Package 'Directional'

May 24, 2022

Type Package

Title A Collection of Functions for Directional Data Analysis

Version 5.5

URL

Date 2022-05-24

Author

Michail Tsagris, Giorgos Athineou, Christos Adam, Anamul Sajib, Eli Amson, Micah J. Waldstein

Maintainer Michail Tsagris <mtsagris@uoc.gr>

Description A collection of functions for directional data (including massive data, with millions of observations) analysis. Hypothesis testing, discriminant and regression analysis, MLE of distributions and more are included. The standard textbook for such data is the "Directional Statistics" by Mardia, K. V. and Jupp, P. E. (2000). Other references include a) Phillip J. Paine, Simon P. Preston Michail Tsagris and Andrew T. A. Wood (2018). An elliptically symmetric angular Gaussian distribution. Statistics and Computing 28(3): 689-697. <doi:10.1007/s11222-017-9756-4>. b) Tsagris M. and Alenazi A. (2019). Comparison of discriminant analysis methods on the sphere. Communications in Statistics: Case Studies, Data Analysis and Applications 5(4):467--491. <doi:10.1080/23737484.2019.1684854>. c) P. J. Paine, S. P. Preston, M. Tsagris and Andrew T. A. Wood (2020). Spherical regression models with general covariates and anisotropic errors. Statistics and Computing 30(1): 153--165. <doi:10.1007/s11222-019-09872-2>. d) Tsagris M. and Alenazi A. (2022). An investigation of hypothesis testing procedures for circular and spherical mean vectors. Communications in Statistics-Simulation and Computation (Accepted for publication). <doi:10.1080/03610918.2022.2045499>.

License GPL-2

Imports bigstatsr, parallel, doParallel, foreach, ggplot2, grDevices, magrittr, RANN, Rfast, Rfast2, rgl, rnaturalearth, sf

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2022-05-24 08:20:02 UTC

R topics documented:

Directional-package	4
A test for testing the equality of the concentration parameters for circular data	5
Angular central Gaussian random values simulation	7
Anova for (hyper-)spherical data	8
Anova for circular data	9
BIC for the model based clustering using mixtures of von Mises-Fisher distributions	10
Bootstrap 2-sample mean test for (hyper-)spherical data	11
Bootstrap 2-sample mean test for circular data	13
Check visually whether matrix Fisher samples is correctly generated or not	14
Circular correlations between one and many circular variables	15
Circular correlations between two circular variables	16
Circular or angular regression	17
Circular-linear correlation	19
Column-wise MLE of the angular Gaussian and the von Mises Fisher distributions	20
Column-wise uniformity Watson test for circular data	21
Contour plot of a mixture of von Mises-Fisher distributions model	22
Contour plot of spherical data using a von Mises-Fisher kernel density estimate	23
Contour plot of the Kent distribution for some data	24
Contour plot of the Kent distribution without any data	25
Contour plots of the von Mises-Fisher distribution	27
Conversion of cosines to azimuth and plunge	28
Converting a rotation matrix on SO(3) to an unsigned unit quaternion	29
Converting an unsigned unit quaternion to rotation matrix on SO(3)	30
Cross validation for estimating the classification rate	31
Cross validation in von Mises-Fisher discrminant analysis	33
Cross validation with ESAG discrminant analysis	34
Density of some (hyper-)spherical distributions	35
Density of some circular distributions	37
Density of the spherical Kent and ESAG distributions	38
Euclidean transformation	39
Euler angles from a rotation matrix on SO(3)	40
Forward Backward Early Dropping selection for circular data using the SPML regression	42
Generate random folds for cross-validation	43
Generation of unit vector(s) with a given angle	44
Goodness of fit test for grouped data	45
Habeck's rotation matrix generation	46
Haversine distance matrix	47
Hypothesis test for IAG distribution over the ESAG distribution	48
Hypothesis test for von Mises-Fisher distribution over Kent distribution	49
Interactive 3D plot of spherical data	51
Inverse of Lambert's equal area projection	52
Inverse of the Euclidean transformation	53
k-NN algorithm using the arc cosinus distance	54
k-NN regression	55
Lambert's equal area projection	57
Logarithm of the Kent distribution normalizing constant	58

Many simple circular or angular regressions	. 59
Maps of the world and the continents	. 60
Mixtures of Von Mises-Fisher distributions	. 61
MLE of (hyper-)spherical distributions	. 62
MLE of some circular distributions	
MLE of some circular distributions with multiple samples	
MLE of the ESAG distribution	
MLE of the Kent distribution	
MLE of the Matrix Fisher distribution on SO(3)	
MLE of the Purkayashta distribution	
MLE of the Wood bimodal distribution on the sphere	
Naive Bayes classifiers for circular data	
Normalised spatial median for directional data	
Permutation based 2-sample mean test for (hyper-)spherical data	
Permutation based 2-sample mean test for circular data	
Prediction in discriminant analysis based on ESAG distribution	
Prediction in discriminant analysis based on von Mises-Fisher distribution	
Prediction with some naive Bayes classifiers for circular data	
Probability density function of the von Mises-Fisher distribution	
Random sample of matrices in SO(p)	
Rayleigh's test of uniformity	
Read a file as a Filebacked Big Matrix	
Rotation axis and angle of rotation given a rotation matrix	
Rotation matrix from a rotation axis and angle of rotation	
Rotation matrix on SO(3) from three Euler angles	
Rotation matrix to rotate a spherical vector along the direction of another	
Saddlepoint approximations of the Fisher-Bingham distributions	
Simulation from a Bingham distribution using any symmetric matrix A	
Simulation from a Matrix Fisher distribution on SO(3)	
Simulation of random values from a Bingham distribution	
Simulation of random values from a mixture of von Mises-Fisher distributions	
Simulation of random values from a spherical Fisher-Bingham distribution	
Simulation of random values from a spherical Kent distribution	
Simulation of random values from rotationally symmetric distributions	
Simulation of random values from some circular distributions	
Spherical and hyperspherical median	
Spherical regression using the projected normal or the von Mises-Fisher distribution	
Spherical-spherical correlation	
Spherical-spherical regression	
Summary statistics for circular data	
Summary statistics for grouped circular data	
Test for a given mean direction	
Test for equality of concentration parameters for spherical data	
Test of equality of the concentration parameters for circular data	
The k-nearest neighbours using the cosinus distance	
Transform unit vectors to angular data	
Tuning of the handwidth parameter in the von Mises kernel	

4 Directional-package

,	Tuning of the k-NN Tuning of the k-NN	Adwidth parameter in the von Mises-Fisher kernel	
	von Mises kernel d	lensity estimation	. 124
	von Mises-Fisher k	ternel density estimation for (hyper-)spherical data	. 125
Index			127
		This is an R package that provides methods for the statistical analy,	

Description

Circular-linear regression, spherical-spherical regression, spherical regression, discriminant analysis, ANOVA for circular and (hyper-)spherical data, tests for eaquality of conentration parameters, fitting distributions, random values generation, contour plots and many more functions are included.

Details

Package: Directional
Type: Package
Version: 5.5
Date: 2022-05-24
License: GPL-2

Maintainers

Michail Tsagris <mtsagris@uoc.gr>

Note

Acknowledgments:

Professor Andy Wood and Dr Simon Preston from the university of Nottingham are highly appreciated for being my supervisors during my post-doc in directional data analysis.

Dr Georgios Pappas (former postDoc at the university of Nottingham) helped me construct the contour plots of the von Mises-Fisher and the Kent distribution.

Dr Christopher Fallaize and Dr Theo Kypraios from the university of Nottingham have provided a function for simulating from the Bingham distribution using rejection sampling. So any questions regarding this function should be addressed to them.

Dr Kwang-Rae Kim (post-doc at the university of Nottingham) answered some of my questions.

Giorgos Borboudakis (PhD student at the university of Crete) pointed out to me a not so clear message in the algorithm of generating random values from the von Mises-Fisher distribution.

Panagiotis (pronounced Panayiotis) Tzirakis (master student at the department of computer science in Heraklion during the 2013-2015 seasons) showed me how to perform parallel computing in R and he is greatly acknowledged and appreciated not only from me but from all the readers of this document. He also helped me with the vectorization of some contour plot functions.

Professor John Kent from the university of Leeds is acknowledged for clarifying one thing with the ovalness parameter in his distribution.

Phillip Paine (postdoc at the university of Nottingham) spotted that the function rfb is rather slow and he suggested me to change it. The function has changed now and this is also due to Joshua Davis (from Carleton College, Northfield, MN) who spotted that mistakes could occur, due a vector not being a matrix.

Professor Kurt Hornik from the Vienna university of economics and business is greatly acknowledged for his patience and contast help with this (and not only) R package.

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

Dr Mojgan Golzy spotted a mistake in the desag and Michail is very happy for that.

Lisette de Jonge-Hoekstra from the University of Groningen found a wrong sentence in the help file of spml.reg which is now deleted.

Peter Harremoes from the Copenhagen Business College spotted a mistake in the confidence interval of the circ.summary which has now been corrected.

If you want more information on many of these algorithms see Chapters 9 and 10 in the following document. https://www.researchgate.net/publication/324363311_Multivariate_data_analysis_in_R

Author(s)

Michail Tsagris <mtsagris@uoc.gr>, Giorgos Athineou <gioathineou@gmail.com>, Christos Adam <pada4m4@gmail.com>, Anamul Sajib <sajibstat@du.ac.bd>, Eli Amson <eli.amson1988@gmail.com> and Micah J. Waldstein <micah@waldste.in>.

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley and Sons.

A test for testing the equality of the concentration parameters for circular data A test for testing the equality of the concentration parameter among g samples, where g >= 2 for circular data

Description

A test for testing the equality of the concentration parameter among g samples, where $g \ge 2$ for circular data. It is a tangential approach.

Usage

```
tang.conc(u, ina, rads = FALSE)
```

Arguments

u A numeric vector containing the values of all samples.

ina A numerical variable or factor indicating the groups of each value.

rads If the data are in radians this should be TRUE and FALSE otherwise.

Details

This test works for circular data.

Value

A vector including:

test The value of the test statistic.

p-value The p-value of the test.

Author(s)

Michail Tsagris

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons. Fisher, N. I. (1995). Statistical analysis of circular data. Cambridge University Press.

See Also

```
embed.circaov, hcf.circaov, lr.circaov, het.circaov, conc.test
```

```
x <- rvonmises(100, 2.4, 15)
ina <- rep(1:4,each = 25)
tang.conc(x, ina, rads = TRUE)</pre>
```

Angular central Gaussian random values simulation

Angular central Gaussian random values simulation

Description

Angular central Gaussian random values simulation.

Usage

```
racg(n, sigma)
```

Arguments

n The sample size, a numerical value. sigma The covariance matrix in \mathbb{R}^d .

Details

The algorithm uses univariate normal random values and transforms them to multivariate via a spectral decomposition. The vectors are then scaled to have unit length.

Value

A matrix with the simulated data.

Author(s)

Michail Tsagris.

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

References

Tyler D. E. (1987). Statistical analysis for the angular central Gaussian distribution on the sphere. Biometrika 74(3): 579-589.

See Also

```
acg.mle, rvmf, rvonmises
```

```
s <- cov( iris[, 1:4] )
x <- racg(100, s)
Directional::acg.mle(x)
Directional::vmf.mle(x)
## the concentration parameter, kappa, is very low, close to zero, as expected.</pre>
```

```
Anova for (hyper-)spherical data

Analysis of variance for (hyper-)spherical data
```

Description

Analysis of variance for (hyper-)spherical data.

Usage

```
hcf.aov(x, ina, fc = TRUE)
hclr.aov(x, ina)
lr.aov(x, ina)
embed.aov(x, ina)
het.aov(x, ina)
```

Arguments

A matrix with the data in Euclidean coordinates, i.e. unit vectors.
 A numerical variable or a factor indicating the group of each vector.
 A boolean that indicates whether a corrected F test should be used or not.

Details

The high concentration (hcf.aov), high concentration likelihood ratio (hclr.aov), log-likelihood ratio (lr.aov), embedding approach (embed.aov) or the non equal concentration parameters approach (het.aov) is used.

Value

A vector with two or three elements, the test statistic, the p-value and the common concentration parameter kappa based on all the data.

Author(s)

Michail Tsagris.

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

References

Mardia K. V. and Jupp P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

Rumcheva P. and Presnell B. (2017). An improved test of equality of mean directions for the Langevin-von Mises-Fisher distribution. Australian & New Zealand Journal of Statistics, 59(1), 119-135.

Tsagris M. and Alenazi A. (2022). An investigation of hypothesis testing procedures for circular and spherical mean vectors. Communications in Statistics-Simulation and Computation (Accepted for publication).

Anova for circular data 9

See Also

```
hcf.boot, spherconc.test, conc.test, hclr.circaov,
```

Examples

```
x <- rvmf(60, rnorm(3), 15)
ina <- rep(1:3, each = 20)
hcf.aov(x, ina)
hcf.aov(x, ina, fc = FALSE)
lr.aov(x, ina)
embed.aov(x, ina)
het.aov(x, ina)</pre>
```

Anova for circular data

Analysis of variance for circular data

Description

Analysis of variance for circular data.

Usage

```
hcf.circaov(u, ina, rads = FALSE)
hclr.circaov(u, ina, rads = FALSE)
lr.circaov(u, ina, rads = FALSE)
het.circaov(u, ina, rads = FALSE)
embed.circaov(u, ina, rads = FALSE)
```

Arguments

u A numeric vector containing the data.

ina A numerical or factor variable indicating the group of each value.

rads If the data are in radians, this should be TRUE and FALSE otherwise.

Details

The high concentration (hcf.circaov), high concentration likelihood ratio (hclr.aov), log-likelihood ratio (lr.circaov), embedding approach (embed.circaov) or the non equal concentration parameters approach (het.circaov) is used.

Value

A vector including:

test The value of the test statistic.
p-value The p-value of the test.

kappa The concentration parameter based on all the data. If the het.circaov is used this

argument is not returned.

Author(s)

Michail Tsagris.

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

Rumcheva P. and Presnell B. (2017). An improved test of equality of mean directions for the Langevin-von Mises-Fisher distribution. Australian & New Zealand Journal of Statistics, 59(1), 119-135.

Tsagris M. and Alenazi A. (2022). An investigation of hypothesis testing procedures for circular and spherical mean vectors. Communications in Statistics-Simulation and Computation (Accepted for publication).

See Also

```
conc.test, hclr.aov, hcfcirc.boot
```

Examples

```
x <- rvonmises(100, 2.4, 15)
ina <- rep(1:4,each = 25)
hcf.circaov(x, ina, rads = TRUE)
lr.circaov(x, ina, rads = TRUE)
het.circaov(x, ina, rads = TRUE)
embed.circaov(x, ina, rads = TRUE)</pre>
```

BIC for the model based clustering using mixtures of von Mises-Fisher distributions

BIC to choose the number of components in a model based clustering

using mixtures of von Mises-Fisher distributions

Description

BIC to choose the number of components in a model based clustering using mixtures of von Mises-Fisher distributions

Usage

```
bic.mixvmf(x, A, n.start = 20)
```

Arguments

x A matrix containing directional data.

A The maximum number of clusters to be tested. Default value is 3.

n.start The number of random starts to try. See also R's built-in function kmeans for more information about this.

Details

If the data are not unit vectors, they are transformed into unit vectors.

Value

A plot of the BIC values and a list including:

BIC The BIC values for all the models tested.

runtime The run time of the algorithm. A numeric vector. The first element is the user

time, the second element is the system time and the third element is the elapsed

time.

Author(s)

Michail Tsagris

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

References

Hornik, K. and Grun, B. (2014). movMF: An R package for fitting mixtures of von Mises-Fisher distributions. Journal of Statistical Software, 58(10):1–31.

See Also

```
mixvmf.mle, rmixvmf, mixvmf.contour
```

Examples

```
x <- as.matrix( iris[, 1:4] )
x <- x / sqrt( rowSums(x^2) )
bic.mixvmf(x, 5)</pre>
```

```
Bootstrap 2-sample mean test for (hyper-)spherical data 
 Bootstrap 2-sample mean test for (hyper-)spherical data
```

Description

Bootstrap 2-sample mean test for (hyper-)spherical data.

Usage

```
hcf.boot(x1, x2, fc = TRUE, B = 999)
lr.boot(x1, x2, B = 999)
hclr.boot(x1, x2, B = 999)
embed.boot(x1, x2, B = 999)
het.boot(x1, x2, B = 999)
```

Arguments

x1	A matrix with the data in Euclidean coordinates, i.e. unit vectors.
x2	A matrix with the data in Euclidean coordinates, i.e. unit vectors.
fc	A boolean that indicates whether a corrected F test should be used or not.
В	The number of bootstraps to perform.

Details

The high concentration (hcf.boot), log-likelihood ratio (lr.boot), high concentration log-likelihood ratio (hclr.boot), embedding approach (embed.boot) or the non equal concentration parameters approach (het.boot) is used.

Value

A vector including two or three numbers, the test statistic value, the bootstrap p-value of the test and the common concentration parameter kappa based on all the data.

Author(s)

Michail Tsagris.

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

Rumcheva P. and Presnell B. (2017). An improved test of equality of mean directions for the Langevin-von Mises-Fisher distribution. Australian & New Zealand Journal of Statistics, 59(1), 119-135.

Tsagris M. and Alenazi A. (2022). An investigation of hypothesis testing procedures for circular and spherical mean vectors. Communications in Statistics-Simulation and Computation (Accepted for publication).

See Also

```
hcf.aov, hcf.perm, spherconc.test, conc.test
```

```
x <- rvmf(60, rnorm(3), 15)
ina <- rep(1:2, each = 30)
x1 <- x[ina == 1, ]
x2 <- x[ina == 2, ]
hcf.boot(x1, x2)
lr.boot(x1, x2)
het.boot(x1, x2)</pre>
```

Bootstrap 2-sample mean test for circular data

Bootstrap 2-sample mean test for circular data

Description

Bootstrap 2-sample mean test for circular data.

Usage

```
hcfcirc.boot(u1, u2, rads = TRUE, B = 999)
lrcirc.boot(u1, u2, rads = TRUE, B = 999)
hclrcirc.boot(u1, u2, rads = TRUE, B = 999)
embedcirc.boot(u1, u2, rads = TRUE, B = 999)
hetcirc.boot(u1, u2, rads = TRUE, B = 999)
```

Arguments

u1	A numeric vector containing the data of the first sample.
u2	A numeric vector containing the data of the first sample.
rads	If the data are in radians, this should be TRUE and FALSE otherwise.
В	The number of bootstraps to perform.

Details

The high concentration (hcfcirc.boot), the log-likelihood ratio test (lrcirc.boot), high concentration log-likelihood ratio (hclrcirc.boot), embedding approach (embedcirc.boot), or the non equal concentration parameters approach (hetcirc.boot) is used.

Value

A vector including two or three numbers, the test statistic value, the bootstrap p-value of the test and the common concentration parameter kappa based on all the data.

Author(s)

Michail Tsagris.

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

Rumcheva P. and Presnell B. (2017). An improved test of equality of mean directions for the Langevin-von Mises-Fisher distribution. Australian & New Zealand Journal of Statistics, 59(1), 119-135.

Tsagris M. and Alenazi A. (2022). An investigation of hypothesis testing procedures for circular and spherical mean vectors. Communications in Statistics-Simulation and Computation (Accepted for publication).

See Also

```
hcf.circaov, het.aov
```

Examples

```
u1 <- rvonmises(20, 2.4, 5)
u2 <- rvonmises(20, 2.4, 10)
hcfcirc.boot(u1, u2)
```

Check visually whether matrix Fisher samples is correctly generated or not Check visually whether matrix Fisher samples is correctly generated or not.

Description

It plots the log probability trace of matrix Fisher distribution which should close to the maximum value of the logarithm of matrix Fisher distribution, if samples are correctly generated.

Usage

```
visual.check(x, Fa)
```

Arguments

x The simulated data. An array with at least 2 3x3 matrices.

Fa An arbitrary 3x3 matrix represents the parameter matrix of this distribution.

Details

For a given parameter matrix Fa, maximum value of the logarithm of matrix Fisher distribution is calculated via the form of singular value decomposition of $Fa = U\Lambda V^T$ which is $tr(\Lambda)$. Multiply the last column of U by -1 and replace small eigenvalue, say, λ_3 by $-\lambda_3$ if $|UV^T| = -1$.

Value

A plot which shows log probability trace of matrix Fisher distribution. The values are also returned.

Author(s)

Anamul Sajib<sajibstat@du.ac.bd>

R implementation and documentation: Anamul Sajib<sajibstat@du.ac.bd>

Habeck M. (2009). Generation of three-dimensional random rotations in fitting and matching problems. Computational Statistics, 24(4):719–731.

Examples

```
Fa <- matrix( c(85, 11, 41, 78, 39, 60, 43, 64, 48), ncol = 3) / 10 x <- rmatrixfisher(1000, Fa) a <- visual.check(x, Fa)
```

Circular correlations between one and many circular variables

Circular correlations between two circular variables

Description

Circular correlations between two circular variables.

Usage

```
circ.cors1(theta, phi)
```

Arguments

theta The first cirular variable expressed in radians, not degrees.

phi The other cirular variable. In the case of "circ.cors1" this is a matrix with many

circular variables. In either case, the values must be in radians, not degrees.

Details

Correlation for circular variables using the cosinus and sinus formula of Jammaladaka and Sen-Gupta (1988).

Value

A matrix with two columns, the correlations and the p-values.

Author(s)

Michail Tsagris

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

References

Jammalamadaka, R. S. and Sengupta, A. (2001). Topics in circular statistics. World Scientific.

Jammalamadaka, S. R. and Sarma, Y. R. (1988). A correlation coefficient for angular variables. Statistical Theory and Data Analysis, 2:349–364.

See Also

```
spml.reg
```

Examples

```
y <- runif(50, 0, 2 * pi)
x <- matrix(runif(50 * 10, 0, 2 * pi), ncol = 10)
circ.cors1(y, x)</pre>
```

Circular correlations between two circular variables

Circular correlations between two circular variables

Description

Circular correlations between two circular variables.

Usage

```
circ.cor1(theta, phi, rads = FALSE)
circ.cor2(theta, phi, rads = FALSE)
```

Arguments

theta The first cirular variable.

phi The other cirular variable.

rads If the data are expressed in rads, then this should be TRUE. If the data are in

degrees, then this is FALSE.

Details

circ.cor1: Correlation for circular variables using the cosinus and sinus formula of Jammaladaka and SenGupta (1988).

circ.cor2: Correlation for circular variables using the cosinus and sinus formula of Mardia and Jupp (2000).

Value

A vector including:

rho The value of the correlation coefficient.

p-value The p-value of the zero correlation hypothesis testing.

Author(s)

Michail Tsagris

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

References

Jammalamadaka, R. S. and Sengupta, A. (2001). Topics in circular statistics. World Scientific.

Jammalamadaka, S. R. and Sarma, Y. R. (1988) . A correlation coefficient for angular variables. Statistical Theory and Data Analysis, 2:349–364.

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

See Also

```
circlin.cor, circ.cor2, spml.reg
```

Examples

```
y <- runif(50, 0, 2 * pi)
x <- runif(50, 0, 2 * pi)
circ.cor1(x, y, rads = TRUE)
circ.cor2(x, y, rads = TRUE)</pre>
```

Circular or angular regression

Circular or angular regression

Description

Regression with circular dependent variable and Euclidean or categorical independent variables.

Usage

```
spml.reg(y, x, rads = TRUE, xnew = NULL, seb = FALSE, tol = 1e-07)
```

have no new x values, leave it NULL (default).

Arguments

У	The dependent variable, a numerical vector, it can be in radians or degrees.
x	The independent variable(s). Can be Euclidean or categorical (factor variables).
rads	If the dependent variable is expressed in rads, this should be TRUE and FALSE otherwise.
xnew	The new values of some independent variable(s) whose circular values you want to predict. Can be Euclidean or categorical. If they are categorical, the user must provide them as dummy variables. It does not accept factor variables. If you

seb a boolean variable. If TRUE, the standard error of the coefficients will be be

returned. Set to FALSE in case of simulation studies or in other cases such as a forward regression setting for example. In these cases, it can save some time.

tol The tolerance value to terminate the Newton-Raphson algorithm.

Details

The Newton-Raphson algorithm is fitted in this regression as described in Presnell et al. (1998).

Value

A list including:

runtime The runtime of the procedure.

iters The number of iterations required until convergence of the EM algorithm.

beta The regression coefficients.

seb The standard errors of the coefficients.

loglik The value of the maximised log-likelihood.

est The fitted values expressed in radians if the observeed data are in radians and in

degrees otherwise. If xnew is not NULL, i.e. if you have new x values, then the

predicted values of y will be returned.

Author(s)

Michail Tsagris

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

References

Presnell Brett, Morrison Scott P. and Littell Ramon C. (1998). Projected multivariate linear models for directional data. Journal of the American Statistical Association, 93(443): 1068-1077.

See Also

```
circlin.cor, circ.cor1, circ.cor2, spher.cor, spher.reg
```

```
x <- rnorm(100)
z <- cbind(3 + 2 * x, 1 -3 * x)
y <- cbind( rnorm(100,z[ ,1], 1), rnorm(100, z[ ,2], 1) )
y <- y / sqrt( rowSums(y^2) )
y <- ( atan( y[, 2] / y[, 1] ) + pi * I(y[, 1] < 0) ) %% (2 * pi)
spml.reg(y, x, rads = TRUE)</pre>
```

Circular-linear correlation 19

Circular-linear correlation

Circular-linear correlation

Description

It calculates the squared correlation between a circular and one or more linear variables.

Usage

```
circlin.cor(theta, x, rads = FALSE)
```

Arguments

theta The circular variable.

x The linear variable or a matrix containing many linear variables.

rads If the circualr variable is in rads, this should be TRUE and FALSE otherwise.

Details

The squared correlation between a circular and one or more linear variables is calculated.

Value

A matrix with as many rows as linear variables including:

R-squared The value of the squared correlation.

p-value The p-value of the zero correlation hypothesis testing.

Author(s)

Michail Tsagris

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

See Also

```
circ.cor1, circ.cor2, spml.reg
```

Examples

```
phi <- rvonmises(50, 2, 20, rads = TRUE) x <- 2 * phi + rnorm(50) y <- matrix(rnorm(50 * 5), ncol = 5) circlin.cor(phi, x, rads = TRUE) circlin.cor(phi, y, rads = TRUE)
```

Column-wise MLE of the angular Gaussian and the von Mises Fisher distributions

Column-wise MLE of the angular Gaussian and the von Mises Fisher distributions

Description

Column-wise MLE of the angular Gaussian and the von Mises Fisher distributions.

Usage

```
colspml.mle(x ,tol = 1e-07, maxiters = 100, parallel = FALSE) colvm.mle(x, tol = 1e-07)
```

Arguments

х	A numerical matrix with data. Each column refers to a different vector of observations of the same distribution. The values of for Lognormal must be greater than zero, for the logitnormal they must by percentages, exluding 0 and 1, whereas for the Borel distribution the x must contain integer values greater
	than 1.
tol	The tolerance value to terminate the Newton-Raphson algorithm.
maxiters	The maximum number of iterations that can take place in each regression.
parallel	Do you want this to be executed in parallel or not. The parallel takes place in C++, and the number of threads is defined by each system's available cores.

Details

For each column, spml.mle function is applied that fits the angular Gaussian distribution estimates its parameters and computes the maximum log-likelihood.

Value

A matrix with four columns. The first two are the mean vector, then the γ parameter, and the fourth column contains maximum log-likelihood.

Author(s)

Michail Tsagris

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

Presnell Brett, Morrison Scott P. and Littell Ramon C. (1998). Projected multivariate linear models for directional data. Journal of the American Statistical Association, 93(443): 1068-1077.

See Also

```
spml.mle, spml.reg, vm.mle, vmf.mle
```

Examples

```
x <- matrix( runif(100 * 10), ncol = 10)
a <- colspml.mle(x)
b <- colvm.mle(x)
x <- NULL</pre>
```

Column-wise uniformity Watson test for circular data

Column-wise uniformity tests for circular data

Description

Column-wise uniformity tests for circular data.

Usage

```
colwatsons(u, rads = FALSE)
```

Arguments

u A numeric matrix containing the circular data which are expressed in radians.

Each column is a different sample.

rads A boolean variable. If the data are in radians, put this TRUE. If the data are

expressed in degrees make this FALSE.

Details

These tests are used to test the hypothesis that the data come from a circular uniform distribution.

Value

A matrix with two columns, the value of the test statistic and its associated p-value.

Author(s)

Michail Tsagris.

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

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, pg. 156-157.

See Also

```
watson, kuiper, fishkent
```

Examples

```
x \leftarrow matrix( rvonmises(n = 50 * 10, m = 2, k = 0), ncol = 10 ) res<-colwatsons(x) x <- NULL
```

Contour plot of a mixture of von Mises-Fisher distributions model

Contour plot of a mixture of von Mises-Fisher distributions model for spherical data only.

Description

Contour lines are produced of mixture model for spherical data only.

Usage

```
mixvmf.contour(u, mod)
```

Arguments

u A two column matrix. The first column is the longitude and the second is the

latitude.

mod This is mix.vmf object, actually it is a list. Run a mixture model and save it as

mod for example, mod = mix.vmf(x, 3).

Details

The contour plot is displayed with latitude and longitude in the axes. No Lambert projection is used here. This works for spherical data only which are given as longitude and latitude.

Value

A plot including: The points and the contour lines.

Author(s)

Michail Tsagris and Christos Adam.

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

Kurt Hornik and Bettina Grun (2014). movMF: An R Package for Fitting Mixtures of von Mises-Fisher Distributions http://cran.r-project.org/web/packages/movMF/vignettes/movMF.pdf

See Also

```
vmf.kerncontour, vmf.contour, mixvmf.mle
```

Examples

```
k <- runif(2, 4, 20)
prob <- c(0.4, 0.6)
mu <- matrix( rnorm(6), ncol = 3 )
mu <- mu / sqrt( rowSums(mu^2) )
x <- rmixvmf(200, prob, mu, k)$x
mod <- mixvmf.mle(x, 2)
y <- euclid.inv(x)
mixvmf.contour(y, mod)</pre>
```

Contour plot of spherical data using a von Mises-Fisher kernel density estimate

Contour plot of spherical data using a von Mises-Fisher kernel density

estimate

Description

Contour plot of spherical data using a von Mises-Fisher kernel density estimate.

Usage

```
vmf.kerncontour(u, thumb = "none", den.ret = FALSE, full = FALSE, ngrid = 100)
```

Arguments

u	A two column matrix. The first coolumn is the latitude and the second is the longitude.
thumb	This is either 'none' (defualt), or 'rot' for the rule of thumb suggested by Garcia-Portugues (2013). If it is "none" it is estimated via cross validation, with the fast function vmfkde.tune.
den.ret	If FALSE (default), plots the contours of the density along with the individual points. If TRUE, will instead return a list with the Longitudes, Latitudes and Densities. Look at the 'value' section for details.
full	If FALSE (default), uses the range of positions from 'u' to calculate and optionally plot densities. If TRUE, calculates densities covering the entire sphere.
ngrid	Sets the resolution of the density calculation.

Details

It calculates the contour plot using a von Mises-Fisher kernel for spherical data only.

Value

The contour lines of the data. If "den.ret" was set to TRUE a list including:

lat The latitude values.long The longitude values.h The optimal bandwidth.

den The kernel density estimate contour points.

Author(s)

Michail Tsagris, Micah J. Waldstein and Christos Adam.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>, Micah J. Waldstein <micah@waldste.in> and Christos Adam <pada4m4@gmail.com>.

References

Garcia Portugues, E. (2013). Exact risk improvement of bandwidth selectors for kernel density estimation with directional data. Electronic Journal of Statistics, 7, 1655–1685.

See Also

```
vmf.kde, vmfkde.tune, vmf.contour, kent.datacontour
```

Examples

```
x <- rvmf(100, rnorm(3), 15)
x <- euclid.inv(x)
## Not run:
vmf.kerncontour(x, "rot")
## End(Not run)</pre>
```

Contour plot of the Kent distribution for some data

Contour plot of the Kent distribution for some data

Description

The contour plot of the Kent distribution on the sphere for some data is produced.

Usage

```
kent.datacontour(x)
```

Arguments

Χ

A two column matrix, where the first column is the latitude and the second comlumn is the longitude. If the matrix has two columns, it is assumed to have unit vectors and in this case it is turned into latitude and longitude.

Details

MLE of the parameters of the Kent distribution are calculated, then the contour plot is plotted using these estimates and finally the data are also plotted.

Value

A plot containing the contours of the distribution along with the data.

Author(s)

Michail Tsagris and Christos Adam.

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

References

Kent John (1982). The Fisher-Bingham distribution on the sphere. Journal of the Royal Statistical Society, Series B, 44(1): 71-80.

See Also

```
kent.contour, kent.mle, vmf.kerncontour
```

Examples

```
x <- rvmf(100, rnorm(3), 10)
kent.mle(x)
y <- euclid.inv(x)
## Not run:
kent.datacontour(y)
## End(Not run)</pre>
```

Contour plot of the Kent distribution without any data

Contour plot of the Kent distribution without any data

Description

The contour plot of the Kent distribution on the sphere is produced. The user can see how the shape and ovalness change as he/she changes the ovlaness parameter.

Usage

```
kent.contour(k, b)
```

Arguments

- k The concentration parameter.
- b The ovalness parameter. It has to be less than k/2 in order for the distribution to be unimodal. Otherwise it is bimodal.

Details

The goal of this function is for the user to see hwo the Kent distribution looks like.

Value

A plot containing the contours of the distribution.

Author(s)

Michail Tsagris and Christos Adam.

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

References

Kent John (1982). The Fisher-Bingham distribution on the sphere. Journal of the Royal Statistical Society, Series B, 44(1): 71-80.

See Also

```
kent.datacontour, kent.mle, vmf.contour, vmf.kerncontour
```

```
## Not run:
kent.contour(10, 4)
## End(Not run)
```

Contour plots of the von Mises-Fisher distribution ${\it Contour\ plots\ of\ the\ von\ Mises-Fisher\ distribution\ on\ the\ sphere}$

Description

Contour plots of the von Mises-Fisher distribution on the sphere.

Usage

```
vmf.contour(k)
```

Arguments

k

The concentration parameter.

Details

The user specifies the concentration parameter only and not the mean direction or data. This is for illustration purposes only. The graph will always contain circles, as the von Mises-Fisher distribution is the analogue of a bivariate normal in two dimensions with a zero covariance.

Value

A contour plot of the von Mises-Fisher distribution.

Author(s)

Michail Tsagris and Christos Adam.

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

See Also

```
rvmf, vmf.mle, vmf.kerncontour, kent.contour, sphereplot
```

```
## Not run:
vmf.contour(5)
## End(Not run)
```

Conversion of cosines to azimuth and plunge

Conversion of cosines to azimuth and plunge

Description

Conversion of cosines to azimuth and plunge.

Usage

```
cosap(x,y,z)
```

Arguments

x x component of cosine.y y component of cosine.z z component of cosine.

Details

Orientation: x>0 is 'eastward', y>0 is 'southward', and z>0 is 'downward'.

Value

A list including:

A The azimuth
P The plunge

Author(s)

Eli Amson

R implementation and documentation: Eli Amson <eli.amson1988@gmail.com>

References

Amson E, Arnold P, Van Heteren AH, Cannoville A, Nyakatura JA. Trabecular architecture in the forelimb epiphyses of extant xenarthrans (Mammalia). Frontiers in Zoology.

See Also

```
euclid, euclid.inv, eul2rot
```

```
cosap(-0.505, 0.510, -0.696)
```

Converting a rotation matrix on SO(3) to an unsigned unit quaternion Converting a rotation matrix on SO(3) to an unsigned unit quaternion

Description

It returns an unsigned unite quaternion in S^3 (the four-dimensional sphere) from a 3×3 rotation matrix on SO(3).

Usage

rot2quat(X)

Arguments

Χ

A rotation matrix in SO(3).

Details

Firstly construct a system of linear equations by equating the corresponding components of the theoretical rotation matrix proposed by Prentice (1986), and given a rotation matrix. Finally, the system of linear equations are solved by following the tricks mentioned in second reference here in order to achieve numerical accuracy to get quaternion values.

Value

A unsigned unite quaternion.

Author(s)

Anamul Sajib

R implementation and documentation: Anamul Sajib <sajibstat@du.ac.bd>

References

Prentice, M. J. (1986). Orientation statistics without parametric assumptions. Journal of the Royal Statistical Society. Series B: Methodological 48(2). //http://www.euclideanspace.com/maths/geometry/rotations/conversions.

See Also

```
quat2rot, rotation, Arotation \ link{rot.matrix}
```

Examples

```
x <- rnorm(4)
x <- x/sqrt( sum(x^2) ) ## an unit quaternion in R4 ##
R <- quat2rot(x)</pre>
rot2quat(R) ## sign is not exact as you can see
```

Converting an unsigned unit quaternion to rotation matrix on SO(3) Converting an unsigned unit quaternion to rotation matrix on SO(3)

Description

It forms a (3×3) rotation matrix on SO(3) from an unsigned unite quaternion in S^3 (the fourdimensional sphere).

Usage

```
quat2rot(x)
```

Arguments Х

An unsigned unit quaternion in S^3 .

Details

Given an unsigned unit quaternion in S^3 it forms a rotation matrix on SO(3), according to the transformation proposed by Prentice (1986).

Value

A rotation matrix.

Author(s)

Anamul Sajib

R implementation and documentation: Anamul Sajib <sajibstat@du.ac.bd>

References

Prentice, M. J. (1986). Orientation statistics without parametric assumptions. Journal of the Royal Statistical Society. Series B: Methodological 48(2).

See Also

```
rot2quat, rotation, Arotation rot.matrix
```

Examples

Cross validation for estimating the classification rate ${\it Cross \ validation \ for \ estimating \ the \ classification \ rate}$

Description

Cross validation for estimating the classification rate.

Usage

```
\label{eq:continuous} \begin{array}{lll} \mbox{dirda.cv(x, ina, folds = NULL, nfolds = 10, k = 2:10, stratified = FALSE,} \\ & \mbox{type = c("vmf", "iag", "esag", "kent", "knn"),} \\ & \mbox{seed = NULL, B = 1000, parallel = FALSE)} \end{array}
```

Arguments

Х	A matrix with the data in Eulcidean coordinates, i.e. unit vectors. The matrix must have three columns, only spherical data are currently supported.
ina	A variable indicating the groupings.
folds	Do you already have a list with the folds? If not, leave this NULL.
nfolds	How many folds to create?
k	If you choose to use k-NN, what will be the k values?
stratified	Should the folds be created in a stratified way? i.e. keeping the distribution of the groups similar through all folds?
seed	If seed is TRUE, the results will always be the same.
type	The type of classifier to use. The avaliable options are "vmf" (von Mises-Fisher distribution), "esag" (ESAG distribution), "kent" (Kent distribution), "knn" (k-NN algorithm). You can chose any of them or all of them. Note that "esag" and "kent" work only with spherical data.
В	If you used k-NN, should a bootstrap correction of the bias be applied? If yes, 1000 is a good value.
parallel	If you want the standard -NN algorithm to take place in parallel set this equal to TRUE.

Details

Cross-validation for the estimation of the performance of a classifier.

The estimated performance of the best classifier is overestimated. After the cross-valdiation procedure, the predicted values produced by all classifiers are colelcted, from all folds, in an $n \times M$ matrix, where n is the number of samples and M the number of all classifiers used. We sample rows (predictions) with replacement from P and denote them as the in-sample values. The non re-sampled rows are denoted as out-of-sample values. The performance of each classifier in the insample rows is calculated and the classifier with the optimal performance is selected, followed by the calculation of performance in the out-of-sample values. This process is repeated B times and the average performance is returned. The only computational overhead is with the repetitive resampling and calculation of the performance, i.e. no model or classifier is fitted nor trained. For more information see Tsamardinos et al. (2018). This procedure though takes place only for the k-NN algorithm.

The good thing with the function is that you can run any method you want by supplying the folds yourselves using the command makefolds. Then suppose you want to run another method. By suppying the same folds you will be able to have comparative results for all methods.

Value

A list including:

perf A vector with the estimated performance of each classifier.

best The classifier with the optimal performance.
boot.perf The bootstrap bias corrected performance.

Author(s)

Michail Tsagris

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

References

Tsagris M. and Alenazi A. (2019). Comparison of discriminant analysis methods on the sphere. Communications in Statistics: Case Studies, Data Analysis and Applications, 5(4), 467–491.

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

Paine P.J. Preston S.P. & Tsagris M. and Wood A.T.A. (2018). An Elliptically Symmetric Angular Gaussian Distribution. Statistics and Computing, 28(3):689–697.

Morris J. E. & Laycock P. J. (1974). Discriminant analysis of directional data. Biometrika, 61(2): 335-341.

Tsamardinos I., Greasidou E. & Borboudakis G. (2018). Machince Learning, 107(12): 1895-1922.

See Also

```
esag.da, vmfda.pred, dirknn, knn.reg
```

Examples

```
x <- rvmf(300, rnorm(3), 10)
ina <- sample.int(4, 300, replace = TRUE)
dirda.cv(x, ina, B = 1000)</pre>
```

Cross validation in von Mises-Fisher discrminant analysis

Cross validation for estimating the classification rate of a discrminant analysis for directional data assuming a von Mises-Fisher distribution

Description

Cross validation for estimating the classification rate of a discrminant analysis for directional data assuming a von Mises-Fisher distribution.

Usage

```
vmf.da(x, ina, fraction = 0.2, R = 200, seed = NULL)
```

Arguments

x A matrix with the data in Eulcidean coordinates, i.e. unit vectors.

ina A variable indicating the groupings.

fraction The fraction of data to be used as test set.

R The number of repetitions.

seed If seed is TRUE, the results will always be the same.

Details

A repeated cross validation procedure is performed to estimate the rate of correct classification.

Value

A list including:

percent The estimated percent of correct classification and two estimated standard devi-

ations. The one is the standard devation of the rates and the other is assuming a

binomial distribution.

ci Three types of confidence intervals, the standard one, another one based on the

binomial distribution and the third one is the empirical one, which calcualtes the

upper and lower 2.5% of the rates.

Author(s)

Michail Tsagris.

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

Tsagris M. and Alenazi A. (2019). Comparison of discriminant analysis methods on the sphere. Communications in Statistics: Case Studies, Data Analysis and Applications, 5(4), 467–491.

Morris J. E. and Laycock P. J. (1974). Discriminant analysis of directional data. Biometrika, 61(2): 335-341.

See Also

```
vmfda.pred, mixvmf.mle, vmf.mle, dirknn
```

Examples

```
x <- rvmf(100, rnorm(4), 15)
ina <- rep(1:2, each = 50)
vmf.da(x, ina, fraction = 0.2, R = 200)</pre>
```

Cross validation with ESAG discrminant analysis

Cross validation for estimating the classification rate of a discrminant analysis for directional data assuming an ESAG distribution

Description

Cross validation for estimating the classification rate of a discrminant analysis for directional data assuming an ESAG distribution.

Usage

```
esag.da(y, ina, fraction = 0.2, R = 100, seed = NULL)
```

Arguments

y A matrix with the data in Eulcidean coordinates, i.e. unit vectors. The matrix

must have three columns, only spherical data are currently supported.

ina A variable indicating the groupings.

fraction The fraction of data to be used as test set.

R The number of repetitions.

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

Details

A repeated cross validation procedure is performed to estimate the rate of correct classification.

Value

A list including:

percent The estimated percent of correct classification and two estimated standard devi-

ations. The one is the standard devation of the rates and the other is assuming a

binomial distribution.

ci Three types of confidence intervals, the standard one, another one based on the

binomial distribution and the third one is the empirical one, which calcualtes the

upper and lower 2.5% of the rates.

Author(s)

Michail Tsagris

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

References

Tsagris M. and Alenazi A. (2019). Comparison of discriminant analysis methods on the sphere. Communications in Statistics: Case Studies, Data Analysis and Applications, 5(4), 467–491.

Paine P.J., Preston S.P., Tsagris M. and Wood A.T.A. (2018). An Elliptically Symmetric Angular Gaussian Distribution. Statistics and Computing, 28(3):689–697.

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

See Also

```
vmf.da, vmfda.pred, dirknn
```

Examples

```
x <- rvmf(100, rnorm(3), 15)
ina <- rep(1:2, each = 50)
esag.da(x, ina, fraction = 0.2, R = 50)</pre>
```

Density of some (hyper-)spherical distributions

Density of some (hyper-)spherical distributions

Description

Density of some (hyper-)spherical distributions.

Usage

```
dvmf(y, k, mu, logden = FALSE )
iagd(y, mu, logden = FALSE)
dpurka(y, a, theta, logden = FALSE)
```

Arguments

У	A matrix or a vector with the data expressed in Euclidean coordinates, i.e. unit vectors.
k	The concentration parameter of the von Mises-Fisher distribution.
а	The concentration parameter of the Purkayastha distribution.
mu	The mean direction (unit vector) of the von Mises-Fisher distribution or the mean direction of the IAG distribution.
theta	The median direction for the Purkayastha distribution.
logden	If you the logarithm of the density values set this to TRUE.

Details

The density of the von Mises-Fisher, of the IAG or of the Purkayastha distribution is computed.

Value

A vector with the (log) density values of y.

Author(s)

Michail Tsagris

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

Kent John (1982). The Fisher-Bingham distribution on the sphere. Journal of the Royal Statistical Society, Series B, 44(1): 71-80.

Purkayastha S. (1991). A Rotationally Symmetric Directional Distribution: Obtained through Maximum Likelihood Characterization. The Indian Journal of Statistics, Series A, 53(1): 70-83

Cabrera J. and Watson G. S. (1990). On a spherical median related distribution. Communications in Statistics-Theory and Methods, 19(6): 1973-1986.

See Also

```
kent.mle, rkent, esag.mle
```

```
m <- colMeans( as.matrix( iris[,1:3] ) )
y <- rvmf(1000, m = m, k = 10)
dvmf(y, k=10, m )</pre>
```

Density of some circular distributions $Density\ of\ some\ circular\ distributions$

Description

Density of some circular distributions.

Usage

```
dvm(x, m, k, rads = FALSE, logden = FALSE)
dspml(x, mu, rads = FALSE, logden = FALSE)
dwrapcauchy(x, m, rho, rads = FALSE, logden = FALSE)
dcircpurka(x, m, a, rads = FALSE, logden = FALSE)
dggvm(x, param, rads = FALSE, logden = FALSE)
dcircbeta(x, m, a, b, rads = FALSE, logden = FALSE)
dcardio(x, m, rho, rads = FALSE, logden = FALSE)
```

Arguments

Χ	A vector with circular data.
m	The mean value of the von Mises distribution and of the cardioid, a scalar. This is the median for the circular Purkayastha distribution.
mu	The mean vector, a vector with two values for the "spml" and with
k	The concentration parameter.
rho	The rho parameter of the wrapped Cauchy distribution.
a	The $alpha$ parameter of the circular Purkayastha distribution or the $alpha$ parameter of the circular beta distribution.
b	The β parameter of the circular beta distribution.
param	The vector of parameters of the GGVM distribution as returned by the function $\ensuremath{ggvm.mle}$.
rads	If the data are in rads, then this should be TRUE, otherwise FALSE.
logden	If you the logarithm of the density values set this to TRUE.

Details

The density of the von Mises, bivariate projected normal, wrapped Cauchy or the circular Purkayastha distributions is computed.

Value

A vector with the (log) density values of x.

Author(s)

Michail Tsagris

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

See Also

```
dkent, rvonmises, desag
```

Examples

```
x \leftarrow rvonmises(500, m = 2.5, k = 10, rads = TRUE)

mod \leftarrow circ.summary(x, rads = TRUE, plot = FALSE)

den \leftarrow dvm(x, mod\$mesos, mod\$kappa, rads = TRUE, logden = TRUE)

mod\$loglik

sum(den)
```

Density of the spherical Kent and ESAG distributions $Density \ of \ the \ spherical \ Kent \ and \ ESAG \ distributions$

Description

Density of the spherical Kent and ESAG distributions.

Usage

```
dkent(y, G, param, logden = FALSE )
desag(y, mu, gam, logden = FALSE)
```

Arguments

У	A matrix or a vector with the data expressed in Euclidean coordinates, i.e. unit vectors.
G	For the Kent distribution only, a 3 x 3 matrix whose first column is the mean direction. The second and third columns are the major and minor axes respectively.
param	For the Kent distribution a vector with the concentration κ and ovalness β parameters. The ψ has been absorbed inside the matrix G.
mu	The mean vector the ESAG distribution, a vector in \mathbb{R}^3 .
gam	The two gamma parameters of the ESAG distribution.
logden	If you the logarithm of the density values set this to TRUE.

Euclidean transformation 39

Details

The density of the spherical Kent or spherical ESAG distribution is computed.

Value

A vector with the (log) density values of y.

Author(s)

Michail Tsagris

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

Kent John (1982). The Fisher-Bingham distribution on the sphere. Journal of the Royal Statistical Society, Series B, 44(1): 71-80.

Paine P.J., Preston S.P., Tsagris M. and Wood A.T.A. (2018). An Elliptically Symmetric Angular Gaussian Distribution. Statistics and Computing, 28(3):689–697.

See Also

```
kent.mle, rkent, esag.mle
```

Examples

```
m <- colMeans( as.matrix( iris[,1:3] ) )
y <- rkent(1000, k = 10, m = m, b = 4)
mod <- kent.mle(y)
dkent( y, G = mod$G, param = mod$param )</pre>
```

Euclidean transformation

Euclidean transformation

Description

It transforms the data from the spherical coordinates to Euclidean coordinates.

Usage

```
euclid(u)
```

Arguments

U

A two column matrix or even one single vector, where the first column (or element) is the latitude and the second is the longitude. The order is important.

Details

It takes the matrix of unit vectors of latitude and longitude and transforms it to unit vectors.

Value

A three column matrix:

U

The Euclidean coordinates of the latitude and longitude.

Author(s)

Michail Tsagris

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

See Also

```
euclid.inv, Arotation, lambert
```

Examples

```
x <- rvmf(10, rnorm(3), 10)
u <- euclid.inv(x)
euclid(u)
x</pre>
```

Euler angles from a rotation matrix on SO(3)

Compute the Euler angles from a rotation matrix on SO(3).

Description

It calculates three euler angles $(\theta_{12}, \theta_{13}, \theta_{23})$ from a (3×3) rotation matrix X, where X is defined as $X = R_z(\theta_{12}) \times R_y(\theta_{13}) \times R_x(\theta_{23})$. Here $R_x(\theta_{23})$ means a rotation of θ_{23} radians about the x axis.

Usage

```
rot2eul(X)
```

Arguments

Χ

A rotation matrix which is defined as a product of three elementary rotations mentioned above. Here $\theta_{12}, \theta_{23} \in (-\pi, \pi)$ and and $\theta_{13} \in (-\pi/2, \pi/2)$.

Details

Given a rotation matrix X, euler angles are computed by equating each element in X with the corresponding element in the matrix product defined above. This results in nine equations that can be used to find the euler angles.

Value

For a given rotation matrix, there are two eqivalent sets of euler angles.

Author(s)

Anamul Sajib<sajibstat@du.ac.bd>

R implementation and documentation: Anamul Sajib<sajibstat@du.ac.bd>

References

Green, P. J. & Mardia, K. V. (2006). Bayesian alignment using hierarchical models, with applications in proteins bioinformatics. Biometrika, 93(2):235–254.

http://www.staff.city.ac.uk/~sbbh653/publications/euler.pdf

See Also

```
eul2rot
```

```
# three euler angles
theta.12 <- sample( seq(-3, 3, 0.3), 1 )
theta.23 <- sample( seq(-3, 3, 0.3), 1 )
theta.13 <- sample( seq(-1.4, 1.4, 0.3), 1 )
theta.12 ; theta.23 ; theta.13

X <- eul2rot(theta.12, theta.23, theta.13)
X ## A rotation matrix
e <- rot2eul(X)$v1
theta.12 <- e[3]
theta.23 <- e[2]
theta.13 <- e[1]
theta.12 ; theta.23 ; theta.13</pre>
```

Forward Backward Early Dropping selection for circular data using the SPML regression

Forward Backward Early Dropping selection for circular data using
the SPML regression

Description

Forward Backward Early Dropping selection for circular data using the SPML regression.

Usage

Arguments

y The response variable, a numeric vector expressed in rads.

x A matrix with continuous independent variables.

alpha The significance threshold value for assessing p-values. Default value is 0.05.

K How many times should the process be repeated? The default value is 0.

backward After the Forward Early Dropping phase, the algorithm proceeds with a the usual

Backward Selection phase. The default value is set to TRUE. It is advised to perform this step as maybe some variables are false positives, they were wrongly selected. This is rather experimental now and there could be some mistakes in

the indices of the selected variables. Do not use it for now.

parallel If you want the algorithm to run in parallel set this TRUE.

tol The tolerance value to terminate the Newton-Raphson algorithm.

maxiters The maximum number of iterations Newton-Raphson will perform.

Details

The algorithm is a variation of the usual forward selection. At every step, the most significant variable enters the selected variables set. In addition, only the significant variables stay and are further examined. The non significant ones are dropped. This goes until no variable can enter the set. The user has the option to re-do this step 1 or more times (the argument K). In the end, a backward selection is performed to remove falsely selected variables. Note that you may have specified, for example, K=10, but the maximum value FBED used can be 4 for example.

Value

If K is a single number a list including:

Note, that the "gam" argument must be the same though.

res A matrix with the selected variables and their test statistic.

info A matrix with the number of variables and the number of tests performed (or

models fitted) at each round (value of K). This refers to the forward phase only.

runtime The runtime required.

Author(s)

Michail Tsagris

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

References

Borboudakis G. and Tsamardinos I. (2019). Forward-backward selection with early dropping. Journal of Machine Learning Research, 20(8): 1-39.

Tsagis M. (2018). Guide on performing feature selection with the R package MXM. https://f1000research.com/articles/7-1505

Presnell Brett, Morrison Scott P. and Littell Ramon C. (1998). Projected multivariate linear models for directional data. Journal of the American Statistical Association, 93(443): 1068-1077.

See Also

```
spml.reg, spml.regs, spml.mle
```

Examples

```
x <- matrix( runif(100 * 50, 1, 100), ncol = 50 )
y <- runif(100)
a <- spml.fbed(y, x)</pre>
```

Generate random folds for cross-validation

Generate random folds for cross-validation

Description

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

Usage

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

Arguments

ina A variable indicating the groupings.

nfolds The number of folds to produce.

stratified A boolean variable specifying whether stratified random (TRUE) or simple random (FALSE) sampling is to be used when producing the folds.

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

Details

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

Value

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

Author(s)

Michail Tsagris

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

See Also

```
dirda.cv
```

Examples

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

Generation of unit vector(s) with a given angle

Generation of unit vector(s) with a given angle

Description

Generation of unit vector(s) with a given angle from a given unit vector.

Usage

```
vec(x, n = 1, deg = 90)
```

Arguments

A unit vector. If it is not a unit vector it becomes one.

n The number of unit vectors to return.

deg The angle between the given vector and the n vectors to be returned. This must

be in degrees and it has to be between 0 and 180 degrees. If the angle is 0, the same unit vector will be returned. If the angle is 180, the same unit vector with

the signs changed will be returned.

Details

The user provides a unit vector and the degrees. The function will return n unit vectors whose angle with the given unit vector equals the degrees given. For example, if you want 10 unit vectors purpendicual to the x put vec(x, 10, 90).

Value

A list including:

runtime The runtime of the procedure.

crit The calculated angle between the given unit vector and each of the generated

unit vectors.

mat A matrix with the n unit vectors.

Author(s)

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

See Also

```
rvmf, rbingham, rfb
```

Examples

```
x <- rnorm(10)
x <- x / sqrt( sum(x^2) )
a <- vec(x, 20, 90)</pre>
```

Goodness of fit test for grouped data

Goodness of fit test for grouped data

Description

Goodness of fit test for grouped data.

Usage

```
group.gof(g, ni, m, k, dist = "vm", rads = FALSE, R = 999, ncores = 1)
```

Arguments

	A	• .	• . 1	1.	
σ	A vector with the group	nointe	either in	radiane	or in degrees
5	A vector with the group	pomis,	citiici iii	radians	of in acgrees.

ni The frequency of each or group class.

m The mean direction in radians or in degrees.

k The concentration parameter, κ .

dist The distribution to be tested, it can be either "vm" or "uniform".

rads If the data are in radians, this should be TRUE and FALSE otherwise.

R The number of bootstrap simulations to perform, set to 999 by default.

ncores The number of cores to use.

Details

When you have grouped data, you can test whether the data come from the von Mises-Fisher distribution or from a uniform distribution.

Value

A list including:

info A vector with two elements, the test statistic value and the bootstrap p-value.

runtime The runtime of the procedure.

Author(s)

Michail Tsagris

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

References

Arthur Pewsey, Markus Neuhauser, and Graeme D. Ruxton (2013). Circular Statistics in R.

See Also

```
pvm, circ.summary, rvonmises
```

Examples

```
x \leftarrow \text{rvonmises}(100, 2, 10)

g \leftarrow \text{seq}(\text{min}(x) - 0.1, \text{max}(x) + 0.1, \text{length} = 6)

ni \leftarrow \text{as.vector}(\text{table}(\text{cut}(x, g)))

group.gof(g, ni, 2, 10, \text{dist} = "vm", \text{rads} = \text{TRUE}, R = 299, \text{ncores} = 1)

group.gof(g, ni, 2, 5, \text{dist} = "vm", \text{rads} = \text{TRUE}, R = 299, \text{ncores} = 1)
```

Habeck's rotation matrix generation

Generation of three-dimensional random rotations using Habeck's algorithm.

Description

It generates random rotations in three-dimensional space that follow a probability distribution, matrix Fisher distribution, arising in fitting and matching problem.

Usage

```
habeck.rot(F)
```

Haversine distance matrix 47

Arguments

F

An arbitrary 3 x 3 matrix represents the parameter matrix of this distribution.

Details

Firstly rotation matrices \mathbf{X} are chosen which are the closest to \mathbf{F} , and then parameterized using euler angles. Then a Gibbs sampling algorithm is implemented to generate rotation matrices from the resulting disribution of the euler angles.

Value

A simulated rotation matrix.

Author(s)

Anamul Sajib<sajibstat@du.ac.bd>

R implementation and documentation: Anamul Sajib<sajibstat@du.ac.bd>

References

Habeck M (2009). Generation of three-dimensional random rotations in fitting and matching problems. Computational Statistics, 24, 719–731.

Examples

```
F \leftarrow 10^{-1} * matrix( c(85, 11, 41, 78, 39, 60, 43, 64, 48), ncol = 3 ) ## Arbitrary F matrix X <- habeck.rot(F) <math>det(X)
```

Haversine distance matrix

Harvesine distance matrix

Description

Haversine distance matrix.

Usage

```
haversine.dist(x)
```

Arguments

A a matrix of two columns. The first column is the latitude and the second the longitude.

Details

The function computes the haversine distance between all observations.

Value

A matrix with the haversine distances between all observations.

Author(s)

Michail Tsagris.

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

References

https://en.wikipedia.org/wiki/Haversine_formula

See Also

```
cosnn, dirknn
```

Examples

```
x <- rvmf(10, rnorm(3), 10)
x <- euclid.inv(x)
haversine.dist(x)</pre>
```

Hypothesis test for IAG distribution over the ESAG distribution $Hypothesis\ test\ for\ IAG\ distribution\ over\ the\ ESAG\ distribution$

Description

The null hypothesis is whether an IAG distribution fits the data well, where the alternative is that ESAG distribution is more suitable.

Usage

```
iagesag(x, B = 1, tol = 1e-07)
```

Arguments

X	A numeric matrix with three columns containing the data as unit vectors in Euclidean coordinates.
В	The number of bootstrap re-samples. By default is set to 999. If it is equal to 1, no bootstrap is performed and the p-value is obtained throught the asymptotic distribution.
tol	The tolerance to accept that the Newton-Raphson algorithm used in the IAG distribution has converged.

Details

Essentially it is a test of rotational symmetry, whether the two γ parameters are equal to zero. This works for spherical data only.

Value

A vector including:

```
test The value of the test statistic.
p-value or Bootstrap p-value
The p-value of the test.
```

Author(s)

Michail Tsagris

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

References

Paine P.J., Preston S.P., Tsagris M. and Wood A.T.A. (2018). An Elliptically Symmetric Angular Gaussian Distribution. Statistics and Computing, 28(3):689–697.

See Also

```
fishkent, esag.mle, kent.mle, iag.mle
```

Examples

```
x <- rvmf(100, rnorm(3), 15)
iagesag(x)
fishkent(x, B = 1)</pre>
```

Hypothesis test for von Mises-Fisher distribution over Kent distribution

Hypothesis test for von Mises-Fisher distribution over Kent distribution

tion

Description

The null hypothesis is whether a von Mises-Fisher distribution fits the data well, where the altenrative is that Kent distribution is more suitable.

Usage

```
fishkent(x, B = 999)
```

Arguments

x A numeric matrix containing the data as unit vectors in Euclidean coordinates.

B The number of bootstrap re-samples. By default is set to 999. If it is equal to 1, no bootstrap is performed and the p-value is obtained throught the asymptotic

distribution.

Details

Essentially it is a test of rotational symmetry, whether Kent's ovalness parameter (beta) is equal to zero. This works for spherical data only.

Value

A vector including:

```
test The value of the test statistic p-value or Bootstrap p-value

The p-value of the test.
```

Author(s)

Michail Tsagris.

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

References

Rivest, L. P. (1986). Modified Kent's statistics for testing goodness of fit for the Fisher distribution in small concentrated samples. Statistics & probability letters, 4(1): 1-4.

See Also

```
iagesag, vmf.mle, kent.mle, rkent
```

```
x <- rvmf(100, rnorm(3), 15)
fishkent(x)
fishkent(x, B = 1)
iagesag(x)</pre>
```

```
Interactive 3D plot of spherical data Interactive \ 3D \ plot \ of \ spherical \ data
```

Description

Interactive 3D plot of spherical data.

Usage

```
sphereplot(x, col = NULL)
```

Arguments

x A matrix with three columns, unit-vectors, spherical data.

col If you want the points to appear with different colours put numbers here, other-

wise leave it NULL.

Value

An interactive 3D plot of the spherical data will appear.

Author(s)

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

See Also

```
lambert, vmf.contour, euclid
```

```
## Not run:
x <- rvmf(100, rnorm(3), 5)
sphereplot(x)
ina <- rbinom(100, 1, 0.5) + 1
sphereplot(x, col = ina)
## End(Not run)</pre>
```

Inverse of Lambert's equal area projection

Inverse of Lambert's equal area projection

Description

It takes some points from the cartesian coordinates and maps them onto the sphere. The inverse os the Lambert's equal area projection.

Usage

```
lambert.inv(z, mu)
```

Arguments

z A two- column matrix containing the Lambert's equal rea projected data.

mu The mean direction of the data on the sphere.

Details

The data are first mapped on the sphere with mean direction equal to the north pole. Then, they are rotated to have the given mean direction. It is the inverse of the Lambert's equal are projection.

Value

A matrix containing spherical data (unit vectors).

Author(s)

Michail Tsagris

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

References

Kent, John T. (1982). The Fisher-Bingham distribution on the sphere. Journal of the Royal Statistical Society. Series B (Methodological) 44(1):71-80.

See Also

lambert

Examples

```
m <- rnorm(3)
m <- m / sqrt( sum(m^2) )
x <- rvmf(20, m, 19)
mu <- vmf.mle(x)$mu
y <- lambert( euclid.inv(x) )
lambert.inv(y, mu)
euclid.inv(x)</pre>
```

Inverse of the Euclidean transformation

Inverse of the Euclidean transformation

Description

It transforms the data from the Euclidan coordinates to latitude dn longitude.

Usage

```
euclid.inv(U)
```

Arguments

U

A matrix of unit vectors, or even one single unit vector in three dimensions.

Details

It takes the matrix of unit vectors and back transforms it to latitude and longitude.

Value

A two column matrix:

The first column is the latitude and the second is the longitude, both expressed in degrees.

Author(s)

Michail Tsagris

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

See Also

```
euclid, Arotation, lambert
```

Examples

```
x <- rvmf(10, rnorm(3), 10)
euclid.inv(x)
euclid( euclid.inv(x) )
x</pre>
```

k-NN algorithm using the arc cosinus distance

k-NN algorithm using the arc cosinus distance

Description

It classifies new observations to some known groups via the k-NN algorithm.

Usage

```
dirknn(xnew, ina, x, k = 5, mesos = TRUE, parallel = FALSE, rann = FALSE)
```

Arguments

xnew	The new data whose membership is to be predicted, a numeric matrix with unit vectors.
ina	A variable indicating the groups of the data x.
X	The data, a numeric matrix with unit vectors.
k	The number of nearest neighbours, set to 5 by default. It can also be a vector with many values.
mesos	A boolean variable used only in the case of the non standard algorithm (type="NS"). Should the average of the distances be calculated (TRUE) or not (FALSE)? If it is FALSE, the harmonic mean is calculated.
parallel	If you want the standard -NN algorithm to take place in parallel set this equal to TRUE.
rann	If you have large scale datasets and want a faster k-NN search, you can use kd- trees implemented in the R package "RANN". In this case you must set this argument equal to TRUE.

Details

The standard algorithm is to keep the k nearest observations and see the groups of these observations. The new observation is allocated to the most frequent seen group. The non standard algorithm is to calculate the classical mean or the harmonic mean of the k nearest observations for each group. The new observation is allocated to the group with the smallest mean distance.

Value

A vector including:

g A matrix with the predicted group(s). It has as many columns as the values of k.

k-NN regression 55

Author(s)

Michail Tsagris

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

References

Tsagris M. and Alenazi A. (2019). Comparison of discriminant analysis methods on the sphere. Communications in Statistics: Case Studies, Data Analysis and Applications, 5(4), 467–491.

See Also

```
dirknn.tune, vmfda.pred, mixvmf.mle
```

Examples

```
k <- runif(4, 4, 20)
prob <- c(0.2, 0.4, 0.3, 0.1)
mu <- matrix(rnorm(16), ncol = 4)
mu <- mu / sqrt( rowSums(mu^2) )
da <- rmixvmf(200, prob, mu, k)
nu <- sample(1:200, 180)
x <- da$x[nu, ]
ina <- da$id[nu]
xx <- da$x[-nu, ]
id <- da$id[-nu]
a1 <- dirknn(xx, ina, x, k = 5, mesos = TRUE)
a2 <- dirknn(xx, ina, x, k = 5, mesos = FALSE)
b <- vmfda.pred(xx, x, ina)
table(id, a1)
table(id, a2)</pre>
```

k-NN regression

k-NN regression with Euclidean or (hyper-)spherical response and or predictor variables

Description

k-NN regression with Euclidean or (hyper-)spherical response and or predictor variables.

Usage

```
knn.reg(xnew, y, x, k = 5, res = "eucl", estim = "arithmetic")
```

56 k-NN regression

Arguments

xnew	The new data, new predictor variables values. A matrix with either euclidean (univariate or multivariate) or (hyper-)spherical data. If you have a circular response, say u, transform it to a unit vector via $(\cos(u), \sin(u))$. If xnew = x, you will get the fitted values.
У	The currently available data, the response variables values. A matrix with either euclidean (univariate or multivariate) or (hyper-)spherical data. If you have a circular response, say u, transform it to a unit vector via (cos(u), sin(u)).
Х	The currently available data, the predictor variables values. A matrix with either euclidean (univariate or multivariate) or (hyper-)spherical data. If you have a circular response, say u, transform it to a unit vector via (cos(u), sin(u)).
k	The number of nearest neighbours, set to 5 by default. This can also be a vector with many values.
res	The type of the response variable. If it is Euclidean, set this argument equal to "res". If it is a unit vector set it to res="spher".
estim	Once the k observations whith the smallest distance are discovered, what should the prediction be? The arithmetic average of the corresponding y values be used estim="arithmetic" or their harmonic average estim="harmonic".

Details

This function covers a broad range of data, Euclidean and spherical, along with their combinations.

Value

A list with as many elements as the number of values of k. Each element in the list contains a matrix (or a vector in the case of Euclidean data) with the predicted response values.

Author(s)

Michail Tsagris

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

See Also

```
knnreg.tune, spher.reg, spml.reg
```

```
y <- iris[, 1]
x <- as.matrix(iris[, 2:4])
x <- x/ sqrt( rowSums(x^2) ) ## Euclidean response
a <- knn.reg(x, y, x, k = 5, res = "eucl", estim = "arithmetic")

y <- iris[, 2:4]
y <- y / sqrt( rowSums(y^2) ) ## Spherical response
x <- iris[, 1]
a <- knn.reg(x, y, x, k = 5, res = "spher", estim = "arithmetic")</pre>
```

```
Lambert's equal area projection

Lambert's equal area projection
```

Description

It calculates the Lambert's equal area projection.

Usage

```
lambert(y)
```

Arguments

У

A two column matrix with the data. The first column is the altitude and the second is the longitude.

Details

The spherical data are first rotated so that their mean direction is the north pole and then are projected on the plane tagent to the sphere at the north pole.

Value

A two-column matrix with the projected points.

Author(s)

Michail Tsagris

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

References

Kent, John T. (1982). The Fisher-Bingham distribution on the sphere. Journal of the Royal Statistical Society. Series B (Methodological) 44(1):71-80.

See Also

```
euclid, lambert.inv
```

```
x <- rvmf(100, rnorm(3), 20)
x <- euclid.inv(x)
a <- lambert(x)
plot(a)</pre>
```

Logarithm of the Kent distribution normalizing constant

Logarithm of the Kent distribution normalizing constant

Description

Logarithm of the Kent distribution normalizing constant.

Usage

```
kent.logcon(k, b, j = 100)
```

Arguments

- k The conencration parameter, κ . b The ovalness parameter, β . j The number of the terms in the sum to use. By default this is 100.
- **Details**

It calculates logarithm of the normalising constant of the Kent distribution.

Value

The value of the logarithm of the normalising constant of the Kent distribution.

Author(s)

Michail Tsagris

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

References

Kent John (1982). The Fisher-Bingham distribution on the sphere. Journal of the Royal Statistical Society, Series B, 44(1): 71-80.

See Also

```
fb.saddle, kent.mle
```

```
kent.logcon(10, 2)
fb.saddle( c(0, 10, 0), c(0, -2, 2) )
```

Many simple circular or angular regressions

Many simple circular or angular regressions

Description

Many regressions with one circular dependent variable and one Euclidean independent variable.

Usage

```
spml.regs(y, x, tol = 1e-07, logged = FALSE, maxiters = 100, parallel = FALSE)
```

Arguments

у	The dependent variable, it can be a numerical vector with data expressed in radians or it can be a matrix with two columns, the cosinus and the sinus of the circular data. The benefit of the matrix is that if the function is to be called multiple times with the same response, there is no need to transform the vector every time into a matrix.
X	A matrix with independent variable.
tol	The tolerance value to terminatate the Newton-Raphson algorithm.
logged	Do you want the logarithm of the p-value be returned? TRUE or FALSE.
maxiters	The maximum number of iterations to implement.
parallel	Do you want the calculations to take plac ein parallel? The default value if FALSE.

Details

The Newton-Raphson algorithm is fitted in these regression as described in Presnell et al. (1998). For each colum of x a circual regression model is fitted and the hypothesis testing of no association between y and this variable is performed.

Value

A matrix with two columns, the test statistics and their associated (log) p-values.

Author(s)

Michail Tsagris.

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

References

Presnell Brett, Morrison Scott P. and Littell Ramon C. (1998). Projected multivariate linear models for directional data. Journal of the American Statistical Association, 93(443): 1068-1077.

See Also

```
spml.reg, spml.mle, iag.mle
```

Examples

```
x <- rnorm(200)
z <- cbind(3 + 2 * x, 1 -3 * x)
y <- cbind( rnorm(100,z[, 1], 1), rnorm(100, z[, 2], 1) )
y <- y / sqrt( rowSums(y^2) )
x <- matrix( rnorm(100 * 50), ncol = 50 )
a <- Directional::spml.regs(y, x)
x <- NULL</pre>
```

Maps of the world and the continents

maps of the world and the continents

Description

It produces maps of the world and the continents.

Usage

```
asia(title = "Asia", coords = NULL)
africa(title = "Africa", coords = NULL)
europe(title = "Europe", coords = NULL)
north.america(title = "North America", coords = NULL)
oceania(title = "Oceania", coords = NULL)
south.america(title = "South America", coords = NULL)
worldmap(title = "World map", coords = NULL)
```

Arguments

title A character vector with the title of the map.

coords If you want specific points to appear on the plot give the coordinates as a matrix, where the first column contains the longitude and the second column contains

the latitude, in degrees.

Details

Maps of the world or the continents are produced. This are experimental functions and plot the countries with specific colouring at the moment. More functionalities will be added in the future.

Value

A map of the selected continent or the whole world.

Author(s)

Christos Adam.

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

See Also

```
sphereplot
```

Examples

```
## Not run:
oceania()
## End(Not run)
```

Mixtures of Von Mises-Fisher distributions

Mixtures of Von Mises-Fisher distributions

Description

It performs model based clustering for circualr, spherical and hyperspherical data assuming von Mises-Fisher distributions.

Usage

```
mixvmf.mle(x, g, n.start = 10)
```

Arguments

x A matrix with the data expressed as unit vectors.

g The number of groups to fit. It must be greater than or equal to 2.

n.start The number of random starts to try. See also R's built-in function kmeans for

more information about this.

Details

The initial step of the algorithm is not based on a spherical k-means, but on s imple k-means. The results are comparable to the package movMF.

Value

A list including:

param A matrix with the mean direction, the concetrations parameter and mixing prob-

ability of each group.

loglik The value of the maximised log-likelihood.

pred The predicted group of each observation.

iter The number of iteration required by the EM algorithm.

runtime The run time of the algorithm. A numeric vector. The first element is the user

time, the second element is the system time and the third element is the elapsed

time.

Author(s)

Michail Tsagris

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

References

Kurt Hornik and Bettina Grun (2014). movMF: An R Package for Fitting Mixtures of von Mises-Fisher Distributions http://cran.r-project.org/web/packages/movMF/vignettes/movMF.pdf

See Also

```
rmixvmf, bic.mixvmf, mixvmf.contour
```

Examples

```
k <- runif(4, 4, 20)
prob <- c(0.2, 0.4, 0.3, 0.1)
mu <- matrix(rnorm(16), ncol = 4)
mu <- mu / sqrt( rowSums(mu^2) )
x <- rmixvmf(200, prob, mu, k)$x
mixvmf.mle(x, 3)
mixvmf.mle(x, 4)
mixvmf.mle(x, 5)</pre>
```

```
MLE of (hyper-)spherical distributions
```

MLE of (hyper-)spherical distributions

Description

MLE of (hyper-)spherical distributions.

Usage

```
vmf.mle(x, fast = FALSE, tol = 1e-07)
multivmf.mle(x, ina, tol = 1e-07, ell = FALSE)
acg.mle(x, tol = 1e-07)
iag.mle(x, tol = 1e-07)
spcauchy.mle(x, tol = 1e-06)
```

Arguments

Χ		A matrix with directional data, i.e. unit vectors.
fas	t	IF you want a faster version, but with fewer information returned, set this equal to TRUE.
ina	ı	A numerical vector with discrete numbers starting from 1, i.e. 1, 2, 3, 4, or a factor variable. Each number denotes a sample or group. If you supply a continuous valued vector the function will obviously provide wrong results.
ell		This is for the multivmf.mle only. Do you want the log-likelihood returned? The default value is TRUE.
tol		The tolerance value at which to terminate the iterations.

Details

The vmf.mle estimates the mean direction and concentration of a fitted von Mises-Fisher distribution.

The von Mises-Fisher distribution for groups of data is also implemented.

The acg.mle fits the angular central Gaussian distribution. There is a constraint on the estimated covariance matrix; its trace is equal to the number of variables. An iterative algorithm takes place and convergence is guaranteed.

The iag mle implements MLE of the spherical projected normal distribution, for spherical data only.

The speauchy mle estimates the parameters of the spherical Cauchy distribution, for any dimension. The name sounds confusing, but it is implemented for arbitrary dimensions, not only the sphere.

Value

For the von Mises-Fisher a list including:

loglik The maximum log-likelihood value.

mu The mean direction.

kappa The concentration parameter.

For the multi von Mises-Fisher a list including:

loglik A vector with the maximum log-likelihood values if ell is set to TRUE. Other-

wise NULL is returned.

mi A matrix with the group mean directions.

ki A vector with the group concentration parameters.

For the angular central Gaussian a list including:

iter The number if iterations required by the algorithm to converge to the solution.

cova The estimated covariance matrix.

For the spherical projected normal a list including:

iters The number of iteration required by the Newton-Raphson.

mesi A matrix with two rows. The first row is the mean direction and the second is

the mean vector. The first comes from the second by normalising to have unit

length.

param A vector with the elements, the norm of mean vector, the log-likelihood and the

log-likelihood of the spherical uniform distribution. The third value helps in

case you want to do a log-likelihood ratio test for uniformity.

For the spherical Cauchy a list including:

mu The mean direction.

rho The concetration parameter, this takes values in [0, 1).

loglik The log-likelihood value.

Author(s)

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

Sra, S. (2012). A short note on parameter approximation for von Mises-Fisher distributions: and a fast implementation of Is(x). Computational Statistics, 27(1): 177–190.

Tyler D. E. (1987). Statistical analysis for the angular central Gaussian distribution on the sphere. Biometrika 74(3): 579-589.

Paine P.J., Preston S.P., Tsagris M and Wood A.T.A. (2018). An Elliptically Symmetric Angular Gaussian Distribution. Statistics and Computing, 28: 689-697.

Kato S. and McCullagh P. (2018). Mobius transformation and a Cauchy family on the sphere. arXiv preprint arXiv:1510.07679.

See Also

```
racg, vm.mle, rvmf
```

```
m <- c(0, 0, 0, 0)
s <- cov(iris[, 1:4])
x <- racg(100, s)
mod <- acg.mle(x)
mod
cov2cor(mod$cova) ## estimated covariance matrix turned into a correlation matrix</pre>
```

```
cov2cor(s) ## true covariance matrix turned into a correlation matrix
vmf.mle(x)
x <- rbind( rvmf(100,rnorm(4), 10), rvmf(100,rnorm(4), 20) )
a <- multivmf.mle(x, rep(1:2, each = 100) )</pre>
```

MLE of some circular distributions

MLE of some circular distributions

Description

MLE of some circular distributions.

Usage

```
spml.mle(x, rads = FALSE, tol = 1e-07)
wrapcauchy.mle(x, rads = FALSE, tol = 1e-07)
circexp.mle(x, rads = FALSE, tol = 1e-06)
circbeta.mle(x, rads = FALSE)
cardio.mle(x, rads = FALSE)
ggvm.mle(phi, rads = FALSE)
```

Arguments

X	A numerical vector with the circular data. They must be expressed in radians.
phi	A numerical vector with the circular data.
rads	If the data are in radians set this to TRUE.
tol	The tolerance level to stop the iterative process of finding the MLEs.

Details

The parameters of the bivariate angular Gaussian (spml.mle), wrapped Cauchy, circular exponential, cardioid, circular beta and geometrically generalised von Mises distributions are estimated. For the Wrapped Cauchy, the iterative procedure described by Kent and Tyler (1988) is used. The Newton-Raphson algorithm for the angular Gaussian is described in the regression setting in Presnell et al. (1998). The circular exponential is also known as wrapped exponential distribution.

Value

A list including:

iters The iterations required until convergence.loglik The value of the maximised log-likelihood.

param A vector consisting of the estimates of the two parameters, the mean direction

for both distributions and the concentration parameter kappa and the rho for the von Mises and wrapped Cauchy respectively. For the circular beta this contains the mean angle and the α and β parameters. For the cardioid distribution this contains the μ and rho parameters. For the generalised von Mises this is a vector consisting of the ζ , κ , μ and α parameters of the generalised von Mises distribution as described in Equation (2.7) of Dietrich and Richter (2017).

gamma The norm of the mean vector of the angular Gaussian distribution.

mu The mean vector of the angular Gaussian distribution.

mumu In the case of "angular Gaussian distribution this is the mean angle in radians.

lambda The lambda parameter of the circular exponential distribution.

Author(s)

Michail Tsagris.

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

References

Mardia K. V. and Jupp P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

Sra S. (2012). A short note on parameter approximation for von Mises-Fisher distributions: and a fast implementation of Is(x). Computational Statistics, 27(1): 177-190.

Presnell Brett, Morrison Scott P. and Littell Ramon C. (1998). Projected multivariate linear models for directional data. Journal of the American Statistical Association, 93(443): 1068-1077.

Kent J. and Tyler D. (1988). Maximum likelihood estimation for the wrapped Cauchy distribution. Journal of Applied Statistics, 15(2): 247–254.

Dietrich T. and Richter W. D. (2017). Classes of geometrically generalized von Mises distributions. Sankhya B, 79(1): 21-59.

https://en.wikipedia.org/wiki/Wrapped_exponential_distribution

See Also

```
circ.summary, purka.mle, rvonmises, vmf.mle, rvmf
```

```
x <- rvonmises(1000, 3, 9)
spml.mle(x, rads = TRUE)
wrapcauchy.mle(x, rads = TRUE)
circexp.mle(x, rads = TRUE)
ggvm.mle(x, rads = TRUE)</pre>
```

MLE of some circular distributions with multiple samples ${\it MLE~of~some~circular~distributions~with~multiple~samples}$

Description

MLE of some circular distributions with multiple samples.

Usage

```
multivm.mle(x, ina, tol = 1e-07, ell = FALSE)
multispml.mle(x, ina, tol = 1e-07, ell = FALSE)
```

Arguments

X	A numerical vector with the circular data. They must be expressed in radians. For the "spml.mle" this can also be a matrix with two columns, the cosinus and the sinus of the circular data.
ina	A numerical vector with discrete numbers starting from 1, i.e. 1, 2, 3, 4, or a factor variable. Each number denotes a sample or group. If you supply a continuous valued vector the function will obviously provide wrong results.
tol	The tolerance level to stop the iterative process of finding the MLEs.
ell	Do you want the log-likelihood returned? The default value is FALSE.

Details

The parameters of the von Mises and of the bivariate angular Gaussian distributions are estimated for multiple samples.

Value

A list including:

iters	The iterations required until convergence. This is returned in the wrapped Cauchy distribution only.
loglik	A vector with the value of the maximised log-likelihood for each sample.
mi	For the von Mises, this is a vector with the means of each sample. For the angular Gaussian (spml), a matrix with the mean vector of each sample
ki	A vector with the concentration parameter of the von Mises distribution at each sample.
gi	A vector with the norm of the mean vector of the angular Gaussian distribution at each sample.

Author(s)

Michail Tsagris

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

References

Mardia K. V. and Jupp P. E. (2000). Directional statistics. Chicester: John Wiley \& Sons.

Sra S. (2012). A short note on parameter approximation for von Mises-Fisher distributions: and a fast implementation of Is(x). Computational Statistics, 27(1): 177-190.

Presnell Brett, Morrison Scott P. and Littell Ramon C. (1998). Projected multivariate linear models for directional data. Journal of the American Statistical Association, 93(443): 1068-1077.

Kent J. and Tyler D. (1988). Maximum likelihood estimation for the wrapped Cauchy distribution. Journal of Applied Statistics, 15(2): 247–254.

See Also

```
colspml.mle, purka.mle
```

Examples

```
y <- rcauchy(100, 3, 1)
x <- y
ina <- rep(1:2, 50)
multivm.mle(x, ina)
multispml.mle(x, ina)</pre>
```

MLE of the ESAG distribution

MLE of the ESAG distribution

Description

MLE of the ESAG distribution.

Usage

```
esag.mle(y, full = FALSE, tol = 1e-06)
```

Arguments

y A matrix with the data expressed in Euclidean coordinates, i.e. unit vectors.

full If you want some extra information, the inverse of the covariance matrix, the

rho parameter (smallest eigenvalue of the covariance matrix) and the angle of

rotation ψ set this equal to TRUE. Otherwise leave it FALSE.

tol A tolerance value to stop performing successive optimizations.

Details

MLE of the MLE of the ESAG distributiontribution, on the sphere, is implemented. ESAG stands for Elliptically Symmetric Angular Gaussian and it was suugested by Paine et al. (2017). Unlike the projected normal distribution this is rotationally symmetric and is a competitor of the spherical Kent distribution (which is also non rotational symmetric).

Value

A list including:

mu The mean vector in \mathbb{R}^3 .

gam The two gamma parameters.

loglik The log-likelihood value.

vinv The inverse of the covariance matrix. It is returned if the argument "full" is TRUE.

rho The rho parameter (smallest eigenvalue of the covariance matrix). It is returned if the argument "full" is TRUE.

psi The angle of rotation ψ set this equal to TRUE. It is returned if the argument "full" is TRUE.

The log-likelihood value of the isotropic angular Gaussian distribution. That is,

the projected normal distribution which is rotational symmetric.

Author(s)

Michail Tsagris

iag.loglik

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

References

Paine P.J., Preston S.P., Tsagris M. and Wood A.T.A. (2018). An Elliptically Symmetric Angular Gaussian Distribution. Statistics and Computing, 28(3):689–697.

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

See Also

```
desag, resag, iag.mle, kent.mle, acg.mle, circ.summary, sphereplot
```

Examples

```
m <- colMeans( as.matrix( iris[,1:3] ) )
y <- resag(1000, m, c(1,0.5) )
esag.mle(y)</pre>
```

MLE of the Kent distribution

MLe of the Kent distribution

Description

It estimates the concentration and the ovalness parameter of some directional data assuming the Kent distribution. The mean direction and major and minor axes are also estimated.

Usage

```
kent.mle(x)
```

Arguments

Х

A matrix containing spherical data in Euclidean coordinates.

Details

The Kent distribution is fitted to some data and its parameters are estimated.

Value

A list including:

runtime	The run time of the procedure.
G	A 3 x 3 matrix whose first column is the mean direction. The second and third columns are the major and minor axes respectively.
param	A vector with the concentration κ and ovalness β parameters and the angle ψ used to rotate H and hence estimate G as in Kent (1982).
logcon	The logarithm of the normalising constant, using the third type approximation (Kume and Wood, 2005).
loglik	The value of the log-likelihood.

Author(s)

Michail Tsagris

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

References

Kent John (1982). The Fisher-Bingham distribution on the sphere. Journal of the Royal Statistical Society, Series B, 44(1): 71-80.

Kume Alfred and Wood Andrew T.A. (2005). Saddlepoint approximations for the Bingham and Fisher-Bingham normalizing constants. Biometrika, 92(2):465-476

See Also

```
kent.mle, fb.saddle, vmf.mle, wood.mle, sphereplot
```

```
x <- rvmf(200, rnorm(3), 15)
kent.mle(x)
vmf.mle(x)
A <- diag( c(-5, 0, 5) )
x <- rfb(200, 15, rnorm(3), A)</pre>
```

```
kent.mle(x)
vmf.mle(x)
```

```
MLE of the Matrix Fisher distribution on SO(3)
MLE \ of \ the \ Matrix \ Fisher \ distribution \ on \ SO(3)
```

Description

It returns the maximum likelihood estimate of the Matrix Fisher parameter F(3x3).

Usage

```
matrixfisher.mle(X)
```

Arguments

Χ

An array containing rotation matrices in SO(3).

Value

The components of $svd(\bar{X})$.

Author(s)

Anamul Sajib & Chris Fallaize.

R implementation and documentation: Anamul Sajib <sajibstat@du.ac.bd> & Chris Fallaize.

References

Prentice M. J. (1986). Orientation statistics without parametric assumptions. Journal of the Royal Statistical Society. Series B: Methodological 48(2).

See Also

```
rmatrixfisher
```

MLE of the Purkayashta distribution MLE of the Purkayashta distribution

Description

MLE of the Purkayashta distribution.

Usage

```
purka.mle(x, tol = 1e-07)
```

Arguments

x A numerical vector with data expressed in radians or a matrix with spherical

data.

tol The tolerance value to terminate the Brent algorithm.

Details

MLE of the Purkayastha distribution is performed.

Value

A list including:

theta The median direction.

alpha The concentration parameter.

loglik The log-likelihood.

alpha.sd The standard error of the concentration parameter.

Author(s)

Michail Tsagris

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

References

Purkayastha S. (1991). A Rotationally Symmetric Directional Distribution: Obtained through Maximum Likelihood Characterization. The Indian Journal of Statistics, Series A, 53(1): 70-83

Cabrera J. and Watson G. S. (1990). On a spherical median related distribution. Communications in Statistics-Theory and Methods, 19(6): 1973-1986.

See Also

circ.cor1

```
x <- cbind( rnorm(100,1,1), rnorm(100, 2, 1) )
x <- x / sqrt(rowSums(x^2))
purka.mle(x)</pre>
```

MLE of the Wood bimodal distribution on the sphere ${\it MLe~of~the~Wood~bimodal~distribution~on~the~sphere}$

Description

It estimates the parameters of the Wood bimodal distribution.

Usage

```
wood.mle(y)
```

Arguments

y

A matrix containing two columns. The first one is the latitude and the second is the longitude, both expressed in degrees.

Details

The Wood distribution is fitted to some data and its parameters are estimated. It is a bimodal distribution which contains 5 parameters, just like the Kent distribution.

Value

A list including:

info A 5 x 3 matrix containing the 5 parameters, gamma, delta, alpha, beta and kappa

along with their corresponding 95% confidence intervals all expressed in de-

grees.

modes The two axis of the modes of the distribution expressed in degrees.

unitvectors A 3 x 3 matrix with the 3 unitvectors associated with the gamma and delta

parameters.

loglik The value of the log-likelihood.

Author(s)

Michail Tsagris.

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

References

Wood A.T.A. (1982). A bimodal distribution on the sphere. Journal of the Royal Statistical Society, Series C, 31(1): 52-58.

See Also

```
kent.mle, esag.mle, vmf.mle, sphereplot
```

Examples

```
x <- rvmf(100, rnorm(3), 15)
x <- euclid.inv(x)
wood.mle(x)</pre>
```

Naive Bayes classifiers for circular data

Naive Bayes classifiers for directional data

Description

Naive Bayes classifiers for directional data.

Usage

```
vm.nb(xnew = NULL, x, ina, tol = 1e-07)
spml.nb(xnew = NULL, x, ina, tol = 1e-07)
```

Arguments

xnew	A numerical matrix with new predictor variables whose group is to be predicted. Each column refers to an angular variable.
X	A numerical matrix with observed predictor variables. Each column refers to an angular variable.
ina	A numerical vector with strictly positive numbers, i.e. 1,2,3 indicating the groups of the dataset. Alternatively this can be a factor variable.
tol	The tolerance value to terminate the Newton-Raphson algorithm.

Details

Each column is supposed to contain angular measurements. Thus, for each column a von Mises distribution or an circular angular Gaussian distribution is fitted. The product of the densities is the joint multivariate distribution.

Value

A list including:

mu A matrix with the mean vectors expressed in radians.

mu1 A matrix with the first set of mean vectors.
mu2 A matrix with the second set of mean vectors.

kappa A matrix with the kappa parameters for the vonMises distribution or with the

norm of the mean vectors for the circular angular Gaussian distribution.

ni The sample size of each group in the dataset.

est The estimated group of the xnew observations. It returns a numerical value back

regardless of the target variable being numerical as well or factor. Hence, it is suggested that you do \"as.numeric(ina)\" in order to see what is the predicted

class of the new data.

Author(s)

Michail Tsagris

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

See Also

```
vmnb.pred, weibull.nb
```

Examples

```
x <- matrix( runif( 100, 0, 1 ), ncol = 2 )
ina <- rbinom(50, 1, 0.5) + 1
a <- vm.nb(x, x, ina)</pre>
```

Normalised spatial median for directional data

Normalised spatial median for directional data

Description

Normalised spatial median for directional data.

Usage

```
nsmedian(x, tol = 1e-07)
```

Arguments

x A matrix with Euclidean data, continuous variables.

tol A tolerance level to terminate the process.

Details

The spatial median, using a fixed point iterative algorithm, for Euclidean data is calculated. It is a robust location estimate. Then it is normalised to become a unit vector. Generally speaking this might be a better alternative than then mediandir.

Value

A vector with the spatial median.

Author(s)

Michail Tsagris.

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

References

Ducharme G. R. and Milasevic P. (1987). Spatial median and directional data. Biometrika, 74(1), 212-215.

Jyrki Mottonen, Klaus Nordhausen and Hannu Oja (2010). Asymptotic theory of the spatial median. In Nonparametrics and Robustness in Modern Statistical Inference and Time Series Analysis: A Festschrift in honor of Professor Jana Jureckova.

T. Karkkaminen and S. Ayramo (2005). On computation of spatial median for robust data mining. Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems, EUROGEN 2005, R. Schilling, W.Haase, J. Periaux, H. Baier, G. Bugeda (Eds) FLM, Munich. http://users.jyu.fi/~samiayr/pdf/ayramo_eurogen05.pdf

See Also

mediandir

Examples

```
m <- rnorm(3)
m <- m / sqrt( sum(m^2) )
x <- rvmf(100, m, 10)
nsmedian(x)
mediandir(x)</pre>
```

Permutation based 2-sample mean test for (hyper-)spherical data $Permutation \ based \ 2\text{-}sample \ mean \ test for \ (hyper-)spherical \ data$

Description

Permutation based 2-sample mean test for (hyper-)spherical data.

Usage

```
hcf.perm(x1, x2, B = 999)
lr.perm(x1, x2, B = 999)
hclr.perm(x1, x2, B = 999)
embed.perm(x1, x2, B = 999)
het.perm(x1, x2, B = 999)
```

Arguments

x1 A matrix with the data in Euclidean coordinates, i.e. unit vectors.
 x2 A matrix with the data in Euclidean coordinates, i.e. unit vectors.
 B The number of permutations to perform.

Details

The high concentration (hcf.perm), log-likelihood ratio (lr.perm), high concentration log-likelihood ratio (hclr.perm), embedding approach (embed.perm) or the non equal concentration parameters approach (het.perm) is used.

Value

A vector including:

test The test statistic value.
p-value The p-value of the F test.

kappa The common concentration parameter kappa based on all the data.

Author(s)

Michail Tsagris

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

Rumcheva P. and Presnell B. (2017). An improved test of equality of mean directions for the Langevin-von Mises-Fisher distribution. Australian & New Zealand Journal of Statistics, 59(1), 119-135.

Tsagris M. and Alenazi A. (2022). An investigation of hypothesis testing procedures for circular and spherical mean vectors. Communications in Statistics-Simulation and Computation (Accepted for publication).

```
hcf.boot, hcf.aov, spherconc.test, conc.test
```

```
x <- rvmf(60, rnorm(3), 15)
ina <- rep(1:2, each = 30)
x1 <- x[ina == 1, ]
x2 <- x[ina == 2, ]
hcf.perm(x1, x2)
lr.perm(x1, x2)
het.boot(x1, x2)</pre>
```

Permutation based 2-sample mean test for circular data

Permutation based 2-sample mean test for circular data

Description

Permutation based 2-sample mean test for circular data.

Usage

```
hcfcirc.perm(u1, u2, rads = TRUE, B = 999)
hetcirc.perm(u1, u2, rads = TRUE, B = 999)
lrcirc.perm(u1, u2, rads = TRUE, B = 999)
hclrcirc.perm(u1, u2, rads = TRUE, B = 999)
embedcirc.perm(u1, u2, rads = TRUE, B = 999)
```

Arguments

u1 A numeric vector containing the data of the first sample.
 u2 A numeric vector containing the data of the first sample.
 rads If the data are in radians, this should be TRUE and FALSE otherwise.
 B The number of permutations to perform.

Details

The high concentration (hcfcirc.perm), log-likelihood ratio (lrcirc.perm), high concentration log-likelihood ratio (hclrcirc.perm), embedding approach (embedcirc.perm) or the non equal concentration parameters approach (hetcirc.perm) is used.

Value

A vector including:

The value of the test statistic.

p-value The p-value of the test.

kappa The concentration parameter based on all the data. If the het.circaov is used this

argument is not returned.

Author(s)

Michail Tsagris.

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

Rumcheva P. and Presnell B. (2017). An improved test of equality of mean directions for the Langevin-von Mises-Fisher distribution. Australian & New Zealand Journal of Statistics, 59(1), 119-135.

Tsagris M. and Alenazi A. (2022). An investigation of hypothesis testing procedures for circular and spherical mean vectors. Communications in Statistics-Simulation and Computation (Accepted for publication).

See Also

```
hcf.circaov, het.aov
```

Examples

```
u1 <- rvonmises(20, 2.4, 5)
u2 <- rvonmises(20, 2.4, 10)
hcfcirc.perm(u1, u2)
lrcirc.perm(u1, u2)
```

 $\label{lem:prediction} \mbox{Prediction in discriminant analysis based on ESAG distribution}$

Prediction of a new observation using discriminant analysis based on ESAGdistribution

Description

Prediction of a new observation using discriminant analysis based on ESAG distribution.

Usage

```
esagda.pred(ynew, y, ina)
```

Arguments

ynew The new observation(s) (unit vector(s)) whose group is to be predicted.

y A data matrix with unit vectors, i.e. spherical directional data.

ina A vector indicating the groups of the data y.

Details

Prediction of the class of a new spherical vector assuming ESAG distribution.

Value

A vector with the predicted group of each new observation.

Author(s)

Michail Tsagris

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

References

Tsagris M. and Alenazi A. (2019). Comparison of discriminant analysis methods on the sphere. Communications in Statistics: Case Studies, Data Analysis and Applications, 5(4), 467–491.

Paine P.J., Preston S.P., Tsagris M. and Wood A.T.A. (2017). An Elliptically Symmetric Angular Gaussian Distribution. Statistics and Computing, 28(3):689–697.

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

See Also

```
esag.da, vmfda.pred, dirknn, knn.reg
```

Examples

```
m1 <- rnorm(3)
m2 <- rnorm(3) + 0.5
y <- rbind( rvmf(100, m1, 3), rvmf(80, m2, 5) )
ina <- c( rep(1,100), rep(2, 80) )
ynew <- rbind(rvmf(10, m1, 10), rvmf(10, m2, 5))
id <- rep(1:2, each = 10)
g <- esagda.pred(ynew, y, ina)
table(id, g)</pre>
```

Prediction in discriminant analysis based on von Mises-Fisher distribution $Prediction \ of \ a \ new \ observation \ using \ discriminant \ analysis \ based \ on \ von \ Mises-Fisher \ distribution$

Description

Prediction of the class of a new observation using discriminant analysis based on von Mises-Fisher distribution.

Usage

```
vmfda.pred(xnew, x, ina)
```

Arguments

xnew	The new observation(s) (unit vector(s)) whose group is to be predicted.
x	A data matrix with unit vectors, i.e. directional data.
ina	A vector indicating the groups of the data x.

Details

Discriminant analysis assuming von Mises-Fisher distributions.

Value

A vector with the predicted group of each new observation.

Author(s)

Michail Tsagris.

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

References

Tsagris M. and Alenazi A. (2019). Comparison of discriminant analysis methods on the sphere. Communications in Statistics: Case Studies, Data Analysis and Applications, 5(4), 467–491.

Morris J. E. and Laycock P. J. (1974). Discriminant analysis of directional data. Biometrika, 61(2): 335-341.

See Also

```
vmf.da, mixvmf.mle, dirknn, knn.reg
```

Examples

```
m1 <- rnorm(5)
m2 <- rnorm(5)
x <- rbind( rvmf(100, m1, 5), rvmf(80, m2, 10) )
ina <- c( rep(1,100), rep(2, 80) )
y <- rbind(rvmf(10, m1, 10), rvmf(10, m2, 5))
id <- rep(1:2, each = 10)
g <- vmfda.pred(y, x, ina)
table(id, g)</pre>
```

Prediction with some naive Bayes classifiers for circular data $Prediction \ with \ some \ naive \ Bayes \ classifiers \ for \ circular \ data$

Description

Prediction with some naive Bayes classifiers for circular data.

Usage

```
vmnb.pred(xnew, mu, kappa, ni)
spmlnb.pred(xnew, mu1, mu2, ni)
```

Arguments

xnew	A numerical matrix with new predictor variables whose group is to be predicted. Each column refers to an angular variable.
mu	A matrix with the mean vectors expressed in radians.
mu1	A matrix with the first set of mean vectors.
mu2	A matrix with the second set of mean vectors.
kappa	A matrix with the kappa parameters for the vonMises distribution or with the norm of the mean vectors for the circular angular Gaussian distribution.
ni	The sample size of each group in the dataset.

Details

Each column is supposed to contain angular measurements. Thus, for each column a von Mises distribution or an circular angular Gaussian distribution is fitted. The product of the densities is the joint multivariate distribution.

Value

A numerical vector with 1, 2, ... denoting the predicted group.

Author(s)

Michail Tsagris

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

```
vm.nb, weibullnb.pred
```

```
x <- matrix( runif( 100, 0, 1 ), ncol = 2 )
ina <- rbinom(50, 1, 0.5) + 1
a <- vm.nb(x, x, ina)
a2 <- vmnb.pred(x, a$mu, a$kappa, a$ni)</pre>
```

Probability density function of the von Mises-Fisher distribution

Probability density function of the von Mises-Fisher distribution

Description

Probability density function of the von Mises-Fisher distribution.

Usage

```
pvm(theta, m, k, rads = FALSE)
```

Arguments

theta A numerical value, either in radians or in degrees.

m The mean direction in radians or in degrees.

k The concentration parameter, κ .

rads If the data are in radians, this should be TRUE and FALSE otherwise.

Details

This value calculates the probability of x being less than theta and is used by group.gof.

Value

The probability that of x being less than theta, where x follows the von Mises-Fisher distribution.

Author(s)

Michail Tsagris

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

References

Arthur Pewsey, Markus Neuhauser, and Graeme D. Ruxton (2013). Circular Statistics in R.

```
group.gof, circ.summary, rvonmises
```

```
pvm(1, 2, 10, rads = TRUE)
pvm(2, 2, 10, rads = TRUE)
```

Random sample of matrices in SO(p)Random sample of matrices in SO(p)

Description

Random sample of matrices in SO(p).

Usage

```
rsop(n, p)
```

Arguments

- n The sample size, the number of matrices you want to generate.
- p The dimensionality of the matrices.

Details

The idea is very simple. Start with a unit vector pointing at the north pole (1,0,...,0). Then generate random numbers from a standard normal and scale them so that they have a unit length. To put it differently, a sample of n values from the uniform distribution on the sphere is generated. Then calculate the rotation matrix required to go from the north pole to each of a generated vector.

Value

If n = 1 one matrix is returned. If n is greater than 1, an array with n matrices inside.

Author(s)

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

References

G. J. A. Amaral, I. L. Dryden & Andrew T. A. Wood (2007). Pivotal Bootstrap Methods for k-Sample Problems in Directional Statistics and Shape Analysis. Journal of the American Statistical Association, 102(478): 695-707.

```
rotation, Arotation, rot.matrix
```

```
x1 <- rsop(1, 3)
x2 <- rsop(10, 3)
x3 <- rsop(100, 10)
```

```
Rayleigh's test of uniformity

Rayleigh's test of uniformity
```

Description

It checkes whether the data are uniformly distributed on the sphere or hypersphere.

Usage

```
rayleigh(x, modif = TRUE, B = 999)
```

Arguments

x A matrix containing the data, unit vectors.

modif If modif is TRUE, the modification as suggested by Jupp (2001) is used.

B If B is greater than 1, bootstap calibation os performed. If it is equal to 1,

classical theory is used.

Details

The Rayleigh test of uniformity is not the best, when there are two antipodal mean directions. In this case it will fail. It is good to test whether there is one mean direction or not. To put it differently, it tests whether the concentration parameter of the Fisher distribution is zero or not.

Value

A vector including:

```
test The value of the test statistic.
p-value or Bootstrap p-value
The (bootstrap) p-value of the test.
```

Author(s)

Michail Tsagris

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

Jupp, P. E. (2001). Modifications of the rayleigh and bingham tests for uniformity of directions. Journal of Multivariate Analysis, 77(2):1-20.

Rayleigh, L. (1919). On the problem of random vibrations, and of random flights in one, two, or three dimensions. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 37(220):321-347.

See Also

```
vmf.mle, meandir.test, acg.mle
```

Examples

```
x <- rvmf(100, rnorm(5), 1) ## Fisher distribution with low concentration rayleigh(x)
```

```
Read a file as a Filebacked Big Matrix

Read a file as a Filebacked Big Matrix
```

Description

Read a file as a Filebacked Big Matrix.

Usage

```
read.fbm(file, select)
```

Arguments

file The File to read.

select Indices of columns to read (sorted). The length of select will be the number of

columns of the resulting FBM.

Details

The functions read a file as a Filebacked Big Matrix object. For more information see the "bigstatsr" package.

Value

A Filebacked Big Matrix object.

Author(s)

Michail Tsagris.

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

See Also

```
vmf.mle, kent.mle
```

Examples

```
## Not run:
dataset <- matrix( runif(100 * 50, 1, 100), ncol = 50 )
write.csv(dataset, "dataset.csv")
a <- read.fbm("dataset.csv", select = 1:50)
## End(Not run)</pre>
```

Rotation axis and angle of rotation given a rotation matrix

Rotation axis and angle of rotation given a rotation matrix

Description

Given a 3 x 3 rotation matrix, the angle and the axis of rotation are calculated.

Usage

```
Arotation(A)
```

Arguments

Α

A 3 x 3 rotation matrix.

Details

If the user does not supply a rotation matrix a message will appear.

Value

A list including:

angle The angle of rotation expressed in degrees.

axis The axis of rotation. A vector of two components, latitude and longitude, ex-

pressed in degrees.

Author(s)

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

References

Course webpage of Howard E. Haber. http://scipp.ucsc.edu/~haber/ph216/rotation_12.pdf Ted Chang (1986). Spherical Regression. Annals of Statistics, 14(3): 907-924.

See Also

```
rot.matrix, rotation, rsop
```

Examples

```
ksi <- c(25.31, 24.29)
theta <- 2.38
A <- rot.matrix(ksi, theta, rads = FALSE)
A
Arotation(A)</pre>
```

Rotation matrix from a rotation axis and angle of rotation

Rotation matrix from a rotation axis and angle of rotation

Description

It calculates a rotation matrix from a rotation axis and angle of rotation.

Usage

```
rot.matrix(ksi, theta, rads = FALSE)
```

Arguments

ksi The rotation axis, a vector with two elements, the first is the latitude and the

second is the longitude.

theta The angle of rotation.

rads If both the ksi and theta are in rads, this should be TRUE. If both the ksi and

theta are in degrees, this should be FALSE.

Details

The function accepts as arguments the rotation axis and the angle of rotation and it calcualtes the requested rotation matrix.

Value

A 3 x 3 rotation matrix.

Author(s)

Michail Tsagris

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

References

Course webpage of Howard E. Haber. http://scipp.ucsc.edu/~haber/ph216/rotation_12.pdf Ted Chang (1986). Spherical Regression. Annals of Statistics, 14(3): 907-924.

See Also

```
Arotation, rotation, rsop
```

Examples

```
ksi <- c(25.31, 24.29)
theta <- 2.38
A <- rot.matrix(ksi, theta, rads = FALSE)
A
Arotation(A)</pre>
```

Rotation matrix on SO(3) from three Euler angles $Construct\ a\ rotation\ matrix\ on\ SO(3)\ from\ the\ Euler\ angles.$

Description

It forms a rotation matrix X on SO(3) by using three Euler angles $(\theta_{12}, \theta_{13}, \theta_{23})$, where X is defined as $X = R_z(\theta_{12}) \times R_y(\theta_{13}) \times R_x(\theta_{23})$. Here $R_x(\theta_{23})$ means a rotation of θ_{23} radians about the x axis.

Usage

```
eul2rot(theta.12, theta.23, theta.13)
```

Arguments

```
theta.12 An Euler angle, a number which must lie in (-\pi,\pi).
theta.23 An Euler angle, a number which must lie in (-\pi,\pi).
theta.13 An Euler angle, a number which must lie in (-\pi/2,\pi/2).
```

Details

Given three euler angles a rotation matrix X on SO(3) is formed using the transformation according to Green and Mardia (2006) which is defined above.

Value

A roation matrix.

Author(s)

Anamul Sajib<sajibstat@du.ac.bd>

R implementation and documentation: Anamul Sajib <sajibstat@du.ac.bd>

References

Green, P. J. & Mardia, K. V. (2006). Bayesian alignment using hierarchical models, with applications in proteins bioinformatics. Biometrika, 93(2):235–254.

See Also

rot2eul

Examples

```
# three euler angles
theta.12 <- sample( seq(-3, 3, 0.3), 1 )
theta.23 <- sample( seq(-3, 3, 0.3), 1 )
theta.13 <- sample( seq(-1.4, 1.4, 0.3), 1 )
theta.12 ; theta.23 ; theta.13

X <- eul2rot(theta.12, theta.23, theta.13)
X # A rotation matrix
det(X)
e <- rot2eul(X)$v1
theta.12 <- e[3]
theta.23 <- e[2]
theta.13 <- e[1]
theta.12 ; theta.23 ; theta.13</pre>
```

Rotation matrix to rotate a spherical vector along the direction of another

Rotation matrix to rotate a spherical vector along the direction of another

Description

A rotation matrix is calculated to rotate a unit vector along the direction of another.

Usage

```
rotation(a, b)
```

Arguments

- a The initial unit vector.
- b The target unit vector.

Details

The function calcualtes a rotation matrix given two vectors. This rotation matrix is the connection between the two spherical only, vectors.

Value

A rotation matrix whose dimension is equal to the length of the unit vectors.

Author(s)

Michail Tsagris

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

References

G. J. A. Amaral, I. L. Dryden & Andrew T. A. Wood (2007). Pivotal Bootstrap Methods for k-Sample Problems in Directional Statistics and Shape Analysis. Journal of the American Statistical Association, 102(478): 695-707.

See Also

```
Arotation, rot.matrix, lambert, lambert.inv, rsop
```

Examples

```
a <- rnorm(3)
a <- a/sqrt(sum(a^2))
b <- rnorm(3)
b <- b/sqrt(sum(b^2))
A <- rotation(a, b)
A
a ; b
a %*% t(A)

a <- rnorm(7)
a <- a/sqrt(sum(a^2))
b <- rnorm(7)
b <- b/sqrt(sum(b^2))
A</pre>
```

```
a ; b a %*% t(A)
```

Saddlepoint approximations of the Fisher-Bingham distributions

Saddlepoint approximations of the Fisher-Bingham distributions

Description

It calculates the logarithm of the normalising constant of the Fisher-Bingham distribution.

Usage

```
fb.saddle(gam, lam)
```

Arguments

gam A numeric vector containing the parameters of the Fisher part.

1am All the eigenvalues of the Bingham part. Not just the non zero ones.

Details

It calculate the three approximations given by Kume and Wood (2005) and it uses the Fisher-Bingham parametrization of that paper.

Value

A list including:

first oder The first order approximation
second oder The second order approximation
third oder The third order approximation

Author(s)

Michail Tsagris

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

References

Kume Alfred and Wood Andrew T.A. (2005). Saddlepoint approximations for the Bingham and Fisher-Bingham normalizing constants. Biometrika, 92(2):465-476

```
kent.logcon, rfb, kent.mle, rbingham
```

```
p <- 3 ; k <- 1 
 0.5 * p * log(2 * pi) - (p/2 - 1) * log(k) + log(besselI(k, p/2 - 1, expon.scaled = TRUE)) + k ## normalising constant of the ## von Mises-Fisher distribution fb.saddle( c(0, k, 0), c(0, 0, 0)) ## saddlepoint approximation ## Normalising constant of the Kent distribution fb.saddle( c(0, 10, 0), c(0, -2, 2)) kent.logcon(10, 2)
```

Simulation from a Bingham distribution using any symmetric matrix A

Simulation from a Bingham distribution using any symmetric matrix A

Description

It simulates random values from a Bingham distribution with any given symmetric matrix.

Usage

```
rbingham(n, A)
```

Arguments

n The sample size.
A A symmetric matrix.

Details

The eigenvalues are fist calcualted and then Chris Fallaize and Theo Kypraio's code (f.rbing) is used. The resulting simulated data anre then right multiplied by the eigenvectors of the matrix A.

Value

A matrix with the siumlated data.

Author(s)

Michail Tsagris

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

References

Kent J.T., Ganeiber A.M. and Mardia K.V. (2013). A new method to simulate the Bingham and related distributions in directional data analysis with applications http://arxiv.org/pdf/1310.8110v1.pdf C.J. Fallaize and T. Kypraios (2014). Exact Bayesian Inference for the Bingham Distribution. Statistics and Computing (To appear). http://arxiv.org/pdf/1401.2894v1.pdf

See Also

```
f.rbing, rfb, rvmf, rkent
```

Examples

```
A <- cov(iris[, 1:3])
x <- rbingham(100, A)
```

Simulation from a Matrix Fisher distribution on SO(3)Simulation from a Matrix Fisher distribution on SO(3)

Description

It simulates random samples (rotation matrices) from a Matrix Fisher distribution with any given parameter matrix, F(3x3).

Usage

```
rmatrixfisher(n, F)
```

Arguments

n the sample size.

F An arbitrary 3x3 matrix.

Details

Firstly corresponding Bingham parameter A is determined for a given Matrix Fisher parameter F using John Kent (2013) algorithm and then Bingham samples for parameter A are generated using rbingham code. Finally convert Bingham samples to Matrix Fisher samples according to the Kent (2013) transformation.

Value

An array with simulated rotation matrices.

Author(s)

Anamul Sajib & Chris Fallaize.

R implementation and documentation: Anamul Sajib <sajibstat@du.ac.bd> & Chris Fallaize.

References

Kent J.T., Ganeiber A.M. and Mardia K.V. (2013). A new method to simulate the Bingham and related distributions in directional data analysis with applications. http://arxiv.org/pdf/1310.8110v1.pdf

See Also

```
matrixfisher.mle
```

Examples

```
F \leftarrow matrix(c(85, 11, 41, 78, 39, 60, 43, 64, 48), ncol = 3) / 10 ### An arbitrary F matrix a <math>\leftarrow rmatrixfisher(10, F)
```

Simulation of random values from a Bingham distribution Simulating from a Bingham distribution

Description

It simulates from a Bingham distribution using the code suggested by Kent et al. (2013).

Usage

```
f.rbing(n, lam, fast = FALSE)
```

Arguments

n Sample size.

lam Eigenvalues of the diagonal symmetric matrix of the Bingham distribution.

fast If you want a fast, efficient simulation set this to TRUE.

Details

The user must have calculated the eigenvalues of the diagonal symmetric matrix of the Bingham distribution. The function accepts the q-1 eigenvalues only. This means, that the user must have subtracted the lowest eigenvalue from the rest and give the non zero ones. The function uses rejection sampling and it was written by Chris Fallaize and Theo Kypraios (University of Nottingham) and kindly offered. Any questions on the code can be addressed to one of the two aforementioned people. It is slightly different than the one Ketn et al. (2013) suggests.

Value

A list including:

X The simulated data.

avtry The estimate of M in the rejection sampling. The average number of simulated

values before a value is accepted. If the argument fast is set to TRUE this infor-

mation will not appear.

Author(s)

Michail Tsagris

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

References

Kent J.T., Ganeiber A.M. and Mardia K.V. (2013). A new method to simulate the Bingham and related distributions in directional data analysis with applications. http://arxiv.org/pdf/1310.8110v1.pdf C.J. Fallaize and T. Kypraios (2014). Exact Bayesian Inference for the Bingham Distribution. Statistics and Computing (No volum assigned yet). http://arxiv.org/pdf/1401.2894v1.pdf

See Also

```
rfb, rvmf, rbingham, rkent, link{rsop}
```

Examples

```
x <- f.rbing( 100, c(1, 0.6, 0.1) )
x
```

Simulation of random values from a mixture of von Mises-Fisher distributions

Simulation of random values from a mixture of von Mises-Fisher distributions

Description

The function simulates random values simulation from a given mixture of von Mises-Fisher distributions.

Usage

```
rmixvmf(n, prob, mu, k)
```

Arguments

n The sample size.

prob This is avector with the mixing probability of each group.

mu A matrix with the mean direction of each group.

k A vector with the concentration parameter of each group.

Details

The function simulates random values simulation from a given mixture of von Mises-Fisher distributions using the rvmf function.

Value

A list including:

id An indicator of the group of each simulated vector.

x A matrix with the simulated data.

Author(s)

Michail Tsagris

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

References

Kurt Hornik and Bettina Grun (2014). movMF: An R Package for Fitting Mixtures of von Mises-Fisher Distributions http://cran.r-project.org/web/packages/movMF/vignettes/movMF.pdf

See Also

```
mixvmf.mle, rvmf, bic.mixvmf
```

Examples

```
k <- runif(4, 4, 20)
prob <- c(0.2, 0.4, 0.3, 0.1)
mu <- matrix(rnorm(16), ncol = 4)
mu <- mu / sqrt( rowSums(mu^2) )
x <- rmixvmf(200, prob, mu, k)$x
bic.mixvmf(x, 5)</pre>
```

Simulation of random values from a spherical Fisher-Bingham distribution

Simulation of random values from a spherical Fisher-Bingham distribution

Description

Simulation of random values from a spherical Fisher-Bingham distribution.

Usage

```
rfb(n, k, m, A)
```

Arguments

n	The cample cize
11	The sample size.

k The concentraion parameter (Fisher part). It has to be greater than 0.

m The mean direction (Fisher part).

A A symmetric matrix (Bingham part).

Details

Random values from a spherical Fisher-Bingham distribution are generated. This functions included the option of simulating from a Kent distribution also.

Value

A matrix with the simulated data.

Author(s)

Michail Tsagris

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

References

Kent J.T., Ganeiber A.M. and Mardia K.V. (2013). A new method to simulate the Bingham and related distributions in directional data analysis with applications. http://arxiv.org/pdf/1310.8110v1.pdf

See Also

```
rbingham, rvmf, rkent, f.rbing
```

Examples

```
k <- 15
mu <- rnorm(3)</pre>
mu <- mu / sqrt( sum(mu^2) )</pre>
A <- cov(iris[, 1:3])
x \leftarrow rfb(50, k, mu, A)
vmf.mle(x) ## fits a von Mises-Fisher distribution to the simulated data
## Next we simulate from a Kent distribution
A \leftarrow diag(c(-5, 0, 5))
n <- 100
x <- rfb(n, k, mu, A) ## data follow a Kent distribution
kent.mle(x) ## fits a Kent distribution
vmf.mle(x) ## fits a von Mises-Fisher distribution
A \leftarrow diag(c(5, 0, -5))
n <- 100
x <- rfb(n, k, mu, A) ## data follow a Kent distribution
kent.mle(x) ## fits a Kent distribution
vmf.mle(x) ## fits a von Mises-Fisher distribution
```

Simulation of random values from a spherical Kent distribution

Simulation of random values from a spherical Kent distribution

Description

Simulation of random values from a spherical Kent distribution.

Usage

```
rkent(n, k, m, b)
```

Arguments

n I	Γhe	samp	le	size.
-----	-----	------	----	-------

k The concentration parameter κ . It has to be greater than 0.

m The mean direction (Fisher part).

b The ovalness parameter, β .

Details

Random values from a Kent distribution on the sphere are generated. The function generates from a spherical Kent distribution using rfb with an arbitrary mean direction and then rotates the data to have the desired mean direction.

Value

A matrix with the simulated data.

Author(s)

Michail Tsagris

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

References

Kent J.T., Ganeiber A.M. and Mardia K.V. (2013). A new method to simulate the Bingham and related distributions in directional data analysis with applications. http://arxiv.org/pdf/1310.8110v1.pdf

```
rfb, rbingham, rvmf, f.rbing
```

```
k <- 15
mu <- rnorm(3)
mu <- mu / sqrt( sum(mu^2) )
A <- diag( c(-5, 0, 5) )
x <- rfb(500, k, mu, A)
kent.mle(x)
y <- rkent(500, k, mu, A[3, 3])
kent.mle(y)</pre>
```

Simulation of random values from rotationally symmetric distributions

Simulation of random values from rotationally symmetric distributions

Description

Simulation of random values from rotationally symmetric distributions. The data can be spherical or hyper-spherical.

Usage

```
rvmf(n, mu, k)
riag(n, mu)
```

Arguments

n	The sample size.
mu	A unit vector showing the mean direction for the von Mises-Fisher distribution. The mean vector of the Independent Angular Gaussian distribution. This does not have to be a unit vector.
k	The concentration parameter of the von Mises-Fisher distribution. If $k=0$, random values from the spherical uniform will be drwan.

Details

The von Mises-Fisher uses the rejection smapling suggested by Andrew Wood (1994). For the Independent Angular Gaussian, values are generated from a multivariate normal distribution with the given mean vector and the identity matrix as the covariance matrix. Then each vector becomes a unit vector.

Value

A matrix with the simulated data.

Author(s)

Michail Tsagris

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

References

Wood A. T. A. (1994). Simulation of the von Mises Fisher distribution. Communications in statistics-simulation and computation, 23(1): 157–164.

Dhillon I. S. & Sra S. (2003). Modeling data using directional distributions. Technical Report TR-03-06, Department of Computer Sciences, The University of Texas at Austin. http://citeseerx.ist.psu.edu/viewdoc/download?

See Also

```
vmf.mle, iag.mle rfb, racg, rvonmises, rmixvmf
```

Examples

```
m <- rnorm(4)
m <- m/sqrt(sum(m^2))
x <- rvmf(100, m, 25)
m
vmf.mle(x)</pre>
```

Simulation of random values from some circular distributions

Simulation of random values from some circular distributions

Description

Simulation of random values from some circular distributions.

Usage

```
rvonmises(n, m, k, rads = TRUE)
rwrapcauchy(n, m, rho, rads = TRUE)
rspml(n, mu, rads = TRUE)
rcircbeta(n, m, a, b, rads = TRUE)
```

Arguments

n The sample size.

m The mean angle expressed in radians or degrees.

mu The mean vector of the SPML in \mathbb{R}^2 .

k The concentration parameter of the von Mises distribution. If k is zero the sam-

ple will be generated from the uniform distribution over $(0, 2\pi)$.

rho	The ρ parameter of the Wrapped Cauchy distribution.
а	The α parameter of the beta distribution
b	The β parameter of the beta distribution
rads	If the mean angle is expressed in radians, this should be TRUE and FALSE

otherwise. The simulated data will be expressed in radians or degrees depending

on what the mean angle is expressed.

Details

For the von Mises distribution, the mean direction is transformed to the Euclidean coordinates (i.e. unit vector) and then the rvmf function is employed. It uses a rejection smapling as suggested by Andrew Wood in 1994. I have mentioned the description of the algorithm as I found it in Dhillon and Sra in 2003. Finally, the data are transformed to radians or degrees.

For the wrapped Cauchy distribution the function generates Cauchy values x and then wrapps around the circle $x=x(mod2\pi)$. For the circular beta the function has some extra steps (see Zheng Sun's master thesis).

Value

A vector with the simulated data.

Author(s)

Michail Tsagris

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

References

Wood, A. T. (1994). Simulation of the von Mises Fisher distribution. Communications in statistics-simulation and computation, 23(1): 157-164.

Dhillon, I. S., & Sra, S. (2003). Modeling data using directional distributions. Technical Report TR-03-06, Department of Computer Sciences, The University of Texas at Austin. http://citeseerx.ist.psu.edu/viewdoc/download?

Zheng Sun (2006). Comparing measures of fit for circular distributions. Master thesis, University of Victoria. https://dspace.library.uvic.ca/bitstream/handle/1828/2698/zhengsun_master_thesis.pdf;sequence=1

Lai, M. (1994). Some results in the statistical analysis of directional data. Master thesis, University of Hong Kong.

Presnell Brett, Morrison Scott P. and Littell Ramon C. (1998). Projected multivariate linear models for directional data. Journal of the American Statistical Association, 93(443): 1068-1077.

See Also

```
circ.summary, rvmf, racg
```

Examples

```
x <- rvonmises(100, 2, 25, rads = TRUE)
circ.summary(x, rads = TRUE)</pre>
```

Simulation of random values from the ESAG distribution

Simulation of random values from the ESAG distribution

Description

Simulation of random values from the ESAG distribution.

Usage

```
resag(n, mu, gam)
```

Arguments

n A number; how many vectors you want to generate. mu The mean vector the ESAG distribution, a vector in \mathbb{R}^3 . gam The two gamma parameters of the ESAG distribution.

Details

A random sample from the ESAG distribution is generated. In case the gammas are zero, the sample is drawn from the Independent Angular Gaussian (or projected normal).

Value

An $n \times 3$ matrix with the simulated unit vectors.

Author(s)

Michail Tsagris.

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

Paine P.J., Preston S.P., Tsagris M. and Wood A.T.A. (2018). An Elliptically Symmetric Angular Gaussian Distribution. Statistics and Computing, 28(3):689–697.

See Also

```
esag.mle, desag, spml.mle, acg.mle, circ.summary
```

Examples

```
m <- colMeans( as.matrix( iris[,1:3] ) )
y <- resag(1000, m, c(1, 0.5) )
esag.mle(y)</pre>
```

Spherical and hyperspherical median

Fast calculation of the spherical and hyperspherical median

Description

It calculates, very fast, the (hyper-)spherical median of a sample.

Usage

```
mediandir(x)
mediandir_2(x)
```

Arguments

Χ

The data, a numeric matrix with unit vectors.

Details

The "mediandir" employes a fixed poit iterative algorithm stemming from the first derivative (Cabrera and Watson, 1990) to find the median direction as described by Fisher (1985) and Fisher, Lewis and Embleton (1987). In the big samples this is much much faster than "mediandir_2", since the search is based on iterations.

Value

The median direction.

Author(s)

Michail Tsagris

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

References

Fisher N. I. (1985). Spherical medians. Journal of the Royal Statistical Society. Series B, 47(2): 342-348.

Fisher N. I., Lewis T. and Embleton B. J. (1987). Statistical analysis of spherical data. Cambridge university press.

Cabrera J. and Watson G. S. (1990). On a spherical median related distribution. Communications in Statistics-Theory and Methods, 19(6): 1973-1986.

```
nsmedian, vmf.mle, kent.mle
```

```
m <- rnorm(3)
m <- m / sqrt( sum(m^2) )
x <- rvmf(100, m, 10)
mediandir(x)
mediandir_2(x)
nsmedian(x)</pre>
```

Spherical regression using the projected normal or the von Mises-Fisher distribution

Spherical regression using the projected normal or the von MisesFisher distribution

Description

Spherical regression using the projected normal or the von Mises-Fisher distribution.

Usage

```
iag.reg(y, x, con = TRUE, xnew = NULL, tol = 1e-06)
vmf.reg(y, x, con = TRUE, xnew = NULL, tol = 1e-06)
```

Arguments

У	A matrix with 3 columns containing the (unit vector) spherical data.
х	The predictor variable(s), they can be continuous, spherical, categorical or a mix of them.
con	Do you want the constant term in the regression?
xnew	If you have new data use it, otherwise leave it NULL.
tol	A tolerance value to decide when to stop the successive optimaizations.

Details

The second parametrization of the projected normal and of the von Mises-Fisher regression (Paine et al., 2019) is applied. For more information see the paper by Paine et al. (2019).

Value

A list including:

loglik	The log-likelihood of the regression model.
fit	This is a measure of fit of the estimated values, defined as $\sum_{i=1}^n y_i^T \hat{y}_i$. This appears if the argument "xnew" is NULL.
beta	The beta coefficients.
seb	The standard error of the beta coefficients.

ki	The norm of the fitted values. In the von Mises-Fisher regression this is the concentration parameter of each observation. In the projected normal this are the norms of the fitted values before being projected onto the sphere. This is returned if the argument "xnew" is NULL.
est	The fitted values of xnew if "xnew" is NULL. If it is not NULL, the fitted values

Author(s)

Michail Tsagris

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

for the "xnew" you supplied will be returned.

References

P. J. Paine, S. P. Preston, M. Tsagris and Andrew T. A. Wood (2019). Spherical regression models with general covariates and anisotropic errors. Statistics and Computing (to appear). https://link.springer.com/content/pdf/10.

See Also

```
esag.mle, vmf.mle, spml.reg
```

Examples

```
y <- rvmf(150, rnorm(3), 5)
a1 <- iag.reg(y, iris[, 4])
a2 <- iag.reg(y, iris[, 4:5])
b1 <- vmf.reg(y, iris[, 4])
b2 <- vmf.reg(y, iris[, 4:5])</pre>
```

Spherical-spherical correlation

Spherical-spherical correlation

Description

Correlation between two spherical variables.

Usage

```
spher.cor(x, y)
```

Arguments

x A spherical variable. A matrix with thre columns, each row is a unit vector.

y A spherical variable. A matrix with thre columns, each row is a unit vector.

Details

A very similar to the classical correlation is calcualted. In addition, a hypothesis test for no correlation is performed. Note, that this is a squared correlation actually, so negative values will never be returned.

Value

A vector including:

R-squared The value of the squared correlation.

p-value The p-value of the no correlation hypothesis testing.

Author(s)

Michail Tsagris

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

References

Kanti V. Mardia and Peter E. Jupp. Directional statistics, pg. 254-255.

See Also

```
spher.reg, vmf.mle, circ.cor1, circ.cor2
```

Examples

```
x <- rvmf(100, rnorm(3), 10)
y <- rvmf(100, rnorm(3), 10)
spher.cor(x, y)</pre>
```

```
Spherical-spherical regression

Spherical-Spherical regression
```

Description

It performs regression when both the dependent and independent variables are spherical.

Usage

```
spher.reg(y, x, rads = FALSE)
```

Arguments

Х

y The dependent variable; a matrix with either two columns, latitude and longitude, either in radians or in degrees. Alternatively it is a matrix with three

columns, unit vectors.

The dependent variable; a matrix with either two columns, latitude and lon-

gitude, either in radians or in degrees. Alternatively it is a matrix with three columns, unit vectors. The two matrices must agree in the scale and dimen-

sions.

rads If the data are expressed in latitude and longitude then it matter to know if they

are in radians or degrees. If they are in radians, then this should be TRUE and FALSE otherwise. If the previous argument, euclidean, is TRUE, this one does

not matter what its value is.

Details

Spherical regression as proposed by Chang (1986) is implemented. If the estimated rotation matrix has a determinant equal to -1, singualr value decomposition is performed and the last unit vector of the second matrix is multiplied by -1.

Value

A list including:

A The estimated rotation matrix.

fitted The fitted values in Euclidean coordinates (unit vectors).

Author(s)

Michail Tsagris

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

References

Ted Chang (1986). Spherical Regression. Annals of Statistics, 14(3): 907–924.

See Also

```
spher.cor, spml.reg, circ.cor1, circ.cor2, sphereplot
```

Examples

```
mx <- rnorm(3)
mx <- mx/sqrt( sum(mx^2) )
my <- rnorm(3)
my <- my/sqrt( sum(my^2) )
x <- rvmf(100, mx, 15)
A <- rotation(mx, my)
y <- x %*% t(A)</pre>
```

```
mod <- spher.reg(y, x)
A
mod$A ## exact match, no noise
y <- x %*% t(A)
y <- y + rvmf(100, colMeans(y), 40)
mod <- spher.reg(y, x)
A
mod$A ## noise added, more relistic example</pre>
```

```
Summary statistics for circular data

Summary statistics for circular data
```

Description

It produces a few summary measures for circular data.

Usage

```
circ.summary(u, rads = FALSE, fast = FALSE, tol = 1e-07, plot = FALSE)
```

Arguments

u	A vector with circular data.
rads	If the data are in rads, then this should be TRUE, otherwise FALSE.
fast	A boolean variable to do a faster implementation.
tol	The tolerance level to stop the Newton-Raphson algorithm for finding kappa.
plot	If you want to see the data plotted on a cicrle make this TRUE.

Details

It returns the circular mean, mean resultant length, variance, standard deviation and concentration parameter. So, basically it returns the estimated values of the parameters of the von Mises distribution.

Value

If fast = FALSE a list including all the following. If fast = TRUE less items are returned.

mesos The circular mean direction.

confint The 95% confidence interval for the circular mean direction.

kappa The concentration parameter.

MRL The mean resultant length.

circvariance The circular variance.

circstd The circular standard deviation.

loglik The log-likelihood of the fitted von Mises distribution.

Author(s)

Michail Tsagris

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

See Also

```
spml.mle, rvonmises, vm.kde, vmf.mle, group.vm, hcf.circaov
```

Examples

```
x <- rvonmises(50, 2.5, 15, rads = TRUE)
circ.summary(x, rads = TRUE, plot = TRUE)</pre>
```

```
Summary statistics for grouped circular data

Summary statistics for grouped circular data
```

Description

It produces a few summary measures for grouped circular data.

Usage

```
group.vm(group, fi, rads = FALSE)
```

Arguments

group	A matrix denoting the classes. Each row consists of two numbers, the lower and
	upper points of each class.

fi The frequency of each class of data.

rads If the data are in rads, then this should be TRUE, otherwise FALSE.

Details

It returns the circular mean, mean resultant length, variance, standard deviation and concentration parameter. So, basically it returns the estimated values of the parameters of the von Mises distribution. The mena resultant length though is group corrected.

Value

A list including:

mesos The circular mean direction.

confint The 95% confidence interval for the circular mean direction.

kappa The concentration parameter.

MRL The mean resultant length.

circvariance The circular variance.

circstd The circular standard deviation.

Author(s)

Michail Tsagris

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

References

Pewsey Arthur, Markus Neuhauser and Graeme D. Ruxton (2013). Circular statistics in R. Oxford University Press.

Mardia K. V. and Jupp P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

See Also

```
circ.summary, rvonmises, vm.kde
```

Examples

```
x <- rvonmises(200, 3, 10)
a <- circ.summary(x, rads = TRUE, plot = FALSE)
group <- seq(min(x) - 0.1, max(x) + 0.1, length = 6)
y <- cut(x, breaks = group, length = 6)
group <- matrix( c( group[1], rep(group[2:5], each = 2), group[6]), ncol = 2, byrow = TRUE)
fi <- as.vector( table(y) )
b <- group.vm(group, fi, rads = TRUE)
a
b</pre>
```

Test for a given mean direction

Test for a given mean direction

Description

A log-likelihood ratio test for testing whether the sample mena direction is equal to some predefined one.

Usage

```
meandir.test(x, mu, B = 999)
```

Arguments

x A matrix with the data, unit vectors.

mu A unit vector with the hypothesized mean direction.

B A number either 1, so no bootstrap calibration is performed or more than 1, so

bootstrap calibration is performed.

Details

The log-likelihood ratio test is performed.

Value

A list including:

mean.dir The sample mean direction pvalue The p-value of the test.

Author(s)

Michail Tsagris.

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

See Also

```
vmf.mle, kent.mle, rayleigh
```

Examples

```
mu <- rnorm(5)
mu <- mu / sqrt( sum(mu^2) )
x <- rvmf(100, mu, 10)
meandir.test(x, mu, 1)
meandir.test(x, mu, 499)</pre>
```

Test for equality of concentration parameters for spherical data

Test for equality of concentration parameters for spherical data

Description

This tests the equality of concentration parameters for spherical data only.

Usage

```
spherconc.test(x, ina)
```

Arguments

A matrix with the data in Euclidean coordinates, i.e. unit vectors
 A variable indicating the groupings of the observations.

Details

The test is designed for spherical data only.

Value

A list including:

mess A message stating the value of the mean resultant and which test statistic was

used, U1, U2 or U3.

res A vector containing the test statistic and its p-value.

Author(s)

Michail Tsagris

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

References

Kanti V. Mardia and Peter E. Jupp. Directional statistics, pg. 226-227.

See Also

```
het.aov, lr.aov, embed.aov, hcf.aov, conc.test, sphereplot
```

Examples

```
x <- rvmf(100, rnorm(3), 15)
ina <- rep(1:4, each = 25)
spherconc.test(x, ina)</pre>
```

Test of equality of the concentration parameters for circular data $A \ test \ for \ testing \ the \ equality \ of \ the \ concentration \ parameter \ among \ g \\ samples, \ where \ g >= 2 \ for \ circular \ data$

Description

A test for testing the equality of the concentration parameter among g samples, where $g \ge 2$ for circular data.

Usage

```
conc.test(u, ina, rads = FALSE)
```

Arguments

u A numeric vector containing the values of all samples.

ina A numerical variable or factor indicating the groups of each value.

rads If the data are in radians this should be TRUE and FALSE otherwise.

Details

This test works for circular data.

Value

A list including:

mess A message informing the use of the test statistic used.

res A numeric vector containing the value of the test statistic and its associated p-

value.

Author(s)

Michail Tsagris

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

See Also

```
embed.circaov, hcf.circaov, lr.circaov, het.circaov
```

Examples

```
x <- rvonmises(100, 2.4, 15)
ina <- rep(1:4,each = 25)
conc.test(x, ina, rads = TRUE)</pre>
```

The k-nearest neighbours using the cosinus distance

The k-nearest neighbours using the cosinus distance

Description

The k-nearest neighbours using the cosinus distance.

Usage

```
cosnn(xnew, x, k = 5, index = FALSE, rann = FALSE)
```

Arguments

xnew	The new data whose k-nearest neighbours are to be found.
X	The data, a numeric matrix with unit vectors.
k	The number of nearest neighbours, set to 5 by default. It can also be a vector with many values.
index	If you want the indices of the closest observations set this equal to TRUE.
rann	If you have large scale datasets and want a faster k-NN search, you can use kd-trees implemented in the R package "RANN". In this case you must set this argument equal to TRUE.

Details

The shortest distances or the indices of the k-nearest neighbours using the cosinus distance are returned.

Value

A matrix with the shortest distance of each xnew from x, if index is FALSE, or the indices of the nearest neighbours of each xnew from x, if index is TRUE.

Author(s)

Michail Tsagris.

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

References

Tsagris M. and Alenazi A. (2019). Comparison of discriminant analysis methods on the sphere. Communications in Statistics: Case Studies, Data Analysis and Applications, 5(4), 467–491.

See Also

```
dirknn, dirknn.tune
```

Examples

```
xnew <- rvmf(10, rnorm(3), 5)
x <- rvmf(50, rnorm(3), 5)
a <- cosnn(xnew, x, k = 5)
b <- cosnn(xnew, x, k = 5, index = TRUE)</pre>
```

Transform unit vectors to angular data

Transform unit vectors to angular data

Description

Transform unit vectors to angular data.

Usage

```
etoa(x)
```

Arguments

х

A numerical matrix with directional data, i.e. unit verctors.

Details

from the Euclidean coordinates the data are mapped to angles, expressed in rads.

Value

A list including:

mu

A matrix with angles. The number of columns is one less than that of the original matrix.

Author(s)

Michail Tsagris

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

References

https://en.wikipedia.org/wiki/N-sphere#Spherical_coordinates

See Also

```
vmnb.pred, weibull.nb
```

Examples

```
x <- rvmf(10, rnorm(3), 5)
y <- etoa(x)</pre>
```

Tuning of the bandwidth parameter in the von Mises kernel

Tuning of the bandwidth parameter in the von Mises kernel for circular data

Description

Tuning of the bandwidth parameter in the von Mises kernel for circular data. Cross validation is used.

Usage

```
vmkde.tune(u, low = 0.1, up = 1, rads = TRUE)
```

Arguments

u The data, a numerical vector.

low The lower value of h to search.

up The lower value of h to search.

rads If the data are in radians this should be TRUE and FALSE otherwise.

Details

Tuning of the bandwidth parameter in the von Mises kernel for circula data via cross validation. The procedure is fast because an optimiser is used.

Value

A vector including two elements:

Optimal h The best H found.

cv The value of the maximised pseudo-likelihood.

Author(s)

Michail Tsagris

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

References

Taylor C. C. (2008). Automatic bandwidth selection for circular density estimation. Computational Statistics & Data Analysis, 52(7), 3493–3500.

Wand M. P., & Jones M. C. (1994). Kernel smoothing. CrC Press.

See Also

```
vm.kde, vmfkde.tune, vmf.kde
```

Examples

```
u \leftarrow rvonmises(100, 2.4, 10, rads = TRUE)
vmkde.tune(u)
```

Tuning of the bandwidth parameter in the von Mises-Fisher kernel

Tuning of the bandwidth parameter in the von Mises-Fisher kernel for

(hyper-)spherical data

Description

Tuning of the bandwidth parameter in the von Mises-Fisher kernel for (hyper-)spherical data whit cross validation.

Usage

```
vmfkde.tune(x, low = 0.1, up = 1)
```

Arguments

x A matrix with the data in Euclidean cordinates, i.e. unit vectors.

The lower value of the bandwdith to search.The upper value of the bandwdith to search.

Details

Fast tuning of the bandwidth parameter in the von Mises-Fisher kernel for (hyper-)spherical data via cross validation.

Value

A vector including two elements:

Optimal h The best H found.

cv The value of the maximised pseudo-likelihood.

Author(s)

Michail Tsagris

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

References

Garcia Portugues E. (2013). Exact risk improvement of bandwidth selectors for kernel density estimation with directional data. Electronic Journal of Statistics, 7, 1655–1685.

Wand M. P., and Jones M. C. (1994). Kernel smoothing. Crc Press.

See Also

```
vmf.kde,vmf.kerncontour, vm.kde, vmkde.tune
```

Examples

```
x <- rvmf(100, rnorm(3), 15)
vmfkde.tune(x)</pre>
```

Tuning of the k-NN algorithm using the arc cosinus distance k-NN algorithm using the arc cosinus distance. Tuning the k neighbours

Description

It estimates the percentage of correct classification via an m-fold cross validation.

Usage

```
dirknn.tune(ina, x, k = 2:10, mesos = TRUE, nfolds = 10, folds = NULL,
parallel = FALSE, stratified = TRUE, seed = NULL, rann = FALSE)
```

Arguments

x The data, a numeric matrix with unit vectors.ina A variable indicating the groups of the data x.

nfolds How many folds to create?

k A vector with the number of nearest neighbours to consider.

mesos A boolean variable used only in the case of the non standard algorithm (type="NS").

Should the average of the distances be calculated (TRUE) or not (FALSE)? If it

is FALSE, the harmonic mean is calculated.

folds Do you already have a list with the folds? If not, leave this NULL.

parallel If you want the standard -NN algorithm to take place in parallel set this equal to

TRUE.

stratified Should the folds be created in a stratified way? i.e. keeping the distribution of

the groups similar through all folds?

seed If seed is TRUE, the results will always be the same.

rann If you have large scale datasets and want a faster k-NN search, you can use kd-

trees implemented in the R package "RANN". In this case you must set this

argument equal to TRUE.

Details

The standard algorithm is to keep the k nearest observations and see the groups of these observations. The new observation is allocated to the most frequent seen group. The non standard algorithm is to calculate the classical mean or the harmonic mean of the k nearest observations for each group. The new observation is allocated to the group with the smallest mean distance.

We have made an eficient (not very much efficient though) memory allocation. Even if you have hundreds of thousands of observations, the computer will not clush, it will only take longer. Instead of calculate the distance matrix once in the beginning we calculate the distances of the out-of-sample observations from the rest. If we calculated the distance matrix in the beginning, once, the resulting matrix could have dimensions thousands by thousands. This would not fit into the memory. If you have a few hundres of observations, the runtime is about the same (maybe less, maybe more) as calculating the distance matrix in the first place.

Value

A list including:

per The average percent of correct classification across the neighbours.

percent The bias corrected percent of correct classification.

runtime The run time of the algorithm. A numeric vector. The first element is the user

time, the second element is the system time and the third element is the elapsed

time.

Author(s)

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

References

Tsagris M. and Alenazi A. (2019). Comparison of discriminant analysis methods on the sphere. Communications in Statistics: Case Studies, Data Analysis and Applications, 5(4), 467–491.

See Also

```
dirknn, vmf.da, mixvmf.mle
```

Examples

```
k <- runif(4, 4, 20)
prob <- c(0.2, 0.4, 0.3, 0.1)
mu <- matrix(rnorm(16), ncol = 4)
mu <- mu / sqrt( rowSums(mu^2) )
da <- rmixvmf(200, prob, mu, k)
x <- da$x
ina <- da$id
dirknn.tune(ina, x, k = 2:6, nfolds = 5, mesos = TRUE)
dirknn.tune(ina, x, k = 2:6, nfolds = 10, mesos = TRUE)</pre>
```

Tuning of the k-NN regression

Tuning of the k-NN regression with Euclidean or (hyper-)spherical response and or predictor variables

Description

Tuning of the k-NN regression with Euclidean or (hyper-)spherical response and or predictor variables. It estimates the percentage of correct classification via an m-fold cross valdiation. The bias is estimated as well using the algorithm suggested by Tibshirani and Tibshirani (2009) and is subtracted.

Usage

```
knnreg.tune(y, x, nfolds = 10, A = 10, ncores = 1, res = "eucl",
estim = "arithmetic", folds = NULL, seed = NULL, graph = FALSE)
```

Arguments

У	The currently available data, the response variables values. A matrix with either euclidean (univariate or multivariate) or (hyper-)spherical data. If you have a circular response, say u, transform it to a unit vector via (cos(u), sin(u)).
х	The currently available data, the predictor variables values. A matrix with either euclidean (univariate or multivariate) or (hyper-)spherical data. If you have a circular response, say u, transform it to a unit vector via (cos(u), sin(u)).
nfolds	How many folds to create?

A	The maximum number of nearest neighbours, set to 5 by default, starting from the 1 nearest neighbor.
ncores	How many cores to use. This is taken into account only when the predictor variables are spherical.
res	The type of the response variable. If it is Euclidean, set this argument equal to "res". If it is a unit vector set it to res="spher".
estim	Once the k observations whith the smallest distance are discovered, what should the prediction be? The arithmetic average of the corresponding y values be used estim="arithmetic" or their harmonic average estim="harmonic".
folds	Do you already have a list with the folds? If not, leave this NULL.
seed	You can specify your own seed number here or leave it NULL.
graph	If this is TRUE a graph with the results will appear.

Details

Tuning of the k-NN regression with Euclidean or (hyper-)spherical response and or predictor variables. It estimates the percentage of correct classification via an m-fold cross valdiation. The bias is estimated as well using the algorithm suggested by Tibshirani and Tibshirani (2009) and is subtracted. The sum of squares of prediction is used in the case of Euclidean responses. In the case of spherical responses the $\sum_{\hat{y}_i^T} y_i$ is calculated.

Value

A list including:

crit	The value of the	criterion to	minimise/maximise	for all	values of the nearest

neighbours.

best_k The best value of the nearest neighbours.

performance The bias corrected optimal value of the criterion, along wit the estimated bias.

For the case of Euclidean reponse this will be higher than the crit and for the

case or spherical responses it will be lower than crit.

runtime The run time of the algorithm. A numeric vector. The first element is the user

time, the second element is the system time and the third element is the elapsed

time.

Author(s)

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

See Also

knn.reg, spher.reg, dirknn.tune

Examples

```
y <- iris[, 1]
x <- iris[, 2:4]
x <- x/ sqrt( rowSums(x^2) )  ## Euclidean response and spherical predictors
knnreg.tune(y, x, A = 5, res = "eucl", estim = "arithmetic")

y <- iris[, 1:3]
y <- y/ sqrt( rowSums(y^2) )  ## Spherical response and Euclidean predictor
x <- iris[, 2]
knnreg.tune(y, x, A = 5, res = "spher", estim = "arithmetic")</pre>
```

Uniformity test for circular data

Uniformity tests for circular data.

Description

Hypothesis tests of uniformity for circular data.

Usage

```
kuiper(u, rads = FALSE, R = 1)
watson(u, rads = FALSE, R = 1)
```

Arguments

A numeric vector containing the circular data, which cna be expressed in degrees or radians.
 A boolean variable. If the data are in radians, put this TRUE. If the data are expressed in degrees make this FALSE.
 If R = 1the asymtptotic p-value will be calcualted. If R is greater than 1 the

bootstrap p-value is returned.

Details

The high concentration (hcf.circaov), log-likelihood ratio (lr.circaov), embedding approach (embed.circaov) or the non equal concentration parameters approach (het.circaov) is used.

Value

A vector including:

Test The value of the test statistic.

p-value The p-value of the test (bootstrap or asymptotic depends upon the value of the

argument R).

Author(s)

Michail Tsagris.

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

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, pg. 153-55 (Kuiper's test) & 156-157 (Watson's test).

See Also

```
rayleigh, vmf.mle, rvonmises
```

Examples

```
x <- rvonmises(n = 40, m = 2, k = 10)
kuiper(x, rads = TRUE)
watson(x, rads = TRUE)
x <- rvonmises(40, m = 2, k = 0)
kuiper(x, rads = TRUE)
watson(x, rads = TRUE)</pre>
```

von Mises kernel density estimation

Kernel density estimation of circular data with a von Mises kernel

Description

Kernel density estimation of circular data with a von Mises kernel.

Usage

```
vm.kde(u, h, thumb = "none", rads = TRUE)
```

Arguments

u A numeric vector containing the data.

h The bandwidth.

thumb It can be either "none", so the bandwidth the user has set will be used, "tay" for

the method of Taylor (2008) or "rot" for the method of Garcia-Portugues (2013).

rads If the data are in radians, this should be TRUE and FALSE otherwise.

Details

The user has the option to use a bandwidth he/she has found in some way (cross-validation) or estimate it as Taylor (2008) or Garcia-Portugues (2013).

Value

A list including:

h The bandwidth. If the user chose one of "tay" or "rot" the estimated bandwidth will be returned.

f The kernel density estimate at the observed points.

Author(s)

Michail Tsagris

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

References

Taylor, C. C. (2008). Automatic bandwidth selection for circular density estimation. Computational Statistics & Data Analysis, 52(7): 3493-3500.

Garcia Portugues, E. (2013). Exact risk improvement of bandwidth selectors for kernel density estimation with directional data. Electronic Journal of Statistics, 7, 1655-1685.

See Also

```
vmkde.tune, vmfkde.tune, vmf.kde
```

Examples

```
x \leftarrow rvonmises(100, 2.4, 10, rads = TRUE)
hist(x, freq = FALSE)
f1 \leftarrow vm.kde(x, h = 0.1, thumb = "rot", rads = TRUE)$f
f2 \leftarrow vm.kde(x, h = 0.1, thumb = "tay", rads = TRUE)$f
h \leftarrow vmkde.tune(x)[1]
f3 \leftarrow vm.kde(x, h = h, thumb = "none", rads = TRUE)$f
points(x, f1, col = 1)
points(x, f2, col = 2)
points(x, f3, col = 3)
```

von Mises-Fisher kernel density estimation for (hyper-)spherical data

Kernel density estimation for (hyper-)spherical data using a von

Mises-Fisher kernel

Description

A von Mises-Fisher kernel is used for the non parametric density estimation.

Usage

```
vmf.kde(x, h, thumb = "none")
```

Arguments

x A matrix with unit vectors, i.e. the data being expressed in Euclidean cordinates.

h The bandwidth to be used.

thumb If this is "none", the given bandwidth is used. If it is "rot" the rule of thumb

suggested by Garcia-Portugues (2013) is used.

Details

A von Mises-Fisher kernel is used for the non parametric density estimation.

Value

A list including:

h The bandwidth used.

f A vector with the kernel density estimate calculated for each of the data points.

Author(s)

Michail Tsagris

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

References

Garcia Portugues, E. (2013). Exact risk improvement of bandwidth selectors for kernel density estimation with directional data. Electronic Journal of Statistics, 7, 1655-1685.

See Also

```
vmfkde.tune, vm.kde, vmf.mle, vmkde.tune
```

Examples

```
x <- rvmf(100, rnorm(5), 15)
h <- vmfkde.tune(x)[1]
f1 <- vmf.kde(x, h = h, thumb = "none")
f2 <- vmf.kde(x, h = h, thumb = "rot")
f1$h ; f2$h</pre>
```

Index

* Analysis of variance	A test for testing the equality of
Anova for circular data, 9	the concentration parameters
* Angle of rotation	for ciruclar data,5
Rotation axis and angle of	Anova for circular data,9
rotation given a rotation	Test of equality of the
matrix, 87	concentration parameters for
Rotation matrix from a rotation	circular data,114
axis and angle of rotation, 88	Uniformity test for circular data,
* Angular central Gaussian distribution	123
Angular central Gaussian random	* Circular regression
values simulation, 7	Circular or angular regression, 17
* Anova	* Circular-linear correlation
Directional-package, 4	Circular-linear correlation, 19
* Axis of rotation	* Concentration parameters
Rotation axis and angle of	Test for equality of concentration
rotation given a rotation	parameters for spherical data,
matrix, 87	113
Rotation matrix from a rotation	* Contour plot
axis and angle of rotation, 88	Contour plot of a mixture of von
* BIC of the mixture models	Mises-Fisher distributions
BIC for the model based clustering	mode1, 22
using mixtures of von	Contour plot of spherical data
Mises-Fisher distributions, 10	using a von Mises-Fisher
* Bimodal distribution on the sphere	kernel density estimate, 23
MLE of the Wood bimodal	Contour plot of the Kent
distribution on the sphere, 73	distribution for some data, 24
* Bingham distribution	Contour plot of the Kent
Simulation from a Bingham	distribution without any data,
distribution using any	25
symmetric matrix A,93	Contour plots of the von
Simulation of random values from a	Mises-Fisher distribution, 27
Bingham distribution, 95	* Cross validation
* Circular correlation type II	Cross validation in von
Circular correlations between two	Mises-Fisher discrminant
circular variables, 16	analysis, 33
* Circular correlation type I	Cross validation with ESAG
Circular correlations between two	discrminant analysis, 34
circular variables, 16	* Cross-validation
* Circular data	Tuning of the k-NN regression, 121

* Directional data	Hypothesis test for IAG		
Directional-package,4	distribution over the ESAG		
* Directional k-NN algorithm	distribution, 48		
k-NN algorithm using the arc	Hypothesis test for von		
cosinus distance, 54	Mises-Fisher distribution		
Tuning of the k-NN algorithm using	over Kent distribution, 49		
the arc cosinus distance, 119	* Graphs		
* Discriminant analysis	Directional-package,4		
Cross validation for estimating	* Grouped data		
the classification rate, 31	Goodness of fit test for grouped		
Cross validation in von	data, 45		
Mises-Fisher discrminant	Summary statistics for grouped		
analysis, 33	circular data,110		
Cross validation with ESAG	* Hypothesis testing		
discrminant analysis, 34	A test for testing the equality of		
Directional-package, 4	the concentration parameters		
Prediction in discriminant	for ciruclar data, 5		
analysis based on ESAG	Anova for circular data, 9		
distribution, 79	Test for equality of concentration		
Prediction in discriminant	parameters for spherical data,		
analysis based on von	113		
Mises-Fisher distribution, 80	Test of equality of the		
* ESAG distribution	concentration parameters for		
MLE of the ESAG distribution, 68	circular data, 114		
Simulation of random values from	Uniformity test for circular data,		
the ESAG distribution, 103	123		
* Equality of concentrations	* Hypothesis test		
A test for testing the equality of	Rayleigh's test of uniformity, 85		
the concentration parameters	* Inverse transformation		
for ciruclar data, 5	Inverse of Lambert's equal area		
Test of equality of the	projection, 52		
concentration parameters for	* Kent distribution		
circular data, 114	Contour plot of the Kent		
* Euclidean coordinates	distribution for some data, 24		
Euclidean transformation, 39	Contour plot of the Kent		
Inverse of the Euclidean	distribution without any data,		
transformation, 53	25		
* Euclidean data	Hypothesis test for IAG		
k-NN regression, 55	distribution over the ESAG		
Tuning of the k-NN regression, 121	distribution, 48		
* Fisher-Bingham distribution	Hypothesis test for von		
Saddlepoint approximations of the	Mises-Fisher distribution		
Fisher-Bingham distributions,	over Kent distribution, 49		
92	Logarithm of the Kent distribution		
Simulation of random values from a	normalizing constant, 58		
spherical Fisher-Bingham	MLE of the Kent distribution, 69		
distribution, 97	Simulation of random values from a		
* Goodness of fit test	spherical Fisher-Bingham		

distribution, 97	normalizing constant, 58
Simulation of random values from a	Saddlepoint approximations of the
spherical Kent distribution, 99	Fisher-Bingham distributions,
* Kernel density estimate	92
Tuning of the bandwidth parameter	* One sample hypothesis test
in the von Mises kernel, 117	Test for a given mean direction,
Tuning of the bandwidth parameter	112
in the von Mises-Fisher	* Probability density function
kernel, 118	Probability density function of
von Mises-Fisher kernel density	the von Mises-Fisher
estimation for	distribution, 83
(hyper-)spherical data, 125	* Projected normal
* Kernel density	Circular or angular regression, 17
von Mises kernel density	* Random values simulation
estimation, 124	Simulation of random values from a
* Lambert's equal area projection	Bingham distribution, 95
Inverse of Lambert's equal area	Simulation of random values from a
projection, 52	mixture of von Mises-Fisher
Lambert's equal area projection, 57	distributions, 96
* Matrix Fisher distribution	Simulation of random values from
MLE of the Matrix Fisher	rotationally symmetric
distribution on $SO(3)$, 71	distributions, 100
* Maximum likelihood estimation	Simulation of random values from
MLE of the Matrix Fisher	some circular distributions,
distribution on $SO(3)$, 71	101
* Mean direction	* Regression
Test for a given mean direction,	Directional-package, 4
112	* Rejection sampling
* Median direction	Simulation of random values from a
Maps of the world and the	Bingham distribution, 95
continents, 60	* Rotation matrix
Spherical and hyperspherical	Random sample of matrices in
median, 104	SO(p), 84
* Mixtures of von Mises-Fisher distributions	Rotation matrix from a rotation
BIC for the model based clustering	axis and angle of rotation, 88
using mixtures of von	Rotation matrix to rotate a
Mises-Fisher distributions, 10	spherical vector along the
Contour plot of a mixture of von	direction of another, 90
Mises-Fisher distributions	* SO(p)
model, 22	Random sample of matrices in
Simulation of random values from a mixture of von Mises-Fisher	SO(p), 84
	* Saddlepoint approximation
distributions, 96	Logarithm of the Kent distribution
* Model based clustering Mixtures of Von Mises-Fisher	normalizing constant, 58
distributions, 61	Saddlepoint approximations of the
* Normalising constant	Fisher-Bingham distributions, 92
Logarithm of the Kent distribution	* Simulated data
rogarithm or the vell attribution	↑ Simulateu uata

Simulation of random values from a	Tuning of the bandwidth parameter	
spherical Fisher-Bingham	in the von Mises kernel, 117	
distribution, 97	* Uniform distribution on the	
Simulation of random values from a	(hyper)-sphere	
spherical Kent distribution, 99	Rayleigh's test of uniformity, 85	
* Simulation of random values	* Uniformity test	
Simulation from a Bingham	Uniformity test for circular data,	
distribution using any	123	
symmetric matrix A,93	* Von Mises distribution	
* Simulation	Summary statistics for circular	
Directional-package, 4	data, 109	
* Spherical coordinates	Summary statistics for grouped	
Euclidean transformation, 39	circular data,110	
Inverse of the Euclidean	* Von Mises-Fisher distributions	
transformation, 53	Cross validation for estimating	
* Spherical data	the classification rate, 31	
Directional-package, 4	Prediction in discriminant	
k-NN regression, 55	analysis based on ESAG	
Lambert's equal area projection, 57	distribution, 79	
Spherical-spherical correlation,	Prediction in discriminant	
106	analysis based on von	
Spherical-spherical regression,	Mises-Fisher distribution, 80	
107	* Wood distribution	
Test for equality of concentration	MLE of the Wood bimodal	
parameters for spherical data,	distribution on the sphere, 73	
113	* bivariate angular Gaussian	
Tuning of the k-NN regression, 121	MLE of some circular	
* Spherical-Spherical regression	distributions, 65	
Spherical-spherical regression,	* circular data	
107	Directional-package,4	
* Squared correlation	MLE of some circular	
Spherical-spherical correlation,	distributions, 65	
106	* directional data	
* Summary statistics	Angular central Gaussian random	
Summary statistics for circular	values simulation, 7	
data, 109	Conversion of cosines to azimuth	
Summary statistics for grouped	and plunge, 28	
circular data,110	* k-NN regression	
* Supervised classification	k-NN regression, 55	
k-NN algorithm using the arc	Tuning of the k-NN regression, 121	
cosinus distance, 54	* maximum likelihood estimation	
Tuning of the k-NN algorithm using	MLE of the ESAG distribution, 68	
the arc cosinus distance, 119	MLE of the Kent distribution, 69	
* Tuning of the bandwidth	* random values simulation	
Tuning of the bandwidth parameter	Angular central Gaussian random	
in the von Mises-Fisher	values simulation, 7	
kernel, 118	* simulation	
* Tuning the bandwidth	Simulation of random values from	

the ESAG distribution, 103	* von Mises-Fisher
* spherical data	von Mises-Fisher kernel density
MLE of the ESAG distribution, 68	estimation for
Simulation of random values from	(hyper-)spherical data, 125
the ESAG distribution, 103	* wrapped Cauchy distribution
* unit vectors	MLE of some circular
Generation of unit vector(s) with	distributions, 65
a given angle,44	
* von Mises distribution	A test for testing the equality of the
Tuning of the bandwidth parameter	concentration parameters for
in the von Mises kernel, 117	ciruclar data,5
* von Mises kernel	acg.mle, 7, 69, 86, 103
von Mises kernel density	acg.mle(MLE of (hyper-)spherical
estimation, 124	distributions), 62
* von Mises-Fisher distribution	africa(Maps of the world and the
Contour plots of the von	continents), 60
Mises-Fisher distribution, 27	Angular central Gaussian random values
Cross validation in von	simulation, 7
Mises-Fisher discrminant	Anova for (hyper-)spherical data, 8
analysis, 33	Anova for circular data, 9
Cross validation with ESAG	Arotation, 29, 30, 40, 53, 84, 89, 91
discrminant analysis, 34	Arotation (Rotation axis and angle of
Goodness of fit test for grouped	rotation given a rotation
data, 45	matrix), 87
Hypothesis test for IAG	asia (Maps of the world and the
distribution over the ESAG	continents), 60
distribution, 48	BIC for the model based clustering
Hypothesis test for von	using mixtures of von
Mises-Fisher distribution	Mises-Fisher distributions, 10
over Kent distribution, 49	bic.mixvmf, 62, 97
Mixtures of Von Mises-Fisher	bic.mixvmf(BIC for the model based
distributions, 61	clustering using mixtures of
Probability density function of	von Mises-Fisher
the von Mises-Fisher	distributions), 10
distribution, 83	Bootstrap 2-sample mean test for
Simulation of random values from	(hyper-)spherical data, 11
rotationally symmetric	Bootstrap 2-sample mean test for
distributions, 100	circular data, 13
Simulation of random values from	22. 22.2. 22.2., 22
some circular distributions,	cardio.mle(MLE of some circular
101	distributions), 65
Tuning of the bandwidth parameter	Check visually whether matrix Fisher
in the von Mises-Fisher	samples is correctly generated
kernel, 118	or not, 14
* von Mises-Fisher kernel	circ.cor1, 18, 19, 72, 107, 108
Contour plot of spherical data	circ.cor1(Circular correlations
using a von Mises-Fisher	between two circular
kernel density estimate, 23	variables), 16

circ.cor2, 17-19, 107, 108	Contour plot of the Kent distribution
circ.cor2(Circular correlations	for some data, 24
between two circular	Contour plot of the Kent distribution
variables), 16	without any data, 25
circ.cors1 (Circular correlations	Contour plots of the von Mises-Fisher
between one and many circular	distribution, 27
variables), 15	Conversion of cosines to azimuth and
circ.summary, 5, 46, 66, 69, 83, 102, 103, 111	plunge, 28
circ.summary (Summary statistics for	Converting a rotation matrix on SO(3)
circular data), 109	to an unsigned unit
circbeta.mle (MLE of some circular	quaternion, 29
distributions), 65	Converting an unsigned unit quaternion
circexp.mle (MLE of some circular	to rotation matrix on $SO(3)$, 30
distributions), 65	cosap (Conversion of cosines to
circlin.cor, 17, 18	azimuth and plunge), 28
circlin.cor(Circular-linear	cosnn, 48
correlation), 19	cosnn(The k-nearest neighbours using
Circular correlations between one and	the cosinus distance), 115
many circular variables, 15	Cross validation for estimating the
Circular correlations between two	classification rate, 31
circular variables, 16	Cross validation in von Mises-Fisher
Circular or angular regression, 17	discrminant analysis, 33
Circular-linear correlation, 19	Cross validation with ESAG discrminant
colspml.mle, 68	analysis, 34
colspml.mle(Column-wise MLE of the	
angular Gaussian and the von	deardia (Deneity of some circular
Mises Fisher distributions), 20	dcardio(Density of some circular distributions),37
**	**
Column-wise MLE of the angular	dcircbeta (Density of some circular
Gaussian and the von Mises	distributions), 37
Fisher distributions, 20	dcircpurka (Density of some circular
Column-wise uniformity Watson test for	distributions), 37
circular data, 21	Density of some (hyper-)spherical
colvm.mle(Column-wise MLE of the	distributions, 35
angular Gaussian and the von	Density of some circular
Mises Fisher distributions), 20	distributions, 37
colwatsons (Column-wise uniformity	Density of the spherical Kent and ESAG
Watson test for circular	distributions, 38
data), 21	desag, 5, 38, 69, 103
conc. test, 6, 9, 10, 12, 77, 114	desag (Density of the spherical Kent
conc.test (Test of equality of the	and ESAG distributions), 38
concentration parameters for	dggvm (Density of some circular
circular data), 114	distributions), 37
Contour plot of a mixture of von	dirda.cv, 44
Mises-Fisher distributions	dirda.cv (Cross validation for
model, 22	estimating the classification
Contour plot of spherical data using a	rate), 31
von Mises-Fisher kernel	Directional-package, 4
density estimate, 23	dirknn, 32, 34, 35, 48, 80, 81, 116, 121

dirknn(k-NN algorithm using the arc	etoa(Transform unit vectors to
cosinus distance), 54	angular data), 116
dirknn.tune, 55, 116, 122	euclid, 28, 51, 53, 57
dirknn.tune(Tuning of the k-NN	euclid (Euclidean transformation), 39
algorithm using the arc	euclid.inv, 28, 40
cosinus distance), 119	euclid.inv(Inverse of the Euclidean
dkent, 38	transformation), 53
dkent (Density of the spherical Kent	Euclidean transformation, 39
and ESAG distributions), 38	eul2rot, 28, 41
dpurka (Density of some	eul2rot(Rotation matrix on SO(3) from
(hyper-)spherical	three Euler angles), 89
distributions), 35	Euler angles from a rotation matrix on
dspml (Density of some circular	SO(3), 40
distributions), 37	europe (Maps of the world and the
dvm (Density of some circular	continents), 60
distributions), 37	,,
<pre>dvmf(Density of some (hyper-)spherical</pre>	f.rbing, 94, 98, 99
distributions), 35	f.rbing(Simulation of random values
dwrapcauchy (Density of some circular	from a Bingham distribution),
distributions), 37	95
d13t1 1bdt10f13), 37	fb.saddle, 58, 70
	fb.saddle(Saddlepoint approximations
embed.aov, <i>114</i>	of the Fisher-Bingham
embed.aov(Anova for (hyper-)spherical	distributions), 92
data), 8	fishkent, 22, 49
embed.boot(Bootstrap 2-sample mean	fishkent (Hypothesis test for von
test for (hyper-)spherical	Mises-Fisher distribution
data), 11	over Kent distribution), 49
embed.circaov, 6, 115	Forward Backward Early Dropping
embed.circaov(Anova for circular	selection for circular data
data), 9	using the SPML regression, 42
<pre>embed.perm(Permutation based 2-sample</pre>	doing the offic regression, 12
mean test for	Generate random folds for
(hyper-)spherical data), 76	cross-validation, 43
<pre>embedcirc.boot(Bootstrap 2-sample</pre>	Generation of unit vector(s) with a
mean test for circular data),	given angle, 44
13	ggvm.mle, 37
<pre>embedcirc.perm (Permutation based</pre>	ggvm.mle(MLE of some circular
2-sample mean test for	distributions), 65
circular data),78	Goodness of fit test for grouped data,
esag.da, <i>32</i> , <i>80</i>	45
esag.da(Cross validation with ESAG	group.gof, 83
discrminant analysis), 34	group.gof(Goodness of fit test for
esag.mle, 36, 39, 49, 74, 103, 106	grouped data), 45
esag.mle (MLE of the ESAG	group.vm, 110
distribution), 68	group.vm(Summary statistics for
esagda.pred (Prediction in	grouped circular data), 110
discriminant analysis based on	gi oupeu cii cutai uataj, 110
ESAG distribution), 79	Habeck's rotation matrix generation, 46
Long arou ibacion, 17	habean a rocation matrix generation, to

habeck.rot(Habeck's rotation matrix	(hyper-)spherical data), 76
generation), 46	hetcirc.boot (Bootstrap 2-sample mean
Haversine distance matrix, 47	test for circular data), 13
haversine.dist (Haversine distance	hetcirc.perm(Permutation based
matrix), 47	2-sample mean test for
hcf.aov, 12, 77, 114	circular data),78
hcf.aov(Anova for (hyper-)spherical	Hypothesis test for IAG distribution
data), 8	over the ESAG distribution, 48
hcf.boot, 9, 77	Hypothesis test for von Mises-Fisher
hcf.boot(Bootstrap 2-sample mean test	distribution over Kent
for (hyper-)spherical data),11	distribution, 49
hcf.circaov, 6, 14, 79, 110, 115	
hcf.circaov(Anova for circular data),9	iag.mle, 49, 60, 69, 101
hcf.perm, <i>12</i>	iag.mle(MLE of (hyper-)spherical
hcf.perm(Permutation based 2-sample	distributions), 62
mean test for	<pre>iag.reg(Spherical regression using the</pre>
(hyper-)spherical data), 76	projected normal or the von
hcfcirc.boot, 10	Mises-Fisher distribution), 105
hcfcirc.boot(Bootstrap 2-sample mean	<pre>iagd(Density of some (hyper-)spherical</pre>
test for circular data), 13	distributions), 35
hcfcirc.perm(Permutation based	iagesag, 50
2-sample mean test for	iagesag(Hypothesis test for IAG
circular data),78	distribution over the ESAG
hclr.aov, <i>10</i>	distribution), 48
hclr.aov(Anova for (hyper-)spherical data), 8	Interactive 3D plot of spherical data, 51
hclr.boot(Bootstrap 2-sample mean test	Inverse of Lambert's equal area
for (hyper-)spherical data), 11	projection, 52
hclr.circaov, 9	Inverse of the Euclidean
hclr.circaov(Anova for circular data),	transformation, 53
9	
hclr.perm(Permutation based 2-sample	k-NN algorithm using the arc cosinus
mean test for	distance, 54
(hyper-)spherical data), 76	k-NN regression, 55
hclrcirc.boot (Bootstrap 2-sample mean	kent.contour, 25, 27
test for circular data), 13	kent.contour (Contour plot of the Kent
hclrcirc.perm (Permutation based	distribution without any
2-sample mean test for	data), 25
circular data), 78	kent.datacontour, 24, 26
het.aov, 14, 79, 114	kent.datacontour(Contour plot of the
het.aov(Anova for (hyper-)spherical	Kent distribution for some
data), 8	data), 24
het.boot (Bootstrap 2-sample mean test	kent.logcon, 92
for (hyper-)spherical data), 11	kent.logcon(Logarithm of the Kent
het.circaov, 6, 115	distribution normalizing
het.circaov (Anova for circular data), 9	constant), 58
het.perm(Permutation based 2-sample mean test for	kent.mle, 25, 26, 36, 39, 49, 50, 58, 69, 70, 74, 87, 92, 104, 112

kent.mle (MLE of the Kent	meandir.test (Test for a given mean
distribution), 69	direction), 112
kmeans, 10, 61	mediandir, 76
knn.reg, 32, 80, 81, 122	mediandir(Spherical and
knn.reg(k-NN regression), 55	hyperspherical median), 104
knnreg.tune, 56	mediandir_2(Spherical and
knnreg.tune (Tuning of the k-NN	hyperspherical median), 104
regression), 121	Mixtures of Von Mises-Fisher
kuiper, 22	distributions, 61
kuiper(Uniformity test for circular	mixvmf.contour, 11, 62
data), 123	mixvmf.contour(Contour plot of a
	mixture of von Mises-Fisher
lambert, 40, 51-53, 91	distributions model), 22
lambert (Lambert's equal area	mixvmf.mle, 11, 23, 34, 55, 81, 97, 121
projection), 57	mixvmf.mle(Mixtures of Von
Lambert's equal area projection, 57	Mises-Fisher distributions), 61
lambert.inv, <i>57</i> , <i>91</i>	MLE of (hyper-)spherical
<pre>lambert.inv(Inverse of Lambert's</pre>	distributions, 62
equal area projection), 52	MLE of some circular distributions, 65
Logarithm of the Kent distribution	MLE of some circular distributions
normalizing constant, 58	with multiple samples, 67
lr.aov, <i>114</i>	MLE of the ESAG distribution, 68
<pre>lr.aov(Anova for (hyper-)spherical</pre>	MLE of the Kent distribution, 69
data), 8	MLE of the Matrix Fisher distribution
<pre>lr.boot(Bootstrap 2-sample mean test</pre>	on SO(3), 71
for (hyper-)spherical data), 11	MLE of the Purkayashta distribution, 72
lr.circaov, 6, 115	MLE of the Wood bimodal distribution
lr.circaov (Anova for circular data), 9	on the sphere, 73
<pre>lr.perm(Permutation based 2-sample</pre>	multispml.mle (MLE of some circular
mean test for	distributions with multiple
(hyper-)spherical data), 76	samples), 67
<pre>lrcirc.boot (Bootstrap 2-sample mean</pre>	multivm.mle (MLE of some circular
test for circular data), 13	distributions with multiple
lrcirc.perm (Permutation based	samples), 67
2-sample mean test for	multivmf.mle (MLE of (hyper-)spherical
circular data), 78	distributions), 62
32. 32. 32. 32. 32. 32. 32. 32. 32. 32.	d13t1 15dt10f13/, 02
makefolds, 32	Naive Bayes classifiers for circular
makefolds (Generate random folds for	data, 74
cross-validation), 43	Normalised spatial median for
Many simple circular or angular	directional data, 75
regressions, 59	north.america (Maps of the world and
Maps of the world and the continents,	the continents), 60
60	nsmedian, 104
matrixfisher.mle, 95	nsmedian (Normalised spatial median
matrixfisher.mle (MLE of the Matrix	for directional data), 75
Fisher distribution on SO(3)),	ioi directional data), 13
71	oceania (Maps of the world and the
meandir.test, 86	continents), 60
· · · ,	

Permutation based 2-sample mean test for (hyper-)spherical data, 76	resag(Simulation of random values from the ESAG distribution), 103
Permutation based 2-sample mean test	rfb, 5, 45, 92, 94, 96, 99, 101
for circular data, 78	rfb (Simulation of random values from
Prediction in discriminant analysis	a spherical Fisher-Bingham
based on ESAG distribution, 79	distribution), 97
Prediction in discriminant analysis	riag(Simulation of random values from
based on von Mises-Fisher	rotationally symmetric
distribution, 80	distributions), 100
Prediction with some naive Bayes	rkent, 36, 39, 50, 94, 96, 98
classifiers for circular data,	rkent (Simulation of random values
82	from a spherical Kent
Probability density function of the	distribution), 99
von Mises-Fisher distribution,	rmatrixfisher, 71
83	rmatrixfisher (Simulation from a
purka.mle, 66, 68	Matrix Fisher distribution on
purka.mle(MLE of the Purkayashta	SO(3)), 94
distribution), 72	rmixvmf, 11, 62, 101
pvm, 46	<pre>rmixvmf(Simulation of random values</pre>
<pre>pvm(Probability density function of</pre>	from a mixture of von
the von Mises-Fisher	Mises-Fisher distributions), 96
distribution), 83	rot.matrix, 30, 84, 88, 91
	rot.matrix(Rotation matrix from a
quat2rot, 29	rotation axis and angle of
quat2rot(Converting an unsigned unit	rotation), 88
quaternion to rotation matrix	rot2eul, <i>90</i>
on SO(3)), 30	rot2eul (Euler angles from a rotation matrix on SO(3)), 40
racg, 64, 101, 102	rot2quat, <i>30</i>
racg (Angular central Gaussian random	rot2quat(Converting a rotation matrix
values simulation), 7	on SO(3) to an unsigned unit
Random sample of matrices in SO(p), 84	quaternion), 29
rayleigh, 112, 124	rotation, 29, 30, 84, 88, 89
rayleigh (Rayleigh's test of	rotation (Rotation matrix to rotate a
uniformity), 85	spherical vector along the
Rayleigh's test of uniformity, 85	direction of another), 90
rbingham, 45, 92, 96, 98, 99	Rotation axis and angle of rotation
rbingham (Simulation from a Bingham	given a rotation matrix, 87
distribution using any	Rotation matrix from a rotation axis
symmetric matrix A), 93	and angle of rotation, 88
rcircbeta (Simulation of random values	Rotation matrix on SO(3) from three
from some circular	Euler angles, 89
distributions), 101	Rotation matrix to rotate a spherical
Read a file as a Filebacked Big	vector along the direction of
Matrix, 86	another, 90
read.fbm(Read a file as a Filebacked	rsop, 88, 89, 91
Big Matrix), 86	rsop (Random sample of matrices in
resag, 69	SO(p)), 84

rspml(Simulation of random values	<pre>spher.reg(Spherical-spherical</pre>
from some circular	regression), 107
distributions), 101	spherconc.test, <i>9</i> , <i>12</i> , <i>77</i>
rvmf, 7, 27, 45, 64, 66, 94, 96–99, 102	spherconc.test(Test for equality of
<pre>rvmf(Simulation of random values from</pre>	concentration parameters for
rotationally symmetric	spherical data), 113
distributions), 100	sphereplot, 27, 61, 69, 70, 74, 108, 114
rvonmises, 7, 38, 46, 66, 83, 101, 110, 111,	sphereplot (Interactive 3D plot of
124	spherical data), 51
rvonmises (Simulation of random values	Spherical and hyperspherical median,
from some circular	104
distributions), 101	Spherical regression using the
rwrapcauchy (Simulation of random	projected normal or the von
values from some circular	Mises-Fisher distribution, 105
distributions), 101	Spherical-spherical correlation, 106
<i>,</i> ,	Spherical-spherical regression, 107
Saddlepoint approximations of the	spml.fbed (Forward Backward Early
Fisher-Bingham distributions,	Dropping selection for
92	circular data using the SPML
Simulation from a Bingham distribution	regression), 42
using any symmetric matrix A,	spml.mle, 21, 43, 60, 103, 110
93	spml.mle(MLE of some circular
Simulation from a Matrix Fisher	distributions), 65
distribution on SO(3),94	spml.nb (Naive Bayes classifiers for
Simulation of random values from a	circular data), 74
Bingham distribution, 95	spml.reg, 5, 16, 17, 19, 21, 43, 56, 60, 106,
Simulation of random values from a	108
mixture of von Mises-Fisher	spml.reg(Circular or angular
distributions, 96	regression), 17
Simulation of random values from a	spml.regs, 43
spherical Fisher-Bingham	-
distribution, 97	spml.regs (Many simple circular or
Simulation of random values from a	angular regressions), 59
spherical Kent distribution, 99	spmlnb.pred (Prediction with some
Simulation of random values from	naive Bayes classifiers for
rotationally symmetric	circular data), 82
distributions, 100	Summary statistics for circular data,
Simulation of random values from some	109
circular distributions, 101	Summary statistics for grouped
Simulation of random values from the	circular data,110
ESAG distribution, 103	
south.america(Maps of the world and	tang.conc(A test for testing the
the continents), 60	equality of the concentration
spcauchy.mle(MLE of (hyper-)spherical	parameters for ciruclar data),
distributions), 62	5
spher.cor, <i>18</i> , <i>108</i>	Test for a given mean direction, 112
spher.cor(Spherical-spherical	Test for equality of concentration
correlation), 106	parameters for spherical data,
spher.reg, 18, 56, 107, 122	113

Test of equality of the concentration vmf.mle (MLE of (hyper-)spherical		Test of equality of the concentration	<pre>vmf.mle(MLE of (hyper-)spherical</pre>
	rest of equality of the concentration	parameters for circular data,	distributions), 62
parameters for circular data, distributions), 62		114	<pre>vmf.reg(Spherical regression using the</pre>
•	parameters for circular data, distributions), 62	The k-nearest neighbours using the	projected normal or the von
vmf.reg(Spherical regression using the	parameters for circular data, distributions), 62 114 vmf.reg(Spherical regression using the	cosinus distance, 115	Mises-Fisher distribution), 105
$\frac{114}{\text{The k-nearest neighbours using the}} \hspace{0.5cm} \text{vmf.reg (Spherical regression using the} \\ \hspace{0.5cm} \text{projected normal or the von} \\$	parameters for circular data, distributions), 62 114 $vmf.reg (Spherical regression using the$ The k-nearest neighbours using the projected normal or the von	Transform unit vectors to angular	vmfda.pred, 32, 34, 35, 55, 80
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	parameters for circular data, distributions), 62 114 1	data, 116	vmfda.pred(Prediction in discriminant
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	parameters for circular data, $\frac{114}{114}$ vmf.reg(Spherical regression using the $\frac{114}{115}$ cosinus distance, $\frac{115}{115}$ mises-Fisher distribution), $\frac{105}{115}$ Transform unit vectors to angular vmfda.pred, $\frac{32}{115}$, $\frac{34}{115}$, $\frac{35}{115}$, $\frac{80}{115}$	Tuning of the bandwidth parameter in	analysis based on von
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	parameters for circular data, 114 vmf.reg(Spherical regression using the cosinus distance, 115 vmfda.pred, 32 , 34 , 35 , 55 , 80 vmfda.pred(Prediction in discriminant	the von Mises kernel, 117	Mises-Fisher distribution), 80
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	parameters for circular data, $\frac{114}{114}$ uning of the bandwidth parameter in distributions), 62 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred, 32, 34, 35, 55, 80 vmfda.pred (Prediction in discriminant analysis based on von	Tuning of the bandwidth parameter in	vmfkde.tune, 23, 24, 118, 125, 126
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the von Mises kernel, 117 Tuning of the von Mises kernel, 117 Tuning of the von Mises kernel, 117 Vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 Vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80	parameters for circular data, 114 vmf.reg(Spherical regression using the rosinus distance, 115 vmfda.pred, 32 , 34 , 35 , 55 , 80 vmfda.pred(Prediction in discriminant analysis based on von the von Mises-Fisher distribution), 80	the von Mises-Fisher kernel,	vmfkde.tune(Tuning of the bandwidth
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, vmfkde.tune, 23, 24, 118, 125, 126 Tomposite the von mises with the von Mises-Fisher kernel, vmfkde.tune (Tuning of the bandwidth)	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, Tuning of the bandwidth parameter in the von Mises-Fisher kernel, Tuning of the bandwidth parameter in the von Mises-Fisher kernel, Tuning of the bandwidth parameter in the von Mises-Fisher kernel, Tuning of the bandwidth parameter in the von Mises-Fisher kernel, Tuning of the bandwidth	118	
Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 107 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune (Tuning of the bandwidth	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, Tuning of the bandwidth parameter in the von Mises-Fisher kernel, Tuning of the bandwidth parameter in the von Mises-Fisher kernel, Tuning of the bandwidth parameter in the von Mises-Fisher kernel, Tuning of the bandwidth parameter in the von Mises-Fisher kernel, Tuning of the bandwidth	Tuning of the k-NN algorithm using the	Mises-Fisher kernel), 118
Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 107 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 distributions), 62 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune (Tuning of the bandwidth parameter in the von	arc cosinus distance, 119	vmkde.tune, <i>119</i> , <i>125</i> , <i>126</i>
Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 107 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-nearest neighbours using the projected normal or the von Mises-Fisher distribution), 105 Tomfide.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119	Tuning of the k-NN regression, 121	vmkde.tune(Tuning of the bandwidth
Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 107 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-nearest neighbours using the projected normal or the von Mises-Fisher distribution), 105 Tomic (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 Tuning of the bandwidth parameter in the von the von Mises-Fisher kernel, 117 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the the projected normal or the von Mises-Fisher distribution), 105 Tuning of the bandwidth parameter in the von the bandwidth parameter in	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119		parameter in the von Mises
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121	Uniformity test for circular data, 123	kernel), 117
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121		vmnb.pred, <i>75</i> , <i>117</i>
Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmf.da.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmf.de.tune, 23, 24, 118, 125, 126 vmfkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred, 75, 117	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression using the projected normal or the von Mises-Fisher distribution), 105 Tuning of the bandwidth parameter in analysis based on von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in wifkde. tune, 23, 24, 118, 125, 126 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 105 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 105 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in analysis based on von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in analysis based on von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in analysis based on von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in analysis based on von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in analysis based on von Mises-Fisher distributio		vmnb.pred(Prediction with some naive
Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred, 75, 117 vmnb.pred (Prediction with some naive	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 parameters for circular data, 123 vmf.reg(Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred (Prediction with some naive		Bayes classifiers for circular
vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 Vmfkde.tune, 23, 24, 118, 125, 126 Vmfkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 Vmkde.tune, 119, 125, 126 Vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 Vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 Vmhb.pred, 75, 117 Vmnb.pred (Prediction with some naive Bayes classifiers for circular	parameters for circular data, 114 vmfb.ered (Prediction with some naive Bayes classifiers for circular data, 115 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel, 117 vmfb.pred, 75, 117 vmnb.pred (Prediction with some naive Bayes classifiers for circular	•	
Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression using the projected normal or the von Mises-Fisher distribution), 105 Tuning of the bandwidth parameter in mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 Tuning of the k-NN regression, 121 Tuning of the bandwidth parameter in the von Mises kernel), 117 Tuning of the bandwidth parameter in the von Mises kernel), 117 Tuning of the bandwidth parameter in the von Mises kernel), 117 Tuning of the bandwidth parameter in the von Mises kernel, 119, 125, 126 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises kernel, 118 Tuning of the bandwidth parameter in the von Mises kernel, 118 Tuning of the bandwidth parameter in the von Mises kernel, 118 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution, 105 Tuning of the bandwidth parameter in the von Mises-Fisher distribution, 105 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the bandwidth parameter in th	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression using the projected normal or the von Mises-Fisher distribution), 105 Tuning of the bandwidth parameter in analysis based on von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 105 Transform unit vectors, 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution, 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution, 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution, 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution, 80 Tuning of the bandwidth parameter in the von Mises-Fisher distribution, 80 Tuning of the bandwidth parameter in the von Mises-Fisher	·	von Mises kernel density estimation,
Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression using the projected normal or the von Mises-Fisher distribution), 105 Tuning of the bandwidth parameter in analysis based on von Mises-Fisher distribution), 80 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression using the projected normal or the von Mises-Fisher distribution), 105 Transform unit vectors to angular Mises-Fisher distribution), 105 Tuning of the bandwidth parameter in Mises-Fisher distribution), 80 Tuning of the k-NN algorithm using the Auton (Tuning of the bandwidth parameter in the von Mises kernel), 118 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the k-NN algorithm using the Auton Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher dernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher dernel, 118 Tuning of the k-NN algorithm using the Auton Mises-Fisher distribution), 80 Tuning of the bandwidth parameter in the von Mises-Fisher dernel, 119 Tuning of the k-NN algorithm using the Auton Mises-Fisher distribution, 80 Tuning of the bandwidth parameter in the von Mises-Fisher dernel, 117 Tuning	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression using the projected normal or the von Mises-Fisher distribution), 105 Tuning of the bandwidth parameter in analysis based on von Mises-Fisher distribution), 80 Vmfkde.tune, 23, 24, 118, 125, 126 Vmfkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 Vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 Vmnb.pred (Prediction with some naive Bayes classifiers for circular data), 82 Von Mises kernel density estimation,		124
vmf.reg (Spherical regression using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 105 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression using the projected normal or the von Mises-Fisher distribution), 105 Vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 Vmfkde.tune, 23, 24, 118, 125, 126 Vmfkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 Vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 Vmmb.pred, 75, 117 Vmmb.pred (Prediction with some naive Bayes classifiers for circular data), 82 Von Mises kernel density estimation, 124	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Vec (Generation of unit vector(s) with a given angle), 44 Visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 distributions), 62 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred (Prediction with some naive Bayes classifiers for circular data), 82 von Mises kernel density estimation, 124		
Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression iusing the projected normal or the von Mises-Fisher distribution), 105 Transform unit vectors to angular Mises-Fisher distribution), 105 Tuning of the bandwidth parameter in analysis based on von Mises-Fisher distribution), 105 Transform unit vectors to angular Mises-Fisher distribution), 105 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the k-NN reg	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 Vm. kde, 110, 111, 118, 119, 126 Vm. kde, 110, 111, 118, 119, 126 Vmf. reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 Vmfda. pred, 32, 34, 35, 55, 80 Vmfda. pred, 97ediction in discriminant analysis based on von Mises-Fisher distribution), 80 Vmfde. tune, 23, 24, 118, 125, 126 Vmfkde. tune, 23, 24, 118, 125, 126 Vmfkde. tune (Tuning of the bandwidth parameter in the von Mises kernel), 118 Vmmb. pred, 75, 117 Vmnb. pred (Prediction with some naive Bayes classifiers for circular data), 82 Von Mises-Fisher distribution), 105 Vmfda. pred, 32, 34, 35, 55, 80 Vmfda. pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 105 Vmfda. pred (Prediction in discriminant analysis based on von Mises-Fisher kernel), 118 Vmfkde. tune, 23, 24, 118, 125, 126 Vmfkde. tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 Vmnb. pred (Prediction with some naive Bayes classifiers for circular data), 82 Von Mises-Fisher kernel density estimation, 124 Von Mises-Fisher kernel density		
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde (von Mises kernel density vm.kde (von Mises kernel density vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred (Prediction with some naive Bayes classifiers for circular data), 82 von Mises-Fisher kernel density estimation, 124 von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminat analysis based on von Mises-Fisher distribution), 105 vmfda.pred (Prediction with some naive Bayes classifiers for circular data), 82 von Mises-Fisher kernel density	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 vec (Generation of unit vector(s) with a given angle), 44 visual check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.kde (von Mises kernel density distributions), 62 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred, 32, 34, 35, 55, 80 vmfda.pred, (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune, (Tuning of the bandwidth parameter in the von Mises kernel), 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel, 117 vmnb.pred, 75, 117 vmfkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred, 75, 117 vmfkde.tune, 23, 24, 118, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred, 75, 117 vmfkde.tune, 13, 125, 126 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel, 117 vmnb.pred, 75, 117 vmnb.pred, 75, 117 vmnb.pred, 75, 100 vmfkde.tune, 23, 24, 118, 125, 126 vmkde.tune, 119, 125, 126 vmkde.tune, 1		(hyper-)spherical data, 125
vmf.reg (Spherical regression using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.kde (von Mises kernel density estimation), 124 vm.kde (von Mises kernel density estimation), 124 vm.kde (von Mises kernel density estimation for (hyper-)spherical data, 125	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 vec (Generation of unit vector(s) with a given angle), 44 vm.kde, 110, 111, 118, 119, 126 vm.kde (von Mises kernel density estimation), 124 matrix Fisher samples is correctly generated or not), 14 vm.kde (von Mises kernel density estimation), 124 distributions), 62 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfdde.tune, 23, 24, 118, 125, 126 vmfkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred (Prediction with some naive Bayes classifiers for circular data), 82 von Mises-Fisher distribution), 80 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel), 118 vmfkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel), 110 vmfkde.tune, 119, 125, 126 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfkde.tune, 23, 24, 118, 125, 126 vmkde.tune, 119, 125, 126 v		. 22
Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde (von Mises kernel density estimation), 124 vm.mle, 21, 64 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfdda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfdde.tune, 23, 24, 118, 125, 126 vmfkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred, 75, 117 vmnb.pred (Prediction with some naive Bayes classifiers for circular data), 82 von Mises kernel density estimation, 124 von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel), 118 vmfkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred, 75, 117	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression using the projected normal or the von Mises-Fisher distribution), 105 Transform unit vectors to angular vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 Vmfdde.tune, 23, 24, 118, 125, 126 Vmfkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 Vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 Vmnb.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel), 118 Vmfkde.tune, 23, 24, 118, 125, 126 Vmfkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 Vmnb.pred, 75, 117 Vmnb.pred (Prediction with some naive Bayes classifiers for circular data), 82 Von Mises kernel density estimation, 124 Von Mises-Fisher kernel density estimation for (hyper-)spherical data, 125		
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the k-NN regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred, 32, 34, 35, 55, 80 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune, (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmmb.pred, 75, 117 vmmb.pred, 75, 117 vmmb.pred (Prediction with some naive Bayes classifiers for circular data, 123 vmmb.pred, 75, 117 vmmb.pred (Prediction with some naive Bayes classifiers for circular data, 123 vmkde.tune, 119, 125, 126 vmkde.tune, 119,	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.kde (von Mises kernel density estimation), 124 vm.mle, 21, 64 vm.nb, 82 vmfda.pred (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred, 32, 34, 35, 55, 80 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfd.pred, 72, 41, 18, 125, 126 vmfkde.tune, 23, 24, 118, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred, 75, 117 vmnb.pred (Prediction with some naive Bayes classifiers for circular data), 82 von Mises kernel density estimation, 124 von Mises-Fisher kernel density estimation for (hyper-)spherical data, 125 watson, 22		
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 Tuning of the k-NN algorithm using the analysis based on von Mises-Fisher distribution), 80 vmfda.pred, 32, 34, 35, 55, 80 vmfdde.tune, 23, 24, 118, 125, 126 vmfkde.tune, (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmmb.pred, 75, 117 vmmb.pred, 75, 117 vmmb.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune, (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmmb.pred, 75, 117 vmmb.pred, 75, 117 vmmb.pred, 75, 117 vmmb.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune,	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.kde (von Mises kernel density estimation), 124 vm.mle, 21, 64 vm.nb, 82 vm.nb (Naive Bayes classifiers for		
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 105 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the k-NN pregression using the projected normal or the von Mises-Fisher distribution), 105 Vmfda. pred, 32, 34, 35, 55, 80 Vmfdde. tune, 23, 24, 118, 125, 126 Vmfkde. tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 Vmkde. tune, 119, 125, 126 Vmkde. tune, 119, 125, 126 Vmkde. tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 Vmnb. pred, 75, 117 Vmnb. pred, 72, 117 Vmnb. pred, Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 Vmfda. pred, 32, 34, 35, 55, 80 Vmfkde. tune, 23, 24, 118, 125, 126 Vmkde. tune (Tuning of the bandwidth parameter in the von Mises kernel), 118 Vmkde. tune, 119, 125, 126 Vmkde. tune, 219, 125, 126 Vmkde. tune, 219, 125, 126 Vmkde. tune, 219, 125, 126 Vmkde. tune, 119, 125, 126 Vmkde. tune, 219, 125, 126 Vmkde. tune, 219, 125, 126 Vmkde. tune, 219, 125, 126 Vmkde. tune, 119, 125, 126 Vmkde. tune, 129, 125, 126 Vmkde. tune, 119, 125, 126 Vmkde. tune, 119, 125, 126 Vmkde. tune, 129, 125, 126 Vmkde. tune, 129, 125,	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.kde (von Mises kernel density estimation), 124 vm.nb, 82 vm.nb, 82 vm.nb (Naive Bayes classifiers for circular data), 74 distributions), 62 vmf.cag(Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmf.da.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda.pred, 32, 34, 35, 55, 80 vmfda.pred, (Prediction in discriminant analysis based on von Mises-Fisher distribution), 105 vmfda.pred, 32, 34, 35, 55, 80 vmfde.tune, (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune, (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmmb.pred, 75, 117 vmmb.pred, 75, 117 vmmb.pred, 75, 117 vmmb.pred, 72, 126 vmkde.tune, (Tuning of the bandwidth parameter in the von Mises kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tun		
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.mle, 21, 64 vm.nb, 82 vm.nb (Naive Bayes classifiers for circular data), 74 vmf.contour, 23, 24, 26, 51 vmf.cetorour data in the von Mises fisher distribution), 105 vmfda.pred, 32, 34, 35, 55, 80 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfdde.tune, 23, 24, 118, 125, 126 vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune, (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred, 72, 317 vmfc.pred, 72, 34, 35, 55, 80 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda.pred, 72, 34, 35, 55, 80 vmfdde.tune, 23, 24, 118, 125, 126 vmfkde.tune, 23, 24, 118, 125, 126 vmkde.tune, (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred, 75, 117 vmnb.pred (Prediction with some naive Bayes classifiers for circular data), 82 von Mises kernel density estimation, 124 von Mises-Fisher distribution), 105 vmfda.pred, 72, 316 vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune, 23, 24, 18, 125, 126 vmfkde.tune, 23, 24, 18, 125, 126 vmfkde.tune, 23, 24, 18, 125, 126 vmfkde.tun	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.kde, 110, 111, 118, 119, 126 vm.kde, 110, 111, 118, 119, 126 vm.kde (von Mises kernel density estimation), 124 vm.nle, 21, 64 vm.nle, 22, 24, 26, 51 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred, 123, 34, 35, 55, 80 vmfda.pred, 122, 18, 125, 126 vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmmb.pred, 75, 117 vmnb.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel), 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmmb.pred, 75, 117 vmmb.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel), 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmmb.pred, 75, 117 vmmb.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel, 119 vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth param		
vmf.reg(Spherical regression using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Vec (Generation of unit vector(s) with a given angle), 44 visual. check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.kde, 110, 111, 118, 119, 126 vm.kde (von Mises kernel density estimation), 124 vm.kde, 121, 64 vm.nb, 82 vm.nb (Naive Bayes classifiers for circular data), 74 vmf.contour, 23, 24, 26, 51 vmf.contour (Contour plots of the von	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.kde, 110, 111, 118, 119, 126 vm.kde, 110, 111, 118, 119, 126 vm.kde, 121, 64 vm.nb, 82 vm.nb, 82 vm.nb (Naive Bayes classifiers for circular data), 74 vmf.contour, 23, 24, 26, 51 vmf.contour (Contour plots of the von distributions), 62 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred, 32, 34, 35, 55, 80 vmfda.pred, 92, 32, 34, 18, 125, 126 vmfkde.tune, 119, 125, 126 vmkde.tune, 179, 125, 126 vmkde.		
vmf.reg(Spherical regression using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.kde, 110, 111, 118, 119, 126 vm.kde (von Mises kernel density estimation, 124 vm.kde, 110, 111, 118, 119, 126 vm.mle, 21, 64 vm.nb, 82 vm.nb (Naive Bayes classifiers for circular data), 74 vmf.contour, 23, 24, 26, 51 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.kde (von Mises kernel density estimation), 124 vm.mle, 21, 64 vm.nb, 82 vm.nb (Naive Bayes classifiers for circular data), 74 vmf.contour, 23, 24, 26, 51 vmf.contour, 23, 24, 26, 51 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf.contour, 23, 24, 26, 51 vmf.contour, 23, 24, 26, 51 vmf.contour glots of the von Mises-Fisher distribution), 27 vmf.contour glota normal or the von Mises-Fisher distribution), 105 vmf.da.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda.pred, 32, 34, 35, 55, 80 vmfda.pred, (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfde.tune, 23, 24, 118, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 60 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda.pred (Prediction with some nature Bayes classifiers for circular data), 82 von Mises-Fisher k		,
vmf.reg (Spherical regression using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 vmiles-Fisher distribution), 80 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfde.tune, 23, 24, 118, 125, 126 vmkde.tune, [70, 110, 118, 118, 119, 126 vmkde.tune, [119, 125, 126 vmkde.tune, [119, 125, 126 vmkde.tune, [119, 125, 126 vmkde.tune, [70, 111, 118, 119, 126 vmfkde.tune, [70, 111, 118, 118, 121, 126 vmfkde.tune, [70, 111, 118, 118, 128, 125, 126 vmfkde.tune, [70, 111, 118, 118, 126 vmfkde.tune, [70, 118, 118, 125, 126 vmfkde.tune, [70, 118, 118, 118, 125, 126 vmfkde.tune, [70, 118, 118, 128, 125, 126 vmfkde.tune, [70, 118, 118, 118, 125, 126 vmfkde.tune, [70, 118, 118, 118, 125, 126 vmfkde.tune,	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the wises-Fisher kernel, 118 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the wmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel, 120 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel, 118 vmfde.tune, 23, 24, 118, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel, 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 vmkde.tune, 132, 24, 118, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mi		
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the k-NN regression using the marker in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 vmfkde. tune, 23, 24, 118, 125, 126 vmfkde. tune, (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmc (Generation of unit vector(s) with a given angle), 44 visual. check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. mkde (von Mises kernel density estimation), 124 vm. mle, 21, 64 vm. nb, 82 vm. nb (Naive Bayes classifiers for circular data), 74 vmf. contour, 23, 24, 26, 51 vmf. contour, 23, 24, 26, 51 vmf. da, 35, 81, 121 vmf. da (Cross validation in von vmfda. pred (Prediction in discrimiant analysis based on von Mises-Fisher distribution), 105 vmfda. pred (Prediction in discrimiant analysis based on von Mises-Fisher distribution), 80 vmfkde. tune, (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde. tune, 719, 125, 126 vmkde. tune, (Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 vmfkde. tune, (Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 vmfkde. tune, (Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 vmfda. pred (Prediction with some naive Bayes classifiers for circular data), 82 von Mises-Fisher density estimation, 124 von Mises-Fisher density estimation, 124 von Mises-Fisher density estimation, 124 von Mises-Fisher density with some naive Bayes classifiers for circular data), 82 von Mises-Fisher density estimation, 124 von Mises-Fisher density with some naiv	matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.kde, 110, 111, 118, 119, 126 vm.nle, 21, 64 vm.nle, 23, 24, 26, 51 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf.da, 275, 211 vmf.da, 176 (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmf.ca, 33, 81, 121 vmf.da (Cross validation in von distributions), 62 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmf.seg, (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmf.da, 23, 34, 35, 55, 80 vmfda.pred, (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda.pred, (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda.pred, (Prediction in discriminant analysis based on von Mises-Fisher distribution), 105 vmfda.pred, (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda.pred, (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfde.tune, 23, 24, 118, 125, 126 vmkde.tune, (Tuning of the bandwidth parameter in the von Mises kernel, 117 vmnb.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda.pred, (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda.pred, (Prediction in discriminat analysis based on von Mises-Fisher distribution), 80 vmfde.tune, 23, 24, 118, 125, 126 vmkde.tune, (Tuning of the bandwidth parameter in the von Mises-Fisher derenel), 118 vmkde.tune, 27, 27, 117 vmnb.pred (Prediction with some naive Bayes classifiers for circular data), 82 von Mises-Fisher kernel density estimation, 124 von Mises-Fisher kernel density estimation, 124 von Mises-Fisher kerne		• • •
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual. check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. kde (von Mises kernel density estimation), 124 vm. mle, 21, 64 vm. nb, 82 vm. nb (Naive Bayes classifiers for circular data), 74 vmf. contour (Contour plots of the von Mises-Fisher distribution), 27 vmf. da, 35, 81, 121 vmf. da (Cross validation in von Mises-Fisher discrminant vmf. da (Cross validation in von Mises-Fisher discrminant vmf. da (Cross validation in von Mises-Fisher discrminant vmf. da (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda. pred, 32, 34, 35, 55, 80 vmfda. pred, 32, 44, 118, 125, 126 vmfdde. tune, 119, 125, 126 vmfde. tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde. tune, 119, 125, 126 vmkde. tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde. tune, 119, 125, 126 vmkde. tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde. tune, 119, 125, 126 vmkde. tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde. tune, 119, 125, 126 vmkde. tune (Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 124 vmkde. tune, 119, 125, 126 vmkde. tune (Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 25 vmfda. pred (Prediction in discriminant	distributions), 62 reflection minimum to the von mises fisher agiven angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.cde, 110, 111, 118, 119, 126 vm.cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.mle, 21, 64 vm.mle, 21,		
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. kde (von Mises kernel density estimation), 124 vm. nb, 82 vm. nb, (Naive Bayes classifiers for circular data), 74 vmf. contour, 23, 24, 26, 51 vmf. da, 35, 81, 121 vmf. da (Cross validation in von Mises-Fisher discrminant analysis), 33 vmf. reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmf. nel von Mises-Fisher distribution), 105 vmfda. pred (Prediction in discriminant analysis based on von Mises-Fisher kernel), 118 vmfda. pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda. pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfkde. tune, (Tuning of the bandwidth parameter in the von Mises kernel), 118 vmkde. tune, (19, 125, 126 vmkde. tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb. pred, 75, 117	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression using the projected normal or the von Mises-Fisher distribution), 105 Vmfda.pred, 32, 34, 35, 55, 80 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel), 118 vmfkde.tune, 23, 24, 118, 125, 126 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred, 75, 117 vmnb.pred, 75, 117 vmnb.pred, 79 distribution), 80 vmfkde.tune, 119, 125, 126 vmkde.tune, 119, 125, 126 vmfkde.tune, 119, 125		
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 Uniformity test for circular data, 123 Vec (Generation of unit vector(s) with a given angle), 44 visual-check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. nb, 82 vm. nb (Naive Bayes classifiers for circular data), 74 vmf.contour, 23, 24, 26, 51 vm. nb, 82 vm. nb (Naive Bayes classifiers for circular data), 74 vmf.contour, 23, 24, 26, 51 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf. da, 35, 81, 121 vmf.da (Cross validation in von Mises-Fisher distribution), 105 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel, 117 vmfkde.tune, (Tuning of the bandwidth parameter in the von Mises kernel, 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel, 117 vmnb.pred, 75, 117 vmnb.pred, 75, 117 vmnb.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfkde.tune, (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel, 117 vmnb.pred, 75, 117 vmnb.pred (Prediction with some naive Bayes classifiers for circular data), 82 von Mises-Fisher kernel, 119 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel, 118 vmkde.tune, 119, 125, 126 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune, 119, 125, 126 vmfloa (Tuning of	maters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. kde (von Mises kernel density estimation), 124 vm. mle, 21, 64 vm.nb, 82 vm.nb (Naive Bayes classifiers for circular data), 74 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf.da, 23, 24, 18, 121 vmf.da (Cross validation in von Mises-Fisher discrminant analysis), 33 vmf.kde, 24, 118, 119, 125 vmf.kde, 24, 118, 119, 125 vmf.da (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda.pred, 32, 34, 35, 55, 80 vmfda.pred, 32, 34, 18, 125, 126 vmfda.pred, 32, 34, 18, 125, 126 vmfda.pred, 32, 34, 18, 125, 126 vmfdae, red, 32, 41, 18, 125, 126 vmfdee.tune, 119, 125, 126 vmkde.tune, 33, 34, 18, 125, 126 vmkde.tune, 33, 34, 18, 125, 126 vmkde.tune, 119, 125, 126 vmkde.tune, 31, 124 vmkde.tune, 119, 125, 126 vmkde.tune, 130, 144 vmkde.tune, 119, 125, 126 vmkde.tune, 119, 125, 126 vmkde.tune, 130, 144 vmkde.tune, 140, 125, 126 vmkde.tune, 140, 125, 126 vmkde.tune, 140, 125, 126 vmfda.pred, 32, 34, 18,		distributions), 63
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Vec (Generation of unit vector(s) with a given angle), 44 Visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.kde (von Mises kernel density estimation), 124 vm.mle, 21, 64 vm.mle, 21, 64 vm.mle, 23, 24, 26, 51 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf.da, 35, 81, 121 vmf.da (Cross validation in von Mises-Fisher kernel) mises-Fisher distribution), 27 vmf.da, 53, 81, 121 vmf.de (Constour plots of the von Mises-Fisher distribution), 27 vmf.da, 25, 81, 121 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf.da, 25, 81, 121 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf.da, 25, 81, 121 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf.da, 27, 42, 26, 51 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf.da, 27, 42, 26, 51 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf.da, 27, 418, 119, 125 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf.da, 27, 418, 125, 126 vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune, 119, 125, 126 vmfkde.tune, 23, 24, 18, 125, 126 vmfkde.tune, 119, 125, 126 vmfkde.tune, 23, 24, 18, 125, 126 vmfkde.tune, 23, 24, 18, 125, 126 vmfkde.tune, 119, 125, 126 vmfkde.tune, 23, 24, 26, 21 vmfkde.tune, 119, 125, 126 vmfkde.tune, 10, 10, 10, 10, 10, 10, 10,	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. kde (von Mises kernel density estimation), 124 vm. nb, 82 vm. nb, 82 vm. nb, 82 vm. nb (Naive Bayes classifiers for circular data), 74 vmf. contour (Contour plots of the von Mises-Fisher distribution), 27 vmf. a, 35, 81, 121 vmf. da (Cross validation in von Mises-Fisher kernel) vmf. da (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmf.ag, (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmf.ag, (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmf.ag, (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmf.ag, (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmf.ag, 29, 41, 82, 32, 43, 35, 55, 80 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher kernel, 117 vmfkde. tune, 23, 24, 118, 125, 126 vmkde. tune, (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 117 vmb.pred, 75, 1/7 vmb.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfkde. tune, 23, 24, 118, 19, 125, 126 vmkde. tune, (Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 80 vmkde. tune, 23, 24, 18, 125 vmkde tune (Tuning of the von Mises kernel), 117 vmb.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmkde. tune, (Tuning of the von Mises kernel), 1		
vmf.reg (Spherical regression using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual. check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. kde (von Mises kernel density estimation), 124 vm. m.b (Naive Bayes classifiers for circular data), 74 vmf. contour, 23, 24, 26, 51 vmf. da, 35, 81, 121 vmf. da (Cross validation in von Mises-Fisher distribution), 27 vmf. kde, 24, 118, 119, 125 vmf. contour, 23, 24, 26, 51 vmf. contour, 23, 24, 26, 51 vmf. contour, 23, 24, 26, 51 vmf. kde, 24, 118, 119, 125 vmf. kde, 10 in discribution, 105 vmfda. pred (Prediction in discribution), 80 vmfkde. tune, 23, 24, 118, 125, 126 vmfkde. tune, (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde. tune, (Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 117 vmmb.pred (Prediction in discribution), 80 vmfkde. tune, 23, 24, 118, 125, 126 vmkde. tune, 119, 125, 126 vmkde. tune, 23, 24, 118, 125 von Mises-Fisher kernel, 117 vmb.p	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 Vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. nb (Naive Bayes classifiers for circular data), 74 vmm. nb (Naive Bayes classifiers for circular data), 74 vmf. contour, 23, 24, 26, 51 vmf. contour, 23, 24, 26, 51 vmf. contour (Contour plots of the von Mises-Fisher distribution), 27 vmf. da, 35, 81, 121 vmf. da (Cross validation in von Mises-Fisher discrminant analysis), 33 vmf. kde, 24, 118, 119, 125 vmf. kde (von Mises-Fisher kernel density estimation for		
vmf.reg(Spherical regression using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Vuniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. kde (von Mises kernel density estimation), 124 vm. mle, 21, 64 vm. nb, 82 vm.nb (Naive Bayes classifiers for circular data), 74 vmf.contour (Contour plots of the von Mises-Fisher distribution), 27 vmf.da, 35, 81, 121 vmf.da (Cross validation in von Mises-Fisher distribution), 23 vmf.kde, 24, 118, 119, 125 vmf.kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf.kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf.kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf.kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. kde (von Mises kernel density estimation), 124 vm. ml. e. 21, 64 vm. f. contour (Contour plots of the von Mises-Fisher distribution), 27 vmf. a. 35, 81, 121 vmf. contour (Contour plots of the von Mises-Fisher distribution), 27 vmf. a. 35, 81, 121 vmf. can distributions, 62 vmf. reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmf. a. 32, 34, 35, 55, 80 vmfkde. tune (Prediction in discriminat analysis based on von Mises-Fisher distribution), 80 vmfkde. tune (Tuning of the bandwidth parameter in the von Mises kernel), 118 vmkde. tune (Tuning of the bandwidth parameter in the von Mises kernel, 117 vmnb. pred, 75, 117 vmnb. pred, 75, 117 vmnb. pred (Prediction in discriminat analysis based on von Mises-Fisher distribution), 80 vmfkde. tune (Tuning of the bandwidth parameter in the von Mises kernel, 118 vmkde. tune (Tuning of the bandwidth parameter in the von Mises kernel, 117 vmnb. pred (Prediction with some naive Bayes classifiers for circular data), 82 von Mises kernel density estimation for (hyper-)spherical data, 125 watson, 22 watson (Uniformity test for circular data), 123 wet (Goneration of unit vector(s) with a given angle, 44 visual.check (Check visually whether acrosing the projection in discrimination, 102 vmkde. tune (Tuning of		
vmf.reg(Spherical regression using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 105 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.m.kde, (10) Mises kernel density estimation, 124 vm.m.la, 21, 64 vm.m.la, 22, 4, 26, 51 vmf.contour, 23, 24, 26, 51 vmf.da (Cross validation in von Mises-Fisher distribution), 27 vmf.da (Cross validation in von Mises-Fisher discrminant analysis), 33 vmf.kde, 24, 118, 119, 125 vmf.kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf.kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf.kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf.kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf.kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf.kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf.kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf.kde (vno Mises (Prediction in discriminant analysis based on von Mises-Fisher discriminat danalysis based on von Mises-Fisher dernel, 117 vmf.contouning fre de Prediction in discriminant analysis based	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. nb, 82 vm. nb, 82 vm. nb, 82 vm. nb (Naive Bayes classifiers for circular data), 74 vmf. contour, 23, 24, 25, 51 vmf. contour (Contour plots of the von Mises-Fisher distribution), 27 vmf. da, 27, 28, 29, 51 vmf. da (Cross validation in von Mises-Fisher discrminant analysis), 33 vmf. kde, 24, 118, 119, 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel) distribution), 105 vmfda. pred, 32, 34, 35, 55, 80 vmfda. pred, 32, 34, 31, 55, 58 vmfda. pred, 32, 34, 118, 125, 126 vmfkde. tune, 123, 24, 118, 125, 126 vmfkde. tune, 123, 24, 118, 119, 125 vmfkde. tune, 123, 24, 118, 119, 125 vmfkde. tune, 123, 24, 118, 119, 125 vmfkde. tune, 123, 24, 1		
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Vec (Generation of unit vector(s) with a given angle), 44 visual. check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. kde (von Mises kernel density estimation), 124 vm. m. b(R), 120 vm. nb (Naive Bayes classifiers for circular data), 74 vmf. contour (Contour plots of the von Mises-Fisher distribution), 27 vmf. da, 35, 81, 121 vmf. cortour (Contour plots of the von Mises-Fisher distribution), 27 vmf. da, 35, 81, 121 vmf. cortour, 23, 24, 26, 51 vmf. cortour (Contour plots of the von Mises-Fisher distribution), 27 vmf. da, 27, 28, 28, 29, 21, 119 vmf. cortour (Contour plots of the von Mises-Fisher distribution), 27 vmf. da, 27, 28, 28, 29, 28 vmfda. pred, 32, 34, 35, 55, 80 vmfda. pred, 32, 34, 35, 55, 80 vmfdda. pred, 32, 34, 18, 125, 126 vmfkde. tune, 119, 125, 126 vmfkde. tune, 129, 128, 125, 126 vmfkde. tune, 119, 125, 126 vmfkde. tune, 1	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. nb (Naive Bayes classifiers for circular data), 72 vmf. contour, 23, 24, 26, 51 vmf. contour, 23, 24, 26, 51 vmf. contour (Contour plots of the von Mises-Fisher distribution), 27 vmf. da, 35, 81, 121 vmf. da (Cross validation in von Mises-Fisher distribution), 27 vmf. da, 35, 81, 121 vmf. de (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde, 24, 118, 119, 125 vmf. kde, 27, 118, 125 vmf. kde, 110, 111, 118, 119, 126 vm. hde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde, 24, 118, 119, 125 vmf. kde, 27, 118, 125 vmf. kde, 110, 111, 118, 119, 126 vm. hde tune, (Tuning of the bandwidth parameter in the von Mises kernel), 118 vmkde. tune, (Tuning of the bandwidth parameter in the von Mises kernel density estimation, 124 vm Mises-Fisher kernel density estimation for (hyper-)spherical data), 123 watson, 12 watson, 123 weibull.nb, 75, 117 weibullnb.pred, 82 wood.mle, 70 wood.mle, 70 wood.mle, 70 wood.mle, 70 worldmap flex the von Mises-Fisher kernel density estimation on the sphere), 73 worldmap flex the von Mises-Fisher kernel density estimation, 102 wmf. kde, 110, 111, 118, 119, 126 vmf. kde, 110, 111, 118, 119, 126 vmkde. tune, 219, 24, 24, 118, 125 vmf. kde, tune, 219, 24, 24, 118, 125 vmf. kde, 110, 111, 118, 119, 126 vmkde. tune, (Tun		
The k-nearest neighbours using the cosinus distance, I15 Transform unit vectors to angular data, I16 Tuning of the bandwidth parameter in the von Mises kernel, I17 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, I18 Tuning of the k-NN algorithm using the arc cosinus distance, I19 Tuning of the k-NN regression, I21 Tuning of the k-NN regression, I21 Uniformity test for circular data, I23 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), I4 vm. kde, I10, I11, I18, I19, I26 vm. kde (von Mises kernel density estimation), I24 vm. mle, 21, 64 vm. mle, 21, 64 vm. nb, 82 vm. nb (Naive Bayes classifiers for circular data), 74 vmf. contour, 23, 24, 26, 51 vmf. da, 35, 81, 121 vmf. da (Cross validation in von Mises-Fisher distribution), 27 vmf. da, 35, 81, 121 vmf. da (Cross validation in von Mises-Fisher distribution), 27 vmf. da, 35, 81, 121 vmf. de (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. ket (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. ket nontour (Contour plot of spherical data using a von	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. kde (von Mises kernel density estimation), 124 vm. mle, 21, 64 vm. mle, 22, 64 vm. nb, 82 vm. nb (Naive Bayes classifiers for circular data), 74 vmf. contour, 23, 24, 26, 51 vmf. da (Cross validation in von Mises-Fisher distribution), 27 vmf. da, 35, 38, 12, 121 vmf.da pred, 32, 34, 35, 53, 80 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfda.pred, 32, 34, 35, 53, 80 vmfda.pred, 32, 34, 35, 53, 80 vmfda.pred, 32, 148, 125, 126 vmfkde.tune, 23, 24, 18, 125, 126 vmfkde.tune, 23, 24, 18, 125, 126 vmfkde.tune, 23, 24, 18, 125, 126 vmfkde.tune, 32, 24, 18, 125, 126 vm	-	
The k-nearest neighbours using the cosinus distance, I15 Transform unit vectors to angular data, I16 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, I17 Tuning of the k-NN algorithm using the arc cosinus distance, I19 Tuning of the k-NN regression, I21 Uniformity test for circular data, I23 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), I4 vm. kde, I10, I11, I18, I19, I26 vm. nb (Naive Bayes classifiers for circular data), 72 vmf. contour, 23, 24, 26, 51 vmf. contour (Contour plots of the von Mises-Fisher distribution), 27 vmf. da, 35, 81, I21 vmf. da (Cross validation in von Mises-Fisher distribution), 27 vmf. da (Cross validation in von Mises-Fisher distribution), 28 vmstde. tune, I19, I25 vmf. kde, 21, I18, I19, I25 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kerncontour, 23, 24, 26, 51 vmf. kerncontour, 23, 25-27, I19 vmf. kerncontour (Contour plot of spherical data), 125 vmf. kerncontour (Contour plot of spherical data), 125 vmf. kerncontour, 23, 25-27, I19 vmf. kerncontour (Contour plot of spherical data), 125 vmf. kerncontour (Contour plot of spherical data), 125 vmf. kerncontour (Contour plot of spherical data), 125	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm.kde (von Mises kernel density estimation), 124 vm. mle, 23, 24, 18, 19, 126 vm. hd (Naive Bayes classifiers for circular data), 74 vmf. contour (Contour plots of the von Mises-Fisher distribution), 27 vmf. da, 35, 81, 121 vmf. da (Cross validation in von Mises-Fisher distribution), 27 vmf. da, 2, 26, 51 vmf. de (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde; 24, 118, 119, 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper)-spherical data), 125 vmf. kde; 24, 118, 119, 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper)-spherical data), 125 vmf. kde; 24, 118, 119, 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper)-spherical data), 125 vmf. kde; 24, 118, 119, 125 vmf. kde; 24, 118, 1	**	
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the k-nN algorithm using the arc cosinus distance, 119 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN algorithm using the projected normal or the von Mises-Fisher kernel, 115 Transform unit vectors to angular data, 125 vmf.kele, 101 Tuning of the bandwidth parameter in the von Mises-Fisher kernel density extend tune, 119, 125, 126 vmkde. tune, (119, 125, 126 vmkde. tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel density estimation, 124 vmm. b, pred, 75, 1/7 vmb. pred, 75, 1/7 vmb. pred (Prediction in discrimant analysis) as a valual check (Check visually well chem, 23, 24, 26, 51 vmf. kell, 111, 118, 119, 126 vm kide, 110, 111, 118, 119, 12	marameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm.kde, 110, 111, 118, 119, 126 vm.mle, 21, 64 vm.mle, 21, 64 vm.nb, 82 vm.nb, 82 vm.nb (Naive Bayes classifiers for circular data), 74 vmf. contour, 23, 24, 26, 51 vmf. da (Cross validation in von Mises-Fisher distribution), 27 vmf. da, 35, 81, 121 vmf. da (Cross validation in von Mises-Fisher distribution), 27 vmf. da, 35, 81, 121 vmf. da (Cross validation in von Mises-Fisher distribution), 27 vmf. da (Spherical regression using the projected normal or the von Mises-fisher distribution), 105 vmfd.a.pred, 32, 34, 35, 55, 80 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher distribution on Mises-fisher kernel density estimation, 124 vmon.bp.red, 79, 117 vmb.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80 vmfkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher distribution), 105 vmfkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher density estimation for (hyper-)spherial data, 123 vm Mises-Fisher distribution), 27 vmf. de (Cross validation), 27 vmf. de (Von Mises Fisher deror), 24 vm Mises-Fisher distribution, 26 vmfkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises-Fisher deror), 117 vmb.pred (Prediction in the vo		
The k-nearest neighbours using the cosinus distance, I15 Transform unit vectors to angular data, I16 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, I17 Tuning of the k-NN algorithm using the arc cosinus distance, I19 Tuning of the k-NN regression, I21 Uniformity test for circular data, I23 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is correctly generated or not), I4 vm. kde, I10, I11, I18, I19, I26 vm. nb (Naive Bayes classifiers for circular data), 72 vmf. contour, 23, 24, 26, 51 vmf. contour (Contour plots of the von Mises-Fisher distribution), 27 vmf. da, 35, 81, I21 vmf. da (Cross validation in von Mises-Fisher distribution), 27 vmf. da (Cross validation in von Mises-Fisher distribution), 28 vmstde. tune, I19, I25 vmf. kde, 21, I18, I19, I25 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kde (von Mises-Fisher kernel density estimation for (hyper-)spherical data), 125 vmf. kerncontour, 23, 24, 26, 51 vmf. kerncontour, 23, 25-27, I19 vmf. kerncontour (Contour plot of spherical data), 125 vmf. kerncontour (Contour plot of spherical data), 125 vmf. kerncontour, 23, 25-27, I19 vmf. kerncontour (Contour plot of spherical data), 125 vmf. kerncontour (Contour plot of spherical data), 125 vmf. kerncontour (Contour plot of spherical data), 125	marameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, 118 Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual. check (Check visually whether matrix Fisher samples is correctly generated or not), 14 vm. kde, 110, 111, 118, 119, 126 vm. nb (Naive Bayes classifiers for circular data), 74 vmf. contour, 23, 24, 26, 51 vmf. contour (Contour plots of the von Mises-Fisher distribution), 27 vmf. da, 35, 81, 121 vmf. da (Cross validation in von Mises-Fisher distribution), 27 vmf. da, 27, 28, 42, 61, 51 vmf. kde (von Mises-Fisher kernel density estimation, 124 vmf. da (Cross validation in von Mises-Fisher distribution), 27 vmf. da, 27, 34, 50, 55, 80 vmfda. pred (Prediction in discriminant analysis based on von Mises-Fisher kernel), 118 vmkde. tune, (119, 125, 126 vmkde. tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde. tune, 119, 125, 126 vmkde. tune, 119, 125, 126 vmkde. tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde. tune, 119, 125, 126 vmkde. tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde. tune, 119, 125, 126 vmkde. tune, 119, 125, 126 vmkde. tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde. tune, 119, 125, 126 vmkde. tune (Tuning of the bandwidth parameter in the von Mises-Fisher kernel), 118 vmkde. tune, 119, 125, 126 vmkde. tune (Mises-Fisher kernel), 118 vmkde. tune, 119, 125, 126 vmkde. tune (Mises-Fisher kernel), 118 vmkde. tune, 119, 125, 126 vmkde. tune (Mises-Fisher kernel), 118 vmkses-Fisher kernel density estimation for (hyper-)spherical dat	101, 104, 100, 107, 110, 112, 124,	
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, the von Mises-Fisher kernel, the von Mises-Fisher kernel, wmfkde.tune, 23, 24, 118, 125, 126 the von Mises-Fisher kernel, vmfkde.tune (Tuning of the bandwidth	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises-Fisher kernel, the von Mises-Fisher kernel, distributions), 62 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 1 vmfda.pred, 32, 34, 35, 55, 80 vmfda.pred (Prediction in discriminary analysis based on von Mises-Fisher distribution), 8 vmfkde.tune, 23, 24, 118, 125, 126 vmfkde.tune (Tuning of the bandwidth	Tuning of the k-NN algorithm using the arc cosinus distance, 119 Tuning of the k-NN regression, 121 Uniformity test for circular data, 123 vec (Generation of unit vector(s) with a given angle), 44 visual.check (Check visually whether matrix Fisher samples is	Mises-Fisher kernel), 118 vmkde.tune, 119, 125, 126 vmkde.tune (Tuning of the bandwidth parameter in the von Mises kernel), 117 vmnb.pred, 75, 117 vmnb.pred (Prediction with some naive Bayes classifiers for circul data), 82 von Mises kernel density estimation,
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the bandwidth parameter in the von Mises kernel, 117	parameters for circular data, 114 The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning distributions), 62 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred, 32, 34, 35, 55, 80 vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80	the von Mises-Fisher kernel,	vmfkde.tune(Tuning of the bandwidth
The k-nearest neighbours using the cosinus distance, 115 Transform unit vectors to angular data, 116 Tuning of the bandwidth parameter in the von Mises kernel, 117 Tuning of the von Mises kernel, 117 Tuning of the von Mises kernel, 117 Tuning of the von Mises kernel, 117 Vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 Vmfda.pred (Prediction in discriminant analysis based on von Mises-Fisher distribution), 80	parameters for circular data, 114 vmf.reg(Spherical regression using the rosinus distance, 115 vmfda.pred, 32 , 34 , 35 , 55 , 80 vmfda.pred(Prediction in discriminant analysis based on von the von Mises-Fisher distribution), 80		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	parameters for circular data, $\frac{114}{114}$ uning of the bandwidth parameter in distributions), 62 vmf.reg (Spherical regression using the projected normal or the von Mises-Fisher distribution), 105 vmfda.pred, 32, 34, 35, 55, 80 vmfda.pred (Prediction in discriminant analysis based on von		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	parameters for circular data, 114 vmf.reg(Spherical regression using the cosinus distance, 115 vmfda.pred, 32 , 34 , 35 , 55 , 80 vmfda.pred(Prediction in discriminant		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	parameters for circular data, $\frac{114}{114}$ vmf.reg(Spherical regression using the $\frac{114}{115}$ cosinus distance, $\frac{115}{115}$ mises-Fisher distribution), $\frac{105}{115}$ Transform unit vectors to angular vmfda.pred, $\frac{32}{115}$, $\frac{34}{115}$, $\frac{35}{115}$, $\frac{80}{115}$		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	parameters for circular data, distributions), 62 114 1		
$\frac{114}{\text{The k-nearest neighbours using the}} \hspace{2.5cm} \text{vmf.reg(Spherical regression using the} \\ \hspace{2.5cm} \text{projected normal or the von} \\$	parameters for circular data, distributions), 62 114 vmf.reg(Spherical regression using the distributions) the projected normal or the von		
vmf.reg(Spherical regression using the	parameters for circular data, distributions), 62 114 vmf.reg(Spherical regression using the		
•	parameters for circular data, distributions), 62		
		parameters for circular data,	**
or education or end contents action	lest of equality of the concentrationvmf_mle(MLF of (hyper-)spherical		