 0.3, by = 0.1), del = seq(0.1, 0.3, by = 0.1),
ncores = 1, mat = NULL)
```

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, M = 10, lambda = seq(0, 2, by = 0.1), mat = NULL,
ncores = 1, 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.
M	The number of folds. Set to 10 by default.
lambda	A vector with the a grid of values of λ to be used.
mat	You can specify your own folds by giving a mat, where each column is a fold. Each column contains indices of the observations. 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.
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@yahoo.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](#)

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

Examples

```
y <- as.vector(iris[, 1])
x <- as.matrix(iris[, 2:4])
ridge.tune( y, x, M = 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, M = 10, a = seq(-1, 1, by = 0.1),
lambda = seq(0, 2, by = 0.1), mat = NULL, ncores = 1,
graph = TRUE, col.nu = 15)
```

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.
M	The number of folds. Set to 10 by default.
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.
mat	You can specify your own folds by giving a mat, where each column is a fold. Each column contains indices of the observations. 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 the 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.

Details

A k-fold cross validation is performed.

Value

If graph is TRUE a field 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@yahoo.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. <http://arxiv.org/pdf/1106.1451.pdf>

See Also

[alfa.ridge](#), [ridge.tune](#)

Examples

```
library(MASS)
y <- as.vector(fg1[, 1])
x <- as.matrix(fg1[, 2:9])
x <- x / rowSums(x)
alfaridge.tune( y, x, M = 10, a = seq(0.1, 1, by = 0.1),
lambda = seq(0, 1, by = 0.1) )
```

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@yahoo.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

[diri.nr](#), [diri.est](#), [diri.contour](#), [rdiri](#)

Examples

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

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@yahoo.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](#)

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 = TRUE, xnew = NULL)
```

```
diri.reg2(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 $\phi = e^{x*b}$.

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.
hipar	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 if it is 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.

<code>sigma</code>	The covariance matrix of the regression parameters (for the mean vector and the <code>phi</code> parameter) in the function "diri.reg2".
<code>lev</code>	The leverage values.
<code>est</code>	For the "diri.reg" this contains the fitted or the predicted values (if <code>xnew</code> is not NULL). For the "diri.reg2" if <code>xnew</code> is NULL, this is also NULL.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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, Ralitz, Robert Rosenheck, and Daniel Zelterman (2008). Dirichlet component regression and its applications to psychiatric data. *Computational statistics & data analysis* 52(12): 5344-5355.

See Also

[js.compreg](#), [kl.compreg](#), [ols.compreg](#), [comp.reg](#), [alfa.reg](#)

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)
```

Divergence based regression for compositional data

Divergence based regression for compositional data

Description

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

Usage

```
kl.compreg(y, x, B = 1, ncores = 1, xnew = NULL, tol = 1e-07, maxiters = 50)
js.compreg(y, x, 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 continuous or categorical or both.
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. This involves numerical optimization. There is no actual log-likelihood. As for the `js.compreg` another divergence is used.

Value

A list including:

runtime	The time required by the regression.
iters	The number of iterations required by the Newton-Raphson in the <code>kl.compreg</code> function.
loglik	The log-likelihood. This is actually a quasi multinomial regression. This is basically 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 <code>xnew</code> if <code>xnew</code> is not NULL.

Author(s)

Michail Tsagris

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

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)
```

Empirical likelihood for a one sample mean vector hypothesis testing

Empirical likelihood for a one sample mean vector hypothesis testing

Description

Empirical likelihood for a one sample mean vector hypothesis testing.

Usage

```
el.test1(x, mu, R = 1, ncores = 1, graph = FALSE)
```

Arguments

x	A matrix containing Euclidean data.
mu	The hypothesized mean vector.
R	If R is 1 no bootstrap calibration is performed and the classical p-value via the χ^2 distribution is returned. If R is greater than 1, the bootstrap p-value is returned.
ncores	The number of cores to use, set to 1 by default.
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

Multivariate hypothesis test for a one sample mean vector. This is a non parametric test and it works for univariate and multivariate data.

Value

A list with the outcome of the function `el.test` which includes the -2 log-likelihood ratio, the observed P-value by chi-square approximation, the final value of Lagrange multiplier λ , the gradient at the maximum, the Hessian matrix, the weights on the observations (probabilities multiplied by the sample size) and the number of iteration performed. In addition the runtime of the procedure is reported. In the case of bootstrap, the bootstrap p-value is also returned.

Author(s)

Michail Tsagris

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

References

Owen, A. (1990). Empirical likelihood ratio confidence regions. *Annals of Statistics*, 18, 90-120.

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

See Also

[eel.test1](#), [hotel1T2](#), [james](#), [hotel2T2](#), [maov](#), [el.test2](#), [comp.test](#)

Examples

```
x <- Rfast::rmvnorm(100, numeric(10), diag( rexp(10, 0.5) ) )
el.test1(x, mu = numeric(10) )
eel.test1(x, mu = numeric(10) )
```

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

Empirical likelihood is a non parametric hypothesis testing procedure for one sample. The generalization to two (or more samples) is via searching for the mean vector that minimizes the sum of the two test statistics.

Value

A list including:

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

Author(s)

Michail Tsagris

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

References

- G.S. James (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
- 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.

See Also

[eel.test2](#), [maovjames](#), [maov](#), [hotel2T2](#), [james](#), [comp.test](#)

Examples

```
e1.test2( y1 = as.matrix(iris[1:25, 1:4]), y2 = as.matrix(iris[26:50, 1:4]), R = 0 )
e1.test2( y1 = as.matrix(iris[1:25, 1:4]), y2 = as.matrix(iris[26:50, 1:4]), R = 1 )
e1.test2( y1 = as.matrix(iris[1:25, 1:4]), y2 = as.matrix(iris[26:50, 1:4]), R = 2 )
```

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-09)
```

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@yahoo.gr> and Giorgos Athineou <gioathineou@gmail.com>

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. Karkkainen 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

[spat.med](#), [sscov](#), [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 value of α 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 values are allowed.

Details

This is a function for choosing or estimating the value of α in the folded model (Tsagris and Stewart, 2018).

Value

A list including:

runtime	The runtime of the algorithm.
best	The estimated optimal α of the folded model.
loglik	The maximised 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@yahoo.gr>

References

Tsagris Michail and Stewart Connie, (2018). A folded model for compositional data analysis. <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. <http://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 asymptotic 95% confidence interval computed from the log-likelihood ratio test.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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.

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 for a one sample mean vector hypothesis testing
Exponential empirical likelihood for a one sample mean vector hypothesis testing

Description

Exponential empirical likelihood for a one sample mean vector hypothesis testing.

Usage

```
eel.test1(x, mu, tol = 1e-06, R = 1)
```

Arguments

x	A matrix containing Euclidean data.
mu	The hypothesized mean vector.
tol	The tolerance value used to stop the Newton-Raphson algorithm.
R	The number of bootstrap samples used to calculate the p-value. If R = 1 (default value), no bootstrap calibration is performed

Details

Multivariate hypothesis test for a one sample mean vector. This is a non parametric test and it works for univariate and multivariate data.

Value

A list including:

p	The estimated probabilities.
lambda	The value of the Lagrangian parameter λ .
iter	The number of iterations required by the newton-Raphson algorithm.
info	The value of the log-likelihood ratio test statistic along with its corresponding p-value.
runtime	The runtime of the process.

Author(s)

Michail Tsagris

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

References

- Jing Bing-Yi and Andrew TA Wood (1996). Exponential empirical likelihood is not Bartlett correctable. *Annals of Statistics* 24(1): 365-369.
- Owen A. B. (2001). *Empirical likelihood*. Chapman and Hall/CRC Press.

See Also

[el.test1](#), [hotel1T2](#), [james](#), [hotel2T2](#), [maov](#), [el.test2](#), [comp.test](#)

Examples

```
x <- Rfast::rmvnorm(100, numeric(10), diag( rexp(10, 0.5) ) )
eel.test1(x, numeric(10) )
el.test1(x, numeric(10) )
```

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 is a non parametric hypothesis testing procedure for one sample. The generalization to two (or more samples) is via searching for the mean vector that minimises the sum of the two test statistics.

Value

A list including:

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

Author(s)

Michail Tsagris

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

References

- Jing Bing-Yi and Andrew TA Wood (1996). Exponential empirical likelihood is not Bartlett correctable. *Annals of Statistics* 24(1): 365-369.
- G.S. James (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
- 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.
- 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.
- 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](#), [maov](#), [hotel2T2](#), [james](#), [comp.test](#)

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 values 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 the 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-likelihood, 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@yahoo.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. <http://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)
```

Fitting a Dirichlet distribution

Fitting a Dirichlet distribution

Description

Estimation of the parameters of a fitted Dirichlet distribution.

Usage

```
diri.est(x, type = "mle")
```

Arguments

<code>x</code>	A matrix containing compositional data.
<code>type</code>	If you want to estimate the parameters use <code>type="mle"</code> . If you want to estimate the mean vector along with the precision parameter, the second parametrisation of the Dirichlet, use <code>type="prec"</code> .

Details

Maximum likelihood estimation of the parameters of a Dirichlet distribution is performed.

Value

A list including:

<code>loglik</code>	The value of the log-likelihood.
<code>param</code>	The estimated parameters.
<code>phi</code>	The estimated precision parameter, if <code>type = "prec"</code> .
<code>a</code>	The estimated mean vector, if <code>type = "prec"</code> .
<code>runtime</code>	The run time of the maximisation procedure.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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](#)

Examples

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

Fitting a Dirichlet distribution via Newton-Rapshon

Fitting a Dirichlet distribution via Newton-Rapshon

Description

Fitting a 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-Rapshon 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.
runtime	The run time of the procedure.

Author(s)

Michail Tsagris

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

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](#)

Examples

```
x <- rdiri( 100, c(5, 7, 5, 8, 10, 6, 4) )
diri.nr(x)
diri.nr(x, type = 2)
diri.est(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")
```

Arguments

- | | |
|-------|--|
| x | A matrix with the compositional data. |
| g | How many clusters to create. |
| model | The type of model to be used. <ol style="list-style-type: none"> 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 Either the additive ("alr") or the isometric ("ilr") log-ratio transformation is to be used..

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@yahoo.gr> and Giorgos Athineou <gioathineou@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

[bic.mixcompnorm](#), [rmixcomp](#), [mixnorm.contour](#)

Examples

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

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, 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 <code>klcompreg.boot</code> the first column is removed.
x	The predictor variable(s), they can be either continuous or categorical or both. In the <code>klcompreg.boot</code> this is the design matrix, with the ones in the first column.
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:

runtime	The time required by the regression.
be	The beta coefficients.
seb	The standard error of the beta coefficients, if bootstrap is chosen, i.e. if $B > 1$.
est	The fitted or the predicted values (if <code>xnew</code> is not NULL).

For the `klcompreg.boot` a matrix with the estimated coefficients.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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 1 sample t-test for Euclidean data
Hotelling's multivariate version of the 1 sample t-test for Euclidean data

Description

Hotelling's test for testing one Euclidean population mean vector.

Usage

```
hotel1T2(x, M, a = 0.05, R = 999, graph = FALSE)
```

Arguments

x	A matrix containing Euclidean data.
a	The significance level, set to 0.05 by default.
M	The hypothesized mean vector.
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

Multivariate hypothesis test for a one sample mean vector. This is the multivariate analogue of the one sample t-test. The p-value can be calculated either asymptotically or via bootstrap.

Value

A list including:

<code>m</code>	The sample mean vector.
<code>info</code>	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.
<code>pvalue</code>	The bootstrap p-value is bootstrap is employed.
<code>runtime</code>	The runtime of the bootstrap calibration.

Author(s)

Michail Tsagris

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

References

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

See Also

[eel.test1](#), [el.test1](#), [james](#), [hotel2T2](#), [maov](#), [el.test2](#), [comp.test](#)

Examples

```
x <- Rfast::rmvnorm(100, numeric(10), diag( rexp(10,0.5) ) )
hotel1T2(x, numeric(10), R = 1)
hotel1T2(x, numeric(10), R = 999, graph = TRUE)
```

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

Multivariate analysis of variance assuming equality of the covariance matrices. The p-value can be calculated either asymptotically or via bootstrap.

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@yahoo.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

Everitt B. (2005). *An R and S-Plus Companion to Multivariate Analysis* p. 139-140. Springer.

See Also

[james](#), [maov](#), [el.test2](#), [comp.test](#)

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 )
james( as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]) )
james( 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@yahoo.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 Hypotheses 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

[maovjames](#), [maov](#), [hotel2T2](#), [el.test2](#)

Examples

```
ina <- rep(1:2, each = 50)
comp.test( as.matrix(iris[1:100, 1:4]), ina, test = "james", R = 0 )
comp.test( as.matrix(iris[1:100, 1:4]), ina, test = "hotel", R = 0 )
comp.test( as.matrix(iris[1:100, 1:4]), ina, test = "el", R = 0 )
comp.test( as.matrix(iris[1:100, 1:4]), ina, test = "eel", R = 0 )
```

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@yahoo.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

Tsagris M.T., Preston S. and Wood A.T.A. (2016). Improved classification for compositional data using the α -transformation. *Journal of Classification* (to appear). <http://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. <http://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

Multivariate analysis of variance without assuming equality of the covariance matrices. The p-value can be calculated either asymptotically or via bootstrap. The James test (1954) or a modification proposed by Krishnamoorthy and Yanping (2006) is implemented. The James test uses a corrected chi-square distribution, whereas the modified version uses an F distribution.

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@yahoo.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

G.S. James (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

Krishnamoorthy K. and Yanping Xia. On Selecting Tests for Equality of Two Normal Mean Vectors (2006). *Multivariate Behavioral Research* 41(4): 533-548

See Also

[hotel2T2](#), [maovjames](#), [el](#), [comp.test](#)

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]) )
hotel2T2( as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]) )
```

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@yahoo.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

[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")
```

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@yahoo.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 Dirichlet parameters are equal.

Value

A list including:

<code>est.par</code>	The estimated parameters under the alternative hypothesis.
<code>one.par</code>	The value of the estimated parameter under the null hypothesis.
<code>res</code>	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@yahoo.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

[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)
```

Mixture model selection via BIC

Mixture model selection via BIC

Description

Mixture model selection via BIC.

Usage

```
bic.mixcompnorm(x, A, type = "alr")
```

Arguments

x	A matrix with compositional data.
A	The maximum number of components, clusters, to be considered.
type	The type of transformation to be used, either additive log-ratio ("alr") or the isometric log-ratio ("ilr").

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.

Author(s)

Michail Tsagris

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

References

Ryan P. Browne, Aisha ElSherbiny and Paul D. McNicholas (2015). mixture: Mixture Models for Clustering and Classification. R package version 1.4.

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](#), [mixnorm.contour](#), [rmixcomp](#)

Examples

```
library(MASS)
x <- as.matrix( iris[, 1:4] )
x <- x/ rowSums(x)
bic.mixcompnorm(x, 6, type = "alr")
bic.mixcompnorm(x, 6, type = "ilr")
```

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

<code>y</code>	A matrix with continuous data.
<code>plot</code>	If <code>plot</code> is TRUE the value of the maximum log-likelihood as a function of the degrees 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:

<code>center</code>	The location estimate.
<code>scatter</code>	The scatter matrix estimate.
<code>df</code>	The estimated degrees of freedom.
<code>loglik</code>	The log-likelihood value.
<code>mesos</code>	The classical mean vector.
<code>covariance</code>	The classical covariance matrix.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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](#), [rmvt](#)

Examples

```
x <- as.matrix(iris[, 1:4])
multivt(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 values are allowed.
a	A value of α .

Details

This is a function for choosing or estimating the value of α in the folded model (Tsagris and Stewart, 2018). 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 maximized 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.

If "a.mle" is called, the log-likelihood is returned only.

Author(s)

Michail Tsagris

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

References

Tsagris Michail and Stewart Connie, (2018). A folded model for compositional data analysis. <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. <http://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])
```

Multivariate analysis of variance

Multivariate analysis of variance

Description

Multivariate analysis of variance with 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-square approximation has been used.
result	The test statistic and the p-value.

Author(s)

Michail Tsagris

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

References

Johnson and Wichern (2007, 6th Edition). Applied Multivariate Statistical Analysis p. 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](#), [comp.test](#)

Examples

```
maov( as.matrix(iris[,1:4]), iris[,5] )
maovjames( as.matrix(iris[,1:4]), iris[,5] )
```

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

<code>x</code>	A matrix containing Euclidean data.
<code>ina</code>	A numerical or factor variable indicating the groups of the data.
<code>a</code>	The significance level, set to 0.005 by default.

Details

Multivariate analysis of variance without assuming equality of the covariance matrices.

Value

A vector with the next 4 elements:

<code>test</code>	The test statistic.
<code>correction</code>	The value of the correction factor.
<code>corr.critical</code>	The corrected critical value of the chi-square distribution.
<code>p-value</code>	The p-value of the corrected test statistic.

Author(s)

Michail Tsagris

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

References

G.S.James (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

[maov](#), [hotel2T2](#), [james](#), [comp.test](#)

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, thumb = "silverman")
```

Arguments

x	A matrix with Euclidean (continuous) data.
h	The bandwidth 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.
thumb	Do you want to use a rule of thumb for the bandwidth parameter? If no, leave it "none", or else 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 necessarily given) bandwidth value. It is used a wrapper for the function [comp.kerncontour](#).

Value

A vector with the density estimates calculated for every vector.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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 linear regression
Multivariate linear regression

Description

Multivariate linear regression.

Usage

```
multivreg(y, x, plot = TRUE, xnew = NULL)
```

Arguments

y	A matrix with the Euclidean (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 <code>lm</code> , 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@yahoo.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@yahoo.gr> and Giorgos Athineou <gioathineou@gmail.com>

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-
*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	A number at least equal to 1. How many principal components to use.
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 calculated 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 including:

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, otherwise it is NULL.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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. <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

[pcr](#), [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 compositional 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 spatial median regression, which is also robust.
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, put it here. Otherwise 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 NULL.

Author(s)

Michail Tsagris

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

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](#), [rmvt](#)

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

<code>n</code>	The sample size, a numerical value.
<code>m</code>	The mean vector in R^d .
<code>s</code>	The covariance matrix in R^d .
<code>dof</code>	The degrees of freedom.
<code>type</code>	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@yahoo.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

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

See Also

[comp.den](#), [rdiri](#), [rcompnorm](#), [rmvt](#)

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)
```

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, 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 have to be continuous.
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.
seb	The standard error of the beta coefficients, if bootstrap is chosen, i.e. if B > 1.
est	The fitted of xnew if xnew is not NULL.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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.

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)
```

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, either 0 and 1 (binary) or discrete (count) data.
x	A matrix with the predictor variable(s), they have to be continuous.
k	A number at least equal to 1. How many principal components to use.
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.

Value

A list including:

model	The summary of the logistic or Poisson regression model.
per	The percentage of variance of the predictor variables retained by the k principal components.
est	The fitted or the predicted values (if xnew is not NULL).

Author(s)

Michail Tsagris

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

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](#), [alfa.pcr](#), [alfapcr.tune](#), [glm](#)

Examples

```
library(MASS)
x <- as.matrix(iris[, 1:4])
y<- rbinom(150, 1, 0.6)
mod<- glm.pcr(y, x, k = 1)
```

Principal components regression

Principal components regression

Description

Principal components regression.

Usage

```
pcr(y, x, k = 1, xnew = NULL)
```

Arguments

y	A real values vector.
x	A matrix with the predictor variable(s), they have to be continuous.
k	The number of principal components to use.
xnew	If you have new data use it, otherwise leave it NULL.

Details

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:

be	The beta coefficients.
per	The percentage of variance of the cross product of the independent variables explained by the k components.
est	The fitted or the predicted values (if xnew is not NULL).

Author(s)

Michail Tsagris

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

References

Jolliffe I.T. (2002). Principal Component Analysis.

See Also

[pcr.tune](#), [alfa.pcr](#), [glm.pcr](#)

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
y <- as.vector(fgl[, 1])
mod1 <- pcr(y, x, 1)
mod2 <- pcr(y, x, 2)
```

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, xnew = NULL, yb = NULL, seb = FALSE)
```

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 multivariate regression.
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. Otherwise 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.
seb	Do you want the standard error of the coefficients to be returned? In the alfareg.tune function this extra computation that is avoided can save some time.

Details

The α -transformation is applied to the compositional data first and then multivariate regression is applied. This involves numerical optimisation.

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 for xnew if xnew is not NULL.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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. <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>
- 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 discriminant analysis for compositional data using the alpha-transformation
Regularised discriminant analysis for compositional data using the α -transformation

Description

Regularised discriminant analysis for compositional data using the α -transformation.

Usage

```
alfa.rda(xnew, x, ina, a, gam = 1, del = 0)
```

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 .
x	A matrix with the available compositional data. Zeros are allowed, but you must be careful to choose strictly positive values 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

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 $1 - \text{gam}$ is the weight of the spherical covariance matrix, $S_a = \text{gam} * S_p + (1 - \text{gam}) * s_p$. Then it is a compromise between LDA and QDA. del is the weight of S_a and $1 - \text{del}$ the weight of each group covariance group. This function is a wrapper for [alfa.rda](#).

Value

A list including:

prob	The estimated probabilities of the new data of belonging to each group.
scores	The estimated scores 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@yahoo.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. <http://arxiv.org/pdf/1106.1451.pdf>

See Also

[rda](#), [alfa](#)

Examples

```
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
ina <- iris[, 5]
mod <- alfa.rda(x, x, ina, 0)
table(ina, mod$est)
mod2 <- alfa.rda(x, x, ina, 0.5)
table(ina, mod2$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 available 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, $S_a = \text{gam} * S_p + (1-\text{gam}) * s_p$. Then it is a compromise between LDA and QDA. `del` is the weight of S_a and `1-del` the weight of each group covariance group. This function is a wrapper for [alfa.rda](#).

Value

A list including:

prob	The estimated probabilities of the new data of belonging to each group.
scores	The estimated scores 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@yahoo.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 TA Wood (2016). Improved classification for compositional data using the α -transformation. Journal of classification, 33(2): 243-261. <http://arxiv.org/pdf/1106.1451.pdf>

See Also

[rda.tune](#), [alfa](#)

Examples

```
x <- as.matrix(iris[, 1:4])
ina <- iris[, 5]
mod <- rda(x, x, ina)
table(ina, mod$est)
```

Ridge regression	<i>Ridge regression</i>
------------------	-------------------------

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 bootstrap 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 error of the coefficients. 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@yahoo.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@yahoo.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 carefull to choose strictly positive vvalues 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@yahoo.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. <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

[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@yahoo.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. <http://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) )
```

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@yahoo.gr> and Giorgos Athineou <gioathineou@gmail.com>

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 the folded model

Simulation of compositional data from the folded model

Description

Simulation of compositional data from the folded model.

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@yahoo.gr>

References

Tsagris Michail and Stewart Connie, (2018). A folded model for compositional data analysis. <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

<code>y</code>	A matrix with the compositional data. Zero values are not allowed.
<code>x</code>	The predictor variable(s), they have to be continuous.
<code>xnew</code>	If you have new data use it, otherwise leave it NULL.
<code>tol</code>	The threshold upon which to stop the iterations of the Newton-Rapshon algorithm.
<code>ses</code>	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 <code>y</code> and 30 for <code>x</code> , when <code>ses = FALSE</code> the algorithm can take 0.20 seconds, but when <code>ses = TRUE</code> 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:

<code>iter</code>	The number of iterations that were required.
<code>runtime</code>	The time required by the regression.
<code>be</code>	The beta coefficients.
<code>seb</code>	The standard error of the beta coefficients is returned if <code>ses=TRUE</code> and <code>NULL</code> otherwise.
<code>est</code>	The fitted of <code>xnew</code> if <code>xnew</code> is not <code>NULL</code> .

Author(s)

Michail Tsagris

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

References

Biman Chakraborty (2003) On multivariate quantile regression. Journal of Statistical Planning and Inference 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)
```

Spatial sign covariance matrix

Spatial sign covariance matrix

Description

Spatial sign covariance matrix.

Usage

```
sscov(x, me = NULL, tol = 1e-09)
```

Arguments

<code>x</code>	A matrix with continuous data.
<code>me</code>	If you have already computed the spatial median plug it in here.
<code>tol</code>	A tolerance level to terminate the process of finding the spatial median. This is set to 1e-09 by default.

Details

The spatial median is at first computed (if not supplied) and then the covariance matrix. This is used in the function [comp.den](#).

Value

The spatial sign covariance matrix.

Author(s)

Michail Tsagris

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

References

Durre A, Vogel D. and D.E. Tyler D.E.(2014). The spatial sign covariance matrix with unknown location. *Journal of Multivariate Analysis*, 130: 107-117. <http://arxiv.org/pdf/1307.5706v2.pdf>

See Also

[spat.med](#), [comp.den](#)

Examples

```
library(MASS)
sscov( as.matrix(iris[, 1:4]) )
```

Ternary diagram

Ternary diagram

Description

Ternary diagram.

Usage

```
ternary(x, means = TRUE, pca = FALSE)
```

Arguments

x	A matrix with the compositional data.
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.

Details

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. There are two ways to create a ternary graph. The one I used here, where each edge is equal to 1 and the one Aitchison (1986) uses. For every given point, the sum of the distances from the edges is equal to 1. 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 matrix with the closed geometric and the simple arithmetic mean vector.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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

[comp.den](#), [alfa](#), [diri.contour](#), [comp.kerncontour](#)

Examples

```
library(MASS)
x <- as.matrix(fg1[, 2:4])
ternary(x, means = FALSE)
x <- as.matrix(iris[, 1:3])
ternary(x, pca = TRUE)
```

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.

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@yahoo.gr>

References

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

See Also

[alfa](#), [link{alfainv,}](#) [alfa.profile](#), [alfa.tune](#)

Examples

```
library(MASS)
x <- as.matrix(fg1[, 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

A matrix including: The pairwise distances of all observations.

Author(s)

Michail Tsagris

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

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. <http://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*. <http://arxiv.org/pdf/1106.1451.pdf>

See Also

[alfa](#), [alfainv](#), [alfa.reg](#)

Examples

```
library(MASS)
x <- as.matrix(fg1[1:20, 2:9])
x <- x / rowSums(x)
alfadist(x, 0.1)
alfadist(x, 1)
```

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 singular 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 returns 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@yahoo.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

Tsagris Michail and Stewart Connie, (2018). A folded model for compositional data analysis. <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. <http://arxiv.org/pdf/1106.1451.pdf>

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

See Also

[alfainv](#), [alfa.profile](#), [alfa.tune](#) [a.est](#), [alpha.mle](#)

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 Frechet mean for compositional data
The Frechet mean for compositional data

Description

Mean vector 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.

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

A vector with the Frechet mean for the given value of α .

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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.

See Also

[alfa](#), [alfainv](#), [profile](#)

Examples

```
library(MASS)
x <- as.matrix(fg1[, 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 greater 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@yahoo.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. <http://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-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, type = "S", apostasi = "ESOV", mesos = TRUE)
```

```
alfa.knn(xnew, x, ina, a = 1, k = 5, type = "S", mesos = TRUE, apostasi = "euclidean")
```

Arguments

<code>xnew</code>	A matrix with the new compositional data whose group is to be predicted. Zeros are allowed, but you must be carefull to choose strictly positive values of α or not to set <code>apostasi="Ait"</code> .
<code>x</code>	A matrix with the available compositional data. Zeros are allowed, but you must be carefull to choose strictly positive vvalues of α or not to set <code>apostasi="Ait"</code> .
<code>ina</code>	A group indicator variable for the available data.
<code>a</code>	The value of α . As zero values in the compositional data are allowed, you must be careful to choose strictly positive vvalues of α . You have the option to put <code>a=NULL</code> . In this case, the <code>xnew</code> and <code>x</code> are assumed to be the already α -transformed data.
<code>k</code>	The number of nearest neighbours to consider.
<code>type</code>	This can be either "S" for the standard k-NN or "NS" for the non standard (see details).
<code>apostasi</code>	The type of distance to use. For the <code>comp.knn</code> this can be one of the following: "ESOV", "taxicab", "Ait", "Hellinger", "angular" or "CS". See the references for them. For the <code>alfa.knn</code> this can be either "euclidean" or "manhattan".
<code>mesos</code>	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).

Details

The k-NN algorithm is applied for the compositional data. There are many metrics and possibilities to choose from. The standard algorithm finds the k nearest observations to a new observation and allocates it to the class which appears most times in the neighbours. The non standard algorithm is slower but perhaps more accurate. For every group it finds the k nearest neighbours to the new observation. 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@yahoo.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 (2016). An approach to measure distance between compositional diet estimates containing essential zeros. *Journal of Applied Statistics* 44.7 (2017): 1137-1152.

See Also

[compknn.tune](#), [rda](#), [alfa](#)

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)
```

Total variability	<i>Total variability</i>
-------------------	--------------------------

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@yahoo.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 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

<code>x</code>	A matrix with Euclidean (continuous) data.
<code>low</code>	The minimum value to search for the optimal bandwidth value.
<code>up</code>	The maximum value to search for the optimal bandwidth value.
<code>s</code>	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:

<code>hopt</code>	The optimal bandwidth value.
<code>maximum</code>	The value of the pseudo-log-likelihood at that given bandwidth value.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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 k-NN algorithm for compositional data

Tuning of the he 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 M-fold cross-validation.

Usage

```
compknn.tune(x, ina, M = 10, A = 5, type= "S", mesos = TRUE,
a = seq(-1, 1, by = 0.1), apostasi = "ESOV", mat = NULL, graph = FALSE)
```

```
alfaknn.tune(x, ina, M = 10, A = 5, type = "S", mesos = TRUE,
a = seq(-1, 1, by = 0.1), apostasi = "euclidean", mat = NULL, graph = FALSE)
```

Arguments

<code>x</code>	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 <code>apostasi= "Ait"</code> .
<code>ina</code>	A group indicator variable for the available data.
<code>M</code>	The number of folds to be used. This is taken into consideration only if the matrix <code>"mat"</code> is not supplied.
<code>A</code>	The maximum number of nearest neighbours to consider. Note that the 1 nearest neighbour is not used.
<code>type</code>	This can be either <code>"S"</code> for the standard k-NN or <code>"NS"</code> for the non standard (see details).
<code>mesos</code>	This is used in the non standard algorithm. If <code>TRUE</code> , the arithmetic mean of the distances is calculated, otherwise the harmonic mean is used (see details).
<code>a</code>	A grid of values of α to be used only if the distance chosen allows for it.
<code>apostasi</code>	The type of distance to use. For the <code>comp.knn</code> this can be one of the following: <code>"ESOV"</code> , <code>"taxicab"</code> , <code>"Ait"</code> , <code>"Hellinger"</code> , <code>"angular"</code> or <code>"CS"</code> . See the references for them. For the <code>alfa.knn</code> this can be either <code>"euclidean"</code> or <code>"manhattan"</code> .
<code>mat</code>	You can specify your own folds by giving a <code>mat</code> , where each column is a fold. Each column contains indices of the observations. You can also leave it <code>NULL</code> and it will create folds.
<code>graph</code>	If set to <code>TRUE</code> 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 standard algorithm finds the k nearest observations to a new observation and allocates it to the class which appears most times in the neighbours. The non standard algorithm is slower but perhaps more accurate. For every group it finds the k nearest neighbours to the new observation. 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 list including:

<code>ela</code>	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).
<code>performance</code>	The estimated rate of correct classification.
<code>best_a</code>	The best value of α . This is returned for "ESOV" and "taxicab" only.
<code>best_k</code>	The best number of nearest neighbours.
<code>runtime</code>	The run time of the cross-validation procedure.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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 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. <http://arxiv.org/pdf/1506.04976v2.pdf>
- Connie Stewart (2016). An approach to measure distance between compositional diet estimates containing essential zeros. *Journal of Applied Statistics* 44.7 (2017): 1137-1152.

See Also

[comp.knn](#), [rda](#), [alfa](#)

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 principal components regression

Tuning of the principal components regression

Description

Tuning the number of principal components in the principal components regression.

Usage

```
pcr.tune(y, x, M = 10, maxk = 50, mat = NULL, ncores = 1, graph = TRUE)
```

Arguments

y	A real valued vector.
x	A matrix with the predictor variables, they have to be continuous.
M	The number of folds in the cross validation.
maxk	The maximum number of principal components to check.
mat	You can specify your own folds by giving a mat, where each column is a fold. Each column contains indices of the observations. 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.
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 regression. This is used by [alfapcr.tune](#).

Value

A list including: If graph is TRUE a plot of the performance versus the number of principal components will appear.

m _{sp}	A matrix with the mean squared error of prediction (MSPE) for every fold.
m _{spe}	A vector with the mean squared error of prediction (MSPE), each value corresponds to a number of principal components.
k	The number of principal components which minimizes the MSPE.
performance	The lowest value of the MSPE.
runtime	The time required by the cross-validation procedure.

Author(s)

Michail Tsagris

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

References

Jolliffe I.T. (2002). Principal Component Analysis.

See Also

[glm.pcr.tune](#), [glm.pcr](#), [alfa.pcr](#), [alfapcr.tune](#)

Examples

```
library(MASS)
x <- as.matrix(fgl[, 2:9])
y <- as.vector(fgl[, 1])
pcr.tune(y, x, M = 10, maxk = 50, mat = NULL, ncores = 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", M = 10, maxk = 50, a = seq(-1, 1, by = 0.1),
mat = NULL, ncores = 1, graph = TRUE, col.nu = 15)
```

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".
M	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 and the solution exists in a closed form, since it the classical multivariate regression. The estimated bias correction via the (Tibshirani and Tibshirani (2009) criterion is applied.
mat	You can specify your own folds by giving a mat, where each column is a fold. Each column contains indices of the observations. 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 the 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.

Details

The α -transformation is applied to the compositional data first and the function "pcr.tune" or "glm-pcr.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@yahoo.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. <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>
- Jolliffe I.T. (2002). *Principal Component Analysis*.

See Also

[alfa](#), [profile](#), [alfa.pcr](#), [pcr.tune](#), [glm.pcr.tune](#), [glm](#)

Examples

```
library(MASS)
y <- as.vector(fg1[, 1])
x <- as.matrix(fg1[, 2:9])
x <- x/ rowSums(x)
mod <- alfapcr.tune(y, x, M = 10, maxk = 50, a = seq(-1, 1, by = 0.1), mat = NULL, ncores = 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 Euclidean data.

Usage

```
rda.tune(x, ina, M = 10, gam = seq(0, 1, by = 0.1), del = seq(0, 1, by = 0.1),
ncores = 1, mat = NULL)
```

Arguments

<code>x</code>	A matrix with the data.
<code>ina</code>	A group indicator variable for the available data.
<code>M</code>	The number of folds in the cross validation.
<code>gam</code>	A grid of values for the γ parameter as defined in Tsagris et al. (2016).
<code>del</code>	A grid of values for the δ parameter as defined in Tsagris et al. (2016).
<code>ncores</code>	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 the process.
<code>mat</code>	You can specify your own folds by giving a <code>mat</code> , where each column is a fold. Each column contains indices of the observations. You can also leave it <code>NULL</code> and it will create folds.

Details

Cross validation is performed to select the optimal parameters for the regularised discriminant analysis and also estimate the rate of accuracy.

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, $S_a = \text{gam} * S_p + (1-\text{gam}) * s_p$. Then it is a compromise between LDA and QDA. `del` is the weight of S_a and $1-\text{del}$ the weight of each group covariance group. This function is a wrapper for [alfa.rda](#).

Value

A list including: If `graph` is TRUE a plot of the performance versus the number of principal components will appear.

<code>per</code>	An array with the estimate rate of correct classification for every fold. For each of the M matrices, the row values correspond to <code>gam</code> and the columns to the <code>del</code> parameter.
<code>percent</code>	A matrix with the mean estimated rates of correct classification. The row values correspond to <code>gam</code> and the columns to the <code>del</code> parameter.
<code>se</code>	A matrix with the standard error of the mean estimated rates of correct classification. The row values correspond to <code>gam</code> and the columns to the <code>del</code> parameter.
<code>result</code>	The estimated rate of correct classification along with the best <code>gam</code> and <code>del</code> parameters.
<code>runtime</code>	The time required by the cross-validation procedure.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.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 TA Wood (2016). Improved classification for compositional data using the α -transformation. Journal of classification, 33(2):243-261. <http://arxiv.org/pdf/1106.1451.pdf>

See Also

[rda](#), [alfa](#)

Examples

```
mod <- rda.tune(as.matrix(iris[, 1:4]), iris[, 5], M = 10, gam = seq(0, 1, by = 0.2),
del = seq(0, 1, by = 0.2), ncores = 1, mat = NULL)
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

```
glmpr.tune(y, x, M = 10, maxk = 10, mat = NULL, ncores = 1, graph = TRUE)
```

```
multinompr.tune(y, x, M = 10, maxk = 10, mat = NULL, ncores = 1, graph = TRUE)
```

Arguments

y	A real valued vector for the "glmpr.tune" with either two numbers, 0 and 1 for example, for the binomial regression or with positive discrete numbers for the poisson. For the "multinompr.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.
M	The number of folds in the cross validation.
maxk	The maximum number of principal components to check.
mat	You can specify your own folds by giving a mat, where each column is a fold. Each column contains indices of the observations. 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.
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, 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@yahoo.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(fg1[, 2:9])
y <- rpois(214, 10)
glmpr.tune(y, x, M=10, maxk = 20, mat = 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), K = 10,
mat = NULL, nc = 1, 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.
K	The number of folds to split the data.
mat	You can specify your own folds by giving a mat, where each column is a fold. Each column contains indices of the observations. 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.
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@yahoo.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

Tsagris Michail (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

[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 = c(0.2, 0.35, 0.05), K = 5, nc = 1)
```

Zero adjusted Dirichlet regression

Zero adjusted Dirichlet regression

Description

Zero adjusted Dirichlet regression.

Usage

```
zadr(y, x, xnew = NULL, tol = 1e-05)
```

```
mixreg(param, z)
```

Arguments

y	A matrix with the compositional data (dependent variable). The number of observations with at least one zero value should not 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 continuous or categorical or both.
xnew	If you have new data use it, otherwise leave it NULL.
tol	A tolerance level to terminate the maximisation process.
param	Some arguments passed on to the mixreg helper function.
z	Some arguments passed on to the mixreg helper function.

Details

A zero adjusted Dirichlet regression is being fitted. The likelihood consists 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.

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.
be	The beta coefficients.
seb	The standard error of the beta coefficients.
sigma	The covariance matrix of the regression parameters (for the mean vector and the phi parameter) in the function "diri.reg2".
est	The fitted or the predicted values (if xnew is not NULL).

Author(s)

Michail Tsagris

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

References

Tsagris M. and Stewart C. (2018). A Dirichlet Regression Model for Compositional Data with Zeros. Accepted at the Lobachevskii Journal of Mathematics.

Preprint available from <https://arxiv.org/pdf/1410.5011.pdf>

See Also

[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

- *Topic **-regression**
 - Tuning the value of alpha in the alpha-regression, [101](#)
- *Topic **-transformation**
 - Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation, [14](#)
 - Estimation of the value of alpha in the folded model, [25](#)
 - Estimation of the value of alpha via the profile log-likelihood, [26](#)
 - Fast estimation of the value of alpha, [31](#)
 - Inverse of the alpha-transformation, [42](#)
 - MLE of the folded model for a given value of alpha, [51](#)
 - Multivariate or univariate regression with compositional data in the covariates side using the alpha-transformation, [58](#)
 - Regression with compositional data using the alpha-transformation, [67](#)
 - Ridge regression with compositional data in the covariates side using the alpha-transformation, [74](#)
 - The alpha-distance, [84](#)
 - The alpha-transformation, [85](#)
 - The Frechet mean for compositional data, [87](#)
 - Tuning the number of PCs in the PCR with compositional data using the alpha-transformation, [96](#)
 - Tuning the value of alpha in the alpha-regression, [101](#)
- *Topic **Additive log-ratio-transformation**
 - The additive log-ratio transformation and its inverse, [83](#)
- *Topic **Bhattacharyya distance**
 - Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distributions, [45](#)
- *Topic **Compositional data**
 - Compositional-package, [3](#)
- *Topic **Contour plots**
 - Compositional-package, [3](#)
- *Topic **Dirichlet mean vector**
 - Log-likelihood ratio test for a Dirichlet mean vector, [46](#)
- *Topic **Dirichlet distribution**
 - Contour plot of a Dirichlet distribution in S^2 , [4](#)
 - Density values of a Dirichlet distribution, [15](#)
 - Dirichlet random values simulation, [17](#)
 - Fitting a Dirichlet distribution, [32](#)
 - Fitting a Dirichlet distribution via Newton-Rapshon, [33](#)
 - Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distributions, [45](#)
- *Topic **Dirichlet regression**
 - Dirichlet regression, [18](#)
- *Topic **Discriminant analysis**
 - Compositional-package, [3](#)
- *Topic **Equality of the covariance**

- matrices**
 - Hotelling's multivariate version of the 2 sample t-test for Euclidean data, [39](#)
 - Multivariate analysis of variance, [52](#)
- *Topic **Euclidean distance**
 - The alpha-distance, [84](#)
- *Topic **Frechet mean**
 - The Frechet mean for compositional data, [87](#)
- *Topic **GLM**
 - Tuning the principal components with GLMs, [100](#)
- *Topic **Gaussian mixture models**
 - Simulation of compositional data from Gaussian mixture models, [77](#)
 - Simulation of compositional data from the folded model, [78](#)
- *Topic **Gaussian mixtures**
 - Gaussian mixture models for compositional data, [35](#)
- *Topic **Gaussianmixture model**
 - Contour plot of a Gaussian mixture model in S^2 , [5](#)
- *Topic **Hypothesis test**
 - Empirical likelihood for a one sample mean vector hypothesis testing, [21](#)
 - Exponential empirical likelihood for a one sample mean vector hypothesis testing, [28](#)
 - Hotelling's multivariate version of the 1 sample t-test for Euclidean data, [38](#)
- *Topic **Kullback-Lebler divergence**
 - Divergence based regression for compositional data, [19](#)
 - Helper functions for the Kullback-Leibler regression, [37](#)
- *Topic **Kullback-Leibler divergence**
 - Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distributions, [45](#)
- *Topic **Log-likelihood ratio test**
 - Log-likelihood ratio test for a Dirichlet mean vector, [46](#)
 - Log-likelihood ratio test for a symmetric Dirichlet distribution, [47](#)
- *Topic **MLE**
 - Compositional-package, [3](#)
- *Topic **Manhattan distance**
 - The alpha-distance, [84](#)
- *Topic **Multivariate analysis of variance**
 - Hotelling's multivariate version of the 2 sample t-test for Euclidean data, [39](#)
 - James multivariate version of the t-test, [44](#)
 - Multivariate analysis of variance, [52](#)
 - Multivariate analysis of variance (James test), [53](#)
- *Topic **Multivariate hypothesis testing**
 - Empirical likelihood hypothesis testing for two mean vectors, [22](#)
 - Exponential empirical likelihood hypothesis testing for two mean vectors, [29](#)
 - Hypothesis testing for two or more compositional mean vectors, [41](#)
- *Topic **Newton-Raphson**
 - Fitting a Dirichlet distribution via Newton-Rapshon, [33](#)
- *Topic **No equality of the covariance matrices**
 - Multivariate analysis of variance (James test), [53](#)
- *Topic **Non parametric test**
 - Empirical likelihood hypothesis testing for two mean vectors, [22](#)
- *Topic **Regression**
 - Compositional-package, [3](#)
- *Topic **Regularised discriminant analysis**
 - Cross validation for the regularised discriminant analysis with compositional data using the

- alpha-transformation, 11
- Regularised discriminant analysis
 - for compositional data using the alpha-transformation, 69
- Regularised discriminant analysis
 - for Euclidean data, 70
- Tuning of the k-NN algorithm for compositional data, 93
- Tuning the parameters of the regularised discriminant analysis, 98
- *Topic **Ridge regression**
 - Cross validation for the ridge regression, 12
- *Topic **Tuning of the hyper-parameters**
 - Tuning the parameters of the regularised discriminant analysis, 98
- *Topic **Unequality of the covariance matrices**
 - James multivariate version of the t-test, 44
- *Topic **bandwidth tuning**
 - Tuning of the bandwidth h of the kernel using the maximum likelihood cross validation, 92
- *Topic **bivariate normal distribution**
 - Contour plot of the normal distribution in S^2 , 7
- *Topic **bivariate skew skewnormal distribution**
 - Contour plot of the skew skew-normal distribution in S^2 , 8
- *Topic **bivariate t distribution**
 - Contour plot of the t distribution in S^2 , 9
- *Topic **compositional data**
 - Hypothesis testing for two or more compositional mean vectors, 41
- *Topic **contour plot**
 - Contour plot of a Dirichlet distribution in S^2 , 4
 - Contour plot of a Gaussian mixture model in S^2 , 5
 - Contour plot of the kernel density estimate in S^2 , 6
 - Contour plot of the normal distribution in S^2 , 7
 - Contour plot of the skew skew-normal distribution in S^2 , 8
 - Contour plot of the t distribution in S^2 , 9
- *Topic **covariance matrix**
 - Spatial sign covariance matrix, 80
- *Topic **cross-validation**
 - Cross validation for the regularised discriminant analysis with compositional data using the alpha-transformation, 11
- *Topic **density values**
 - Density values of a Dirichlet distribution, 15
- *Topic **generalised linear models**
 - Principal component generalised linear models, 65
- *Topic **inverse transformation**
 - Inverse of the alpha-transformation, 42
- *Topic **k-NN algorithm**
 - The k-NN algorithm for compositional data, 89
- *Topic **kernel density estimate**
 - Multivariate kernel density estimation, 55
- *Topic **kernel density**
 - Contour plot of the kernel density estimate in S^2 , 6
 - Tuning of the bandwidth h of the kernel using the maximum likelihood cross validation, 92
- *Topic **linear regression**
 - Principal components regression, 66
 - Tuning of the principal components regression, 95
- *Topic **list(list(\alpha))**
 - Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation, 14
 - Estimation of the value of alpha in the folded model, 25
 - Estimation of the value of alpha

- via the profile
 - log-likelihood, [26](#)
- Fast estimation of the value of alpha, [31](#)
- Inverse of the
 - alpha-transformation, [42](#)
- MLE of the folded model for a given value of alpha, [51](#)
- Multivariate or univariate regression with compositional data in the covariates side using the
 - alpha-transformation, [58](#)
- Regression with compositional data using the
 - alpha-transformation, [67](#)
- Ridge regression with
 - compositional data in the covariates side using the alpha-transformation, [74](#)
- The alpha-distance, [84](#)
- The alpha-transformation, [85](#)
- The Frechet mean for
 - compositional data, [87](#)
- Tuning the number of PCs in the PCR with compositional data using the
 - alpha-transformation, [96](#)
- Tuning the value of alpha in the alpha-regression, [101](#)
- *Topic **location and scatter**
 - Estimating location and scatter parameters for compositional data, [24](#)
- *Topic **maximum likelihood cross validation**
 - Tuning of the bandwidth h of the kernel using the maximum likelihood cross validation, [92](#)
- *Topic **maximum likelihood estimation**
 - Estimation of the value of alpha in the folded model, [25](#)
 - Fitting a Dirichlet distribution, [32](#)
 - Fitting a Dirichlet distribution via Newton-Raphson, [33](#)
 - MLE of the folded model for a
 - given value of alpha, [51](#)
- *Topic **maximum log-likelihood estimation**
 - Fast estimation of the value of alpha, [31](#)
- *Topic **mean vector**
 - Empirical likelihood for a one sample mean vector hypothesis testing, [21](#)
 - Exponential empirical likelihood for a one sample mean vector hypothesis testing, [28](#)
 - Hotelling's multivariate version of the 1 sample t-test for Euclidean data, [38](#)
- *Topic **mixtures of normal distributions**
 - Mixture model selection via BIC, [49](#)
- *Topic **model based clustering**
 - Gaussian mixture models for compositional data, [35](#)
- *Topic **model selection**
 - Mixture model selection via BIC, [49](#)
- *Topic **multivariate kernel**
 - Multivariate kernel density estimation, [55](#)
- *Topic **multivariate linear regression**
 - Multivariate linear regression, [56](#)
- *Topic **multivariate normal distribution**
 - Multivariate normal random values simulation on the simplex, [57](#)
- *Topic **multivariate regression**
 - Dirichlet regression, [18](#)
 - Divergence based regression for compositional data, [19](#)
 - Helper functions for the Kullback-Leibler regression, [37](#)
 - Non linear least squares regression for compositional data, [64](#)
 - Regression with compositional data using the
 - alpha-transformation, [67](#)
 - Spatial median regression, [79](#)
- *Topic **multivariate regression**
 - Multivariate regression with

- compositional data, [60](#)
- *Topic **multivariate skew normal distribution**
 - Multivariate skew normal random values simulation on the simplex, [61](#)
- *Topic **multivariate t distribution**
 - MLE for the multivariate t distribution, [50](#)
 - Multivariate t random values simulation on the simplex, [62](#)
- *Topic **non parametric test**
 - Exponential empirical likelihood hypothesis testing for two mean vectors, [29](#)
 - Hypothesis testing for two or more compositional mean vectors, [41](#)
- *Topic **ordinary least squares**
 - Non linear least squares regression for compositional data, [64](#)
- *Topic **orthogonal matrix**
 - The Helmert sub-matrix, [88](#)
- *Topic **parameters tuning**
 - Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation, [14](#)
- *Topic **plot**
 - Ridge regression plot, [73](#)
 - Ridge regression with the alpha-transformation plot, [75](#)
- *Topic **principal components regression**
 - Multivariate or univariate regression with compositional data in the covariates side using the alpha-transformation, [58](#)
 - Principal component generalised linear models, [65](#)
 - Tuning the number of PCs in the PCR with compositional data using the alpha-transformation, [96](#)
- *Topic **principal components**
 - Principal components regression, [66](#)
 - Tuning of the principal components regression, [95](#)
 - Tuning the principal components with GLMs, [100](#)
- *Topic **profile log-likelihood**
 - Estimation of the value of alpha via the profile log-likelihood, [26](#)
- *Topic **random values simulation**
 - Dirichlet random values simulation, [17](#)
 - Multivariate normal random values simulation on the simplex, [57](#)
 - Multivariate skew normal random values simulation on the simplex, [61](#)
 - Multivariate t random values simulation on the simplex, [62](#)
- *Topic **random vectors simulation**
 - Simulation of compositional data from Gaussian mixture models, [77](#)
 - Simulation of compositional data from the folded model, [78](#)
- *Topic **regularisation**
 - Ridge regression, [72](#)
- *Topic **ridge regression**
 - Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation, [14](#)
 - Ridge regression, [72](#)
 - Ridge regression plot, [73](#)
 - Ridge regression with compositional data in the covariates side using the alpha-transformation, [74](#)
 - Ridge regression with the alpha-transformation plot, [75](#)
- *Topic **robust estimation**
 - Estimating location and scatter parameters for compositional data, [24](#)
- *Topic **robust statistics**
 - Spatial sign covariance matrix, [80](#)
- *Topic **spatial median regression**
 - Multivariate regression with compositional data, [60](#)
 - Spatial median regression, [79](#)

- *Topic **symmetric Dirichlet distribution**
 - Log-likelihood ratio test for a symmetric Dirichlet distribution, [47](#)
- *Topic **ternary plot**
 - Ternary diagram, [81](#)
- *Topic **tuning**
 - Tuning the number of PCs in the PCR with compositional data using the alpha-transformation, [96](#)
- *Topic **visualisation**
 - Ternary diagram, [81](#)
- *Topic
 - Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation, [14](#)
 - Estimation of the value of alpha in the folded model, [25](#)
 - Estimation of the value of alpha via the profile log-likelihood, [26](#)
 - Fast estimation of the value of alpha, [31](#)
 - Inverse of the alpha-transformation, [42](#)
 - MLE of the folded model for a given value of alpha, [51](#)
 - Multivariate or univariate regression with compositional data in the covariates side using the alpha-transformation, [58](#)
 - Regression with compositional data using the alpha-transformation, [67](#)
 - Ridge regression with compositional data in the covariates side using the alpha-transformation, [74](#)
 - The alpha-distance, [84](#)
 - The alpha-transformation, [85](#)
 - The Frechet mean for compositional data, [87](#)
 - Tuning the number of PCs in the PCR with compositional data using the alpha-transformation, [96](#)
 - Tuning the value of alpha in the alpha-regression, [101](#)
- a.est, [51](#), [52](#), [79](#), [85](#), [86](#)
- a.est (Estimation of the value of alpha in the folded model), [25](#)
- a.mle, [85](#)
- a.mle (MLE of the folded model for a given value of alpha), [51](#)
- alef (The alpha-transformation), [85](#)
- alfa, [26](#), [27](#), [32](#), [43](#), [52](#), [70](#), [71](#), [79](#), [82](#), [83](#), [85](#), [87](#), [88](#), [90](#), [91](#), [94](#), [98](#), [99](#), [103](#)
- alfa (The alpha-transformation), [85](#)
- alfa.knn (The k-NN algorithm for compositional data), [89](#)
- alfa.pcr, [66](#), [67](#), [96](#), [98](#), [101](#)
- alfa.pcr (Multivariate or univariate regression with compositional data in the covariates side using the alpha-transformation), [58](#)
- alfa.profile, [26](#), [31](#), [32](#), [52](#), [83](#), [86](#), [91](#)
- alfa.profile (Estimation of the value of alpha via the profile log-likelihood), [26](#)
- alfa.rda, [12](#), [69](#), [71](#), [99](#)
- alfa.rda (Regularised discriminant analysis for compositional data using the alpha-transformation), [69](#)
- alfa.reg, [19](#), [65](#), [80](#), [85](#), [103](#), [104](#)
- alfa.reg (Regression with compositional data using the alpha-transformation), [67](#)
- alfa.ridge, [15](#), [72–74](#), [76](#)
- alfa.ridge (Ridge regression with compositional data in the covariates side using the alpha-transformation), [74](#)
- alfa.tune, [27](#), [83](#), [86](#), [91](#)
- alfa.tune (Fast estimation of the value of alpha), [31](#)
- alfadist, [43](#)
- alfadist (The alpha-distance), [84](#)
- alfadista (The alpha-distance), [84](#)
- alfainv, [26](#), [27](#), [32](#), [52](#), [85–88](#)

- alfainv (Inverse of the alpha-transformation), 42
- alfaknn.tune (Tuning of the k-NN algorithm for compositional data), 93
- alfapcr.tune, 59, 66, 95, 96, 100, 101
- alfapcr.tune (Tuning the number of PCs in the PCR with compositional data using the alpha-transformation), 96
- alfarda.tune (Cross validation for the regularised discriminant analysis with compositional data using the alpha-transformation), 11
- alfareg.tune, 60, 68, 69
- alfareg.tune (Tuning the value of alpha in the alpha-regression), 101
- alfaridge.plot, 74, 75
- alfaridge.plot (Ridge regression with the alpha-transformation plot), 75
- alfaridge.tune, 13, 75
- alfaridge.tune (Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation), 14
- alpha.mle, 26, 79, 85, 86
- alpha.mle (MLE of the folded model for a given value of alpha), 51
- alr (The additive log-ratio transformation and its inverse), 83
- alrinv (The additive log-ratio transformation and its inverse), 83
- bic.mixcompnorm, 6, 36, 77
- bic.mixcompnorm (Mixture model selection via BIC), 49
- bivt.contour, 5, 7–9, 50, 51
- bivt.contour (Contour plot of the t distribution in S^2), 9
- comp.den, 50, 51, 58, 62, 63, 81, 82
- comp.den (Estimating location and scatter parameters for compositional data), 24
- comp.kerncontour, 5, 55, 82, 93
- comp.kerncontour (Contour plot of the kernel density estimate in S^2), 6
- comp.knn, 94
- comp.knn (The k-NN algorithm for compositional data), 89
- comp.reg, 19, 21, 38, 57, 65, 69, 79, 80
- comp.reg (Multivariate regression with compositional data), 60
- comp.test, 22, 23, 29, 30, 39, 40, 45, 53, 54
- comp.test (Hypothesis testing for two or more compositional mean vectors), 41
- compknn.tune, 90
- compknn.tune (Tuning of the k-NN algorithm for compositional data), 93
- Compositional-package, 3
- Contour plot of a Dirichlet distribution in S^2 , 4
- Contour plot of a Gaussian mixture model in S^2 , 5
- Contour plot of the kernel density estimate in S^2 , 6
- Contour plot of the normal distribution in S^2 , 7
- Contour plot of the skew skew-normal distribution in S^2 , 8
- Contour plot of the t distribution in S^2 , 9
- Cross validation for the regularised discriminant analysis with compositional data using the alpha-transformation, 11
- Cross validation for the ridge regression, 12
- Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation, 14
- ddiri, 33, 34, 47
- ddiri (Density values of a Dirichlet distribution), 15
- Density values of a Dirichlet distribution, 15
- diri.contour, 6–10, 16, 17, 33, 34, 82

- diri.contour (Contour plot of a Dirichlet distribution in S^2), 4
- diri.est, 16, 17, 34, 46–48
- diri.est (Fitting a Dirichlet distribution), 32
- diri.nr, 16, 17, 33, 46–48
- diri.nr (Fitting a Dirichlet distribution via Newton–Rapshon), 33
- diri.reg, 21, 38, 57, 61, 65, 69, 80, 104
- diri.reg (Dirichlet regression), 18
- diri.reg2 (Dirichlet regression), 18
- Dirichlet random values simulation, 17
- Dirichlet regression, 18
- dirimean.test, 48
- dirimean.test (Log-likelihood ratio test for a Dirichlet mean vector), 46
- Divergence based regression for compositional data, 19
- eel.test1, 22, 39
- eel.test1 (Exponential empirical likelihood for a one sample mean vector hypothesis testing), 28
- eel.test2, 23
- eel.test2 (Exponential empirical likelihood hypothesis testing for two mean vectors), 29
- el, 45
- el.test, 21
- el.test1, 29, 39
- el.test1 (Empirical likelihood for a one sample mean vector hypothesis testing), 21
- el.test2, 22, 29, 30, 39, 40, 42
- el.test2 (Empirical likelihood hypothesis testing for two mean vectors), 22
- Empirical likelihood for a one sample mean vector hypothesis testing, 21
- Empirical likelihood hypothesis testing for two mean vectors, 22
- Estimating location and scatter parameters for compositional data, 24
- Estimation of the value of alpha in the folded model, 25
- Estimation of the value of alpha via the profile log-likelihood, 26
- Exponential empirical likelihood for a one sample mean vector hypothesis testing, 28
- Exponential empirical likelihood hypothesis testing for two mean vectors, 29
- Fast estimation of the value of alpha, 31
- Fitting a Dirichlet distribution, 32
- Fitting a Dirichlet distribution via Newton–Rapshon, 33
- frechet (The Frechet mean for compositional data), 87
- Gaussian mixture models for compositional data, 35
- glm, 66, 98
- glm.pcr, 59, 67, 96, 101
- glm.pcr (Principal component generalised linear models), 65
- glm.pcr.tune, 96, 98
- glm.pcr.tune (Tuning the principal components with GLMs), 100
- helm (The Helmert sub-matrix), 88
- Helper functions for the Kullback–Leibler regression, 37
- hotel1T2, 22, 29
- hotel1T2 (Hotelling’s multivariate version of the 1 sample t-test for Euclidean data), 38
- hotel2T2, 22, 23, 29, 30, 39, 42, 45, 53, 54
- hotel2T2 (Hotelling’s multivariate version of the 2 sample t-test for Euclidean data), 39
- Hotelling’s multivariate version of the 1 sample t-test for Euclidean data, 38
- Hotelling’s multivariate version of the 2 sample t-test for Euclidean data, 39

- Hypothesis testing for two or more compositional mean vectors, [41](#)
- Inverse of the alpha-transformation, [42](#)
- james, [22](#), [23](#), [29](#), [30](#), [39](#), [40](#), [53](#), [54](#)
- james (James multivariate version of the t-test), [44](#)
- James multivariate version of the t-test, [44](#)
- js.compreg, [19](#), [21](#), [38](#), [57](#), [61](#), [65](#), [69](#), [80](#)
- js.compreg (Divergence based regression for compositional data), [19](#)
- kl.compreg, [19](#), [37](#), [57](#), [65](#), [69](#), [104](#)
- kl.compreg (Divergence based regression for compositional data), [19](#)
- kl.compreg2 (Helper functions for the Kullback-Leibler regression), [37](#)
- kl.diri (Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distributions), [45](#)
- klcompreg.boot (Helper functions for the Kullback-Leibler regression), [37](#)
- Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distributions, [45](#)
- lm, [56](#)
- lm.ridge, [72](#)
- Log-likelihood ratio test for a Dirichlet mean vector, [46](#)
- Log-likelihood ratio test for a symmetric Dirichlet distribution, [47](#)
- maov, [22](#), [23](#), [29](#), [30](#), [39](#), [40](#), [42](#), [54](#)
- maov (Multivariate analysis of variance), [52](#)
- maovjames, [23](#), [30](#), [42](#), [45](#), [53](#)
- maovjames (Multivariate analysis of variance (James test)), [53](#)
- mix.compnorm, [5](#), [6](#), [50](#), [77](#)
- mix.compnorm (Gaussian mixture models for compositional data), [35](#)
- mixnorm.contour, [5](#), [7–10](#), [36](#), [50](#)
- mixnorm.contour (Contour plot of a Gaussian mixture model in S^2), [5](#)
- mixreg (Zero adjusted Dirichlet regression), [103](#)
- Mixture model selection via BIC, [49](#)
- mkde, [93](#)
- mkde (Multivariate kernel density estimation), [55](#)
- mkde.tune, [55](#)
- mkde.tune (Tuning of the bandwidth h of the kernel using the maximum likelihood cross validation), [92](#)
- MLE for the multivariate t distribution, [50](#)
- MLE of the folded model for a given value of alpha, [51](#)
- multinompr.tune (Tuning the principal components with GLMs), [100](#)
- Multivariate analysis of variance, [52](#)
- Multivariate analysis of variance (James test), [53](#)
- Multivariate kernel density estimation, [55](#)
- Multivariate linear regression, [56](#)
- Multivariate normal random values simulation on the simplex, [57](#)
- Multivariate or univariate regression with compositional data in the covariates side using the alpha-transformation, [58](#)
- Multivariate regression with compositional data, [60](#)
- Multivariate skew normal random values simulation on the simplex, [61](#)
- Multivariate t random values simulation on the simplex, [62](#)
- multivreg, [61](#), [80](#)
- multivreg (Multivariate linear regression), [56](#)
- multivt, [25](#)
- multivt (MLE for the multivariate t distribution), [50](#)
- Non linear least squares regression

- for compositional data, [64](#)
- `norm.contour`, [5, 7, 9, 10](#)
- `norm.contour` (Contour plot of the normal distribution in S^2), [7](#)
- `ols.compreg`, [19, 21, 38, 57, 69, 104](#)
- `ols.compreg` (Non linear least squares regression for compositional data), [64](#)
- `pcr`, [59, 66](#)
- `pcr` (Principal components regression), [66](#)
- `pcr.tune`, [67, 98, 101](#)
- `pcr.tune` (Tuning of the principal components regression), [95](#)
- Principal component generalised linear models, [65](#)
- Principal components regression, [66](#)
- profile, [87, 98](#)
- `rcompnorm`, [62, 63](#)
- `rcompnorm` (Multivariate normal random values simulation on the simplex), [57](#)
- `rcompsn`, [58](#)
- `rcompsn` (Multivariate skew normal random values simulation on the simplex), [61](#)
- `rcompt`, [58](#)
- `rcompt` (Multivariate t random values simulation on the simplex), [62](#)
- `rda`, [70, 90, 94, 99](#)
- `rda` (Regularised discriminant analysis for Euclidean data), [70](#)
- `rda.tune`, [12, 71](#)
- `rda.tune` (Tuning the parameters of the regularised discriminant analysis), [98](#)
- `rdiri`, [16, 33, 34, 47, 48, 58, 62, 63](#)
- `rdiri` (Dirichlet random values simulation), [17](#)
- Regression with compositional data using the alpha-transformation, [67](#)
- Regularised discriminant analysis for compositional data using the alpha-transformation, [69](#)
- Regularised discriminant analysis for Euclidean data, [70](#)
- `rfolded` (Simulation of compositional data from the folded model), [78](#)
- Ridge regression, [72](#)
- Ridge regression plot, [73](#)
- Ridge regression with compositional data in the covariates side using the alpha-transformation, [74](#)
- Ridge regression with the alpha-transformation plot, [75](#)
- `ridge.plot`, [73, 76](#)
- `ridge.plot` (Ridge regression plot), [73](#)
- `ridge.reg`, [13, 74, 75](#)
- `ridge.reg` (Ridge regression), [72](#)
- `ridge.tune`, [15, 73, 74](#)
- `ridge.tune` (Cross validation for the ridge regression), [12](#)
- `rmixcomp`, [36, 50](#)
- `rmixcomp` (Simulation of compositional data from Gaussian mixture models), [77](#)
- `rmvt`, [51, 62, 63](#)
- Simulation of compositional data from Gaussian mixture models, [77](#)
- Simulation of compositional data from the folded model, [78](#)
- `skewnorm.contour`, [8, 10](#)
- `skewnorm.contour` (Contour plot of the skew skew-normal distribution in S^2), [8](#)
- `spat.med`, [25, 81](#)
- Spatial median regression, [79](#)
- Spatial sign covariance matrix, [80](#)
- `spatmed.reg`, [61](#)
- `spatmed.reg` (Spatial median regression), [79](#)
- `sscov`, [25](#)
- `sscov` (Spatial sign covariance matrix), [80](#)
- `sym.test`, [47](#)
- `sym.test` (Log-likelihood ratio test for a symmetric Dirichlet distribution), [47](#)
- ternary (Ternary diagram), [81](#)
- Ternary diagram, [81](#)

- The additive log-ratio transformation
and its inverse, [83](#)
- The alpha-distance, [84](#)
- The alpha-transformation, [85](#)
- The Frechet mean for compositional
data, [87](#)
- The Helmert sub-matrix, [88](#)
- The k-NN algorithm for compositional
data, [89](#)
- Total variability, [91](#)
- totvar(Total variability), [91](#)
- Tuning of the bandwidth h of the
kernel using the maximum
likelihood cross validation, [92](#)
- Tuning of the k-NN algorithm for
compositional data, [93](#)
- Tuning of the principal components
regression, [95](#)
- Tuning the number of PCs in the PCR
with compositional data using
the alpha-transformation, [96](#)
- Tuning the parameters of the
regularised discriminant
analysis, [98](#)
- Tuning the principal components with
GLMs, [100](#)
- Tuning the value of alpha in the
alpha-regression, [101](#)

- zadr (Zero adjusted Dirichlet
regression), [103](#)
- Zero adjusted Dirichlet regression, [103](#)