Package 'circular'

August 29, 2024

Title Circular Statistics

LazyLoad yes

LazyData yes

Version 0.5-1

Date 2024-08-28

Depends R (>= 3.0.0), stats

Imports boot, mvtnorm

Description Circular Statistics, from ``Topics in circular Statistics'' (2001) S. Rao Jammala-madaka and A. SenGupta, World Scientific.

License GPL-2

NeedsCompilation yes

Encoding UTF-8

Author Ulric Lund [aut],
Claudio Agostinelli [aut] (<https://orcid.org/0000-0001-6702-4312>),
Hiroyoshi Arai [ctb],
Alessando Gagliardi [ctb],
Eduardo García-Portugués [ctb, cre]
(<https://orcid.org/0000-0002-9224-4111>),
Dimitri Giunchi [ctb],
Jean-Olivier Irisson [ctb],
Matthew Pocernich [ctb],
Federico Rotolo [ctb]

Maintainer Eduardo García-Portugués <edgarcia@est-econ.uc3m.es>

Repository CRAN

Date/Publication 2024-08-29 07:50:02 UTC

Contents

Contents 3

A1   *Ratio of First and Zeroth Order Bessel Functions*

Description

Evaluates the first and zeroth order Bessel functions of the first kind at a specified non-negative real number, and returns the ratio.

Usage

```
A1(kappa)
```

Arguments

kappa   non-negative numeric value at which to evaluate the Bessel functions.

Details

The function uses besselI.

Value

If I1(kappa) is the first order Bessel function and I0(kappa) is the zeroth order Bessel function, then A1(kappa) returns I1(kappa)/I0(kappa).

Author(s)

Claudio Agostinelli

See Also

besselI, A1inv.

A1FirstDerivative | *First derivative of the Ratio of First and Zeroth Order Bessel Functions.*

Description

Evaluates the first derivative of the Ratio of First and Zeroth Order Bessel Functions

Usage

```
A1FirstDerivative(kappa)
```

Arguments

kappa   non-negative numeric value at which to evaluate the first derivative of A1 function.

Details

The formula (3.48) of Fisher (1993), pag. 52 is implemented. The function uses A1 and besselI.

Value

The value of the first derivative of A1 function in the point kappa.

Author(s)

Claudio Agostinelli and Alessandro Gagliardi.

References

N.I. Fisher (1993) Statistical Analysis of Circular Data, Cambridge University Press.

See Also

A1, besselI, A1inv.

A1inv	*Inverse of A1*

Description

Inverse function of the ratio of the first and zeroth order Bessel functions of the first kind. This function is used to compute the maximum likelihood estimate of the concentration parameter of a von Mises distribution.

Usage

```
A1inv(x)
```

Arguments

x numeric value in the interval between 0 and 1.

Details

A1inv(0) = 0 and A1inv(1) = Inf. This function is useful in estimating the concentration parameter of data from a von Mises distribution. Our function use the results in Best and Fisher (1981). Tables use tabulated values by Gumbel, Greenwood and Durand (1953).

Value

Returns the value k, such that A1inv(x) = k, i.e. A1(k) = x.

Author(s)

Claudio Agostinelli

References

BEST, D.J. and FISHER, N.I. 1981. The bias of the maximum likelihood estimators for the von Mises-Fisher concentration parameters. Communications in Statistics, 10, 493-502.

GUMBEL, E.J., GREENWOOD, J.A. AND DURAND, D. 1953. The circular normal distribution: theory and tables. J. Amer. Statis. Assoc., 48, 131-152.

See Also

mle.vonmises, A1, besselI.

Examples

```
#Generate data from a von Mises distribution
data <- rvonmises(n=50, mu=circular(pi), kappa=4)
#Estimate the concentration parameter
s <- sum(sin(data))
c <- sum(cos(data))
mean.dir <- atan2(s, c)
kappa <- A1inv(mean(cos(data - mean.dir)))
```

A1SecondDerivative | *Second derivative of the Ratio of First and Zeroth Order Bessel Functions.*

Description

Evaluates the second derivative of the second derivative of the Ratio of First and Zeroth Order Bessel Functions.

Usage

```
A1SecondDerivative(kappa)
```

Arguments

kappa   non-negative numeric value at which to evaluate the second derivative of A1 function.

Details

Formula (3.49) of Fisher (1993), pag. 52 is implemented. The function uses A1, A1FirstDerivative and besselI.

Value

The value of the second derivative of A1 function in the point kappa.

Author(s)

Claudio Agostinelli and Alessandro Gagliardi.

References

N.I. Fisher (1993) Statistical Analysis of Circular Data, Cambridge University Press.

See Also

A1, A1FirstDerivative, besselI, A1inv.

---

angular.deviation　　　*A measure of deviation for Circular Data*

Description

Returns the square root of twice one minus the mean resultant length divided by the sample size of a vector of circular data.

Usage

```
angular.deviation(x, na.rm = FALSE)
```

Arguments

x　　　　　　　a vector. The object is coerced to class `circular`.

na.rm　　　　　logical, indicating if `NA`'s should be omitted.

Value

Returns the square root of twice one minus the mean resultant length divided by the sample size.

Author(s)

Claudio Agostinelli

References

Batschelet, E. (1981) Circular Statistics in Biology. Academic Press, London.

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 1.3, World Scientific Press, Singapore.

Zar, J.H. (2010) Biostatistical Analysis. Fifth edition. Pearson Educational International.

See Also

`sd.circular`, `angular.variance`, `mean.circular`, `rho.circular` and `summary.circular`.

Examples

```
x <- rvonmises(n=100, mu=circular(0), kappa=1)
angular.deviation(x)
```

angular.variance   *A measure of variance for Circular Data*

Description

Returns twice one minus the mean resultant length divided by the sample size of a vector of circular data.

Usage

```
angular.variance(x, na.rm = FALSE)
```

Arguments

x	a vector. The object is coerced to class `circular`.
na.rm	logical, indicating if `NA`'s should be omitted.

Value

Returns twice one minus the mean resultant length divided by the sample size.

Author(s)

Claudio Agostinelli

References

Batschelet, E. (1981) Circular Statistics in Biology. Academic Press, London.

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 1.3, World Scientific Press, Singapore.

Zar, J.H. (2010) Biostatistical Analysis. Fifth edition. Pearson Educational International.

See Also

`var.circular`, `angular.deviation`, `mean.circular`, `rho.circular` and `summary.circular`.

Examples

```
x <- rvonmises(n=100, mu=circular(0), kappa=1)
angular.variance(x)
```

---

aov.circular | *Analysis of Variance for circular data*

Description

One Critrion Analysis of Variance for circular data

Usage

```
aov.circular(x, group, kappa = NULL,
     method = c("F.test", "LRT"), F.mod = TRUE, control.circular=list())
## S3 method for class 'aov.circular'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	a vector of class `circular`.
group	a vector identifying the groups or samples.
kappa	the common value of the concentration parameter. Used when `method` is `"LRT"`. If left unspecified (by default) the maximum likelihood estimate of kappa is computed and used in the test statistic.
method	the test statistic to use; either a high-concentration F-test or a likelihood ratio test.
F.mod	logical; if TRUE, the AOV F-statistic is modified by a factor of $1+3/8k$ to improve its approximate F distribution. Default is TRUE.
control.circular	
	the coordinate system used in the output for the objects `mu` and `mu.all`. See [circular](#) for details.
digits	the number of digits to be printed.
...	additional arguments.

Details

The samples are assumed to have been drawn from von Mises populations with equal concentration parameter, kappa. The null hypothesis being tested is that all populations also have the same mean direction.

If `method` is `"F.test"` a high concentration F-test makes use of a decomposition of total sample variation into between groups and within groups variation, analogous to the one-way classification analysis of variance for linear data. Stephens (1972) presented an improved modification to the F-test derived from this decomposition. This is implemented when `F.mod` is TRUE.

A likelihood ratio test is performed when `method` is `"LRT"`. This function uses the test statistic presented by Cordeiro, Paula, and Botter (1994) which has an approximate chi-squared distribution. If the common concentration parameter is known, it can be specified and used in the computation of the test statistic. Otherwise, the maximum likelihood estimate of the common concentration parameter is used.

Value

An object of class `aov.circular` with the following components:

mu	mean direction for each sample with class `circular`.
mu.all	mean direction of all samples combined with class `circular`.
kappa	concentration parameter for each sample.
kappa.all	concentration parameter for all samples combined.
rho	mean resultant length for each sample.
rho.all	mean resultant length for all samples combined.
method	the test statistic used.
df	degrees of freedom.
statistic	the value of the test statistic.
p.value	the p.value of the test statistic.
call	the match.call result.

If the `method` is `"F.test"` then the object contains also:

SSE	Sum of squares used in F-test.
MSE	Mean squares used in F-test.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Cordeiro, G., Paula, G. and Botter, D. (1994). Improved likelihood ratio tests for dispersion models. International Statistical Review, 62, 257-274.

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 5.3, World Scientific Press, Singapore.

Mardia, K. and Jupp, P. (1999). Directional Statistics, Section 7.4, John Wiley and Sons, England.

Stephens, M. (1972). Multi-sample tests for the von Mises distribution. Technical Report 190, Department of Statistics, Stanford University.

Examples

```
x <- c(rvonmises(50, circular(0), 1), rvonmises(100, circular(pi/3), 10))
group <- c(rep(0, 50), rep(1, 100))

aov.circular(x, group)
aov.circular(x, group, method="LRT")
```

---

arrows.circular   *Add Arrows to a Circular Plot*

Description

Draw arrows in a circular plot.

Usage

```
arrows.circular(x, y = NULL, x0 = 0, y0 = 0, na.rm = FALSE,
  shrink = 1, plot.info = NULL, zero = NULL, rotation = NULL, ...)
```

Arguments

x	a vector. The object is coerced to class `circular`.
y	a vector with the same length as x.
x0	a vector of origins (x axis).
y0	a vector of origins (y axis).
na.rm	logical, indicating if `NA`'s should be omitted.
shrink	parameter that controls the size of the plotted circle. Default is 1. Larger values shrink the circle, while smaller values enlarge the circle.
plot.info	an object from `plot.circular` that contains information on the zero, the rotation and next.points.
zero	the zero used in the plot. Ignored if plot.info is provided.
rotation	the rotation used in the plot. Ignored if plot.info is provided.
...	further parameters passed to `arrows`.

Note

The function call `arrows` and it is not a method of `arrows`.

Author(s)

Claudio Agostinelli

See Also

`arrows`

Examples

```
plot(rvonmises(10, circular(0), kappa=1))
arrows.circular(rvonmises(10, circular(0), kappa=1))
arrows.circular(rvonmises(10, circular(0), kappa=1), y=runif(10), col=2)
arrows.circular(rvonmises(10, circular(0), kappa=1), y=runif(10),
  x0=runif(10, -1, 1), y0=runif(10, -1, 1), col=3)
```

```
as.data.frame.circular
```
as.data.frame.circular

Description

This function is a method of [as.data.frame](#) for a circular object.

Usage

```
## S3 method for class 'circular'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	object of class [circular](#).
row.names	NULL or a character vector giving the row names for the data frame. Missing values are not allowed.
optional	logical; if TRUE setting row names is optional.
...	additional arguments to be passed to or from methods.

Author(s)

Claudio Agostinelli

```
asytriangular
```
Asymmetric Triangular Density Function

Description

Density the Asymmetric Triangular circular distribution.

Usage

```
dasytriangular(x, rho)
```

Arguments

x	a vector. The object is coerced to class [circular](#).
rho	concentration parameter of the distribution. rho must be between 0 and $1/pi$.

Value

dasytriangular gives the density.

Author(s)

Claudio Agostinelli

References

Mardia (1972) Statistics for Directional Data, Wiley. Pag. 52

Examples

```
ff <- function(x) dasytriangular(x, rho=0.3)
curve.circular(ff, shrink=1.2, join=TRUE)
```

---

axialvonMises   *Axial von Mises Density Function*

Description

Density for the axial von Mises circular distribution.

Usage

```
daxialvonmises(x, mu, kappa, l = 2)
```

Arguments

x	a vector. The object is coerced to class [circular](#).
mu	mean direction of the distribution. The object is coerced to class [circular](#)
kappa	non-negative numeric value for the concentration parameter of the distribution.
l	a positive number. l=2 provide the axial distribution in the range [0, pi].

Value

daxialvonmises gives the density.

Author(s)

Claudio Agostinelli

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 2.2.4, World Scientific Press, Singapore.

axis.circular	*Add Axis to a Circular Plot*

Description

Add axis to a plot of circular data points on the current graphics device.

Usage

```
axis.circular(at=NULL, labels=NULL,  units = NULL, template=NULL,
          modulo=NULL, zero=NULL, rotation=NULL, tick=TRUE, lty, lwd,
          cex, col, font, tcl=0.025, tcl.text=0.125, digits=2)
```

Arguments

at	the points at which tick-marks are to be drawn. If NULL the tick-marks are placed to 0, pi/2, pi and 3pi/2 radians.
labels	a vector of character strings to be placed at the tickpoints. If NULL the labels are chosen according to units and template arguments.
units	either radians or degrees. If NULL the value is taken from the attributes of the object at.
template	either none or geographics. If NULL the value is taken from the attributes of the object at.
modulo	either asis or 2pi or pi. If NULL the value is taken from the attributes of the object at.
zero	the zero of the plot (in radians, counterclockwise). If NULL the value is taken from the attributes of the object at.
rotation	the rotation of the plot. If NULL the value is taken from the attributes of the object at.
tick	logical: if TRUE ticks are plotted at tick-marks.
lty, lwd	line type, width for the tick marks. If missing means to use 'par("lty")' and 'par("lwd")'.
cex	a numerical value giving the amount by which plotting text and symbols should be scaled relative to the default.
col	color for the the tick marks. If missing means to use 'par("col.axis")'.
font	font for text. If missing means to use 'par("font.axis")'.
tcl	The length of tick marks.
tcl.text	The position of the axis labels.
digits	number of digits used to print axis values.

Author(s)

Claudio Agostinelli

See Also

[plot.circular](#) and [ticks.circular](#).

Examples

```
data.vm <- rvonmises(n=100, mu=circular(0), kappa=3)
plot(data.vm, axes=FALSE, ticks=FALSE)
axis.circular(at=circular(seq(0, 11/6*pi, pi/6)), labels=c("0",
expression(frac(pi,6)), expression(paste(frac(1,3), pi)),
expression(frac(pi,2)), expression(paste(frac(2,3), pi)),
expression(paste(frac(5,6), pi)), expression(pi),
expression(paste(frac(7,6), pi)), expression(paste(frac(4,3), pi)),
expression(paste(frac(3,2), pi)), expression(paste(frac(5,3), pi)),
expression(paste(frac(11,6), pi))))
```

---

bandwidth	*Bandwidth Selectors for Kernel Density Estimation for Circular Data*

Description

Bandwidth selectors for circular kernels in [density.circular](#).

Usage

```
bw.cv.mse.circular(x, lower=NULL, upper=NULL, tol = 1e-4,
  kernel = c("vonmises", "wrappednormal"), K = NULL, min.k = 10)

bw.cv.ml.circular(x, lower=NULL, upper=NULL, tol = 1e-4,
  kernel = c("vonmises", "wrappednormal"), K = NULL, min.k = 10)

bw.nrd.circular(x, lower=NULL, upper=NULL,
  kappa.est=c("ML","trigmoments"), kappa.bias=FALSE, P=3)
```

Arguments

x	the data from which the bandwidth is to be computed. The object is coerced to class circular.
lower, upper	range over which to minimize for cross validatory bandwidths. The default is almost always satisfactory, although it is recommended experiment a little with different ranges. A warning message indicates if the resulting bandwidth is too near to the endpoints of the interval search.
tol	for cross validatory bandwidths, the convergence tolerance for [optimize](#).
kernel	a character string giving the smoothing kernel to be used. This must be one of "vonmises" or "wrappednormal".
K	number of terms to be used in approximating the wrappednormal density. See [dwrappednormal](#).

min.k	minimum number of terms used in approximating the wrappednormal density. See [dwrappednormal](#).
kappa.est	a numerical value or one available method.
kappa.bias	logical. If TRUE, when kappa.est=="ML" a bias correction in the estimation of kappa is applied.
P	integer, the maximum order of the sample trigonometric moments used in the estimation of kappa when kappa.est=="trigmoments", see Details.

Details

bw.cv.mse.circular and bw.cv.ml.circular implement cross validatory bandwidths minimizing squared–error loss and Kullback–Leibler loss, respectively. This is done by minimizing the second and third equations in section 5 of Hall, Watson and Cabrera (1987). Kullback–Leibler loss is equivalent to maximize the cross validation log–likelihood with respect to the bandwidth parameter.

bw.nrd.circular implements a rule-of-thumb for choosing the bandwidth of a von Mises kernel density estimator with underlying population von Mises. It was proposed by Taylor (2008, equation (7)) and is the circular analogue of the usual rule of thumb used for the normal distribution. The only remarkable difference between them is that Taylor's bandwidth supposes a von Mises population for the derivation of AMISE, while normal rule of thumb only introduces distribution assumption to compute the density curvature. Estimation of the spread is done by maximum likelihood. The "trigmoments" method for the estimation of kappa is implemented as follows. Let μ_p be the p-th sample trigonometric moment. Let k_p be the estimates of kappa using the p-th sample trigonometric moment, as solution (using uniroot function) of the equation $A_p(k) = \frac{1}{n} \sum_{i=1}^{n} \cos(px_i - \mu_p)$. We let kappa equal to $max(k_1, k_2, \cdots, k_P)$, see Taylor (2008) for further details.

Note that circular bandwidth has a different scale from linear bandwidth (see Hall, Watson and Cabrera (1987)). The behaviour of the circular bandwidth is the *inverse* of the linear: large values overestimate the density, whereas small values underestimate.

Value

A bandwidth on a scale suitable for the bw argument of density.circular.

Warning

Plug-in bandwidth selector bw.nrd.circular assumes that the underlying population is von Mises. If this is not true, it might lead to serious misestimations of the circular bandwidth. Example 2 below shows how this behaviour can appear with multimodality populations. In those cases, the use of kappa.est="trigmoments" could be of help.

Author(s)

Claudio Agostinelli and Eduardo Garcia–Portugues

References

P. Hall and G.S. Watson and J. Cabrera (1987). Kernel Density Estimation with Spherical Data, Biometrika, 74, 4, 751–762.

C.C Taylor (2008). Automatic bandwidth selection for circular density estimation. Computational Statistics and Data Analysis, 52, 7, 3493–3500.

See Also

density.circular

Examples

```
set.seed(12345)

## Example 1: von Mises ##
theta1 <- rvonmises(n=150,mu=circular(pi),kappa=2)

bw.nrd1 <- bw.nrd.circular(theta1)
bw.cv.mse1 <- bw.cv.mse.circular(theta1)
bw.cv.ml1 <- bw.cv.ml.circular(theta1)

## Linear plot
plot(function(x) dvonmises(circular(x), mu=circular(pi), kappa=2),
type="l", lwd=2, col=1, main="von Mises", xlab=expression(theta),
ylab="Density", from=0, to=2*pi)
plot(approxfun(density.circular(x=theta1, bw=bw.nrd1)), col=2, from=0, to=2*pi, add=TRUE)
plot(approxfun(density.circular(x=theta1, bw=bw.cv.mse1)), col=3,
from=0, to=2*pi, add=TRUE)
plot(approxfun(density.circular(x=theta1, bw=bw.cv.ml1)), col=4, from=0,
to=2*pi, add=TRUE)
legend("topright", legend=c("True", "Taylor", "LSCV", "MLCV"), col=1:4, lwd=2)
rug(theta1)

## Circular plot
dvonmises1 <- function(x) dvonmises(circular(x), mu=circular(pi), kappa=2)
curve.circular(dvonmises1, lwd=2, col=1, main="von Mises", xlim=c(-1.5,
1.5), ylim=c(-1.5,1.5))
lines(density.circular(x=theta1, bw=bw.nrd1), col=2)
lines(density.circular(x=theta1, bw=bw.cv.mse1), col=3)
lines(density.circular(x=theta1, bw=bw.cv.ml1), col=4)
legend("topright", legend=c("True", "Taylor", "LSCV", "MLCV"), col=1:4, lwd=2)
points(theta1)

## Example 2: mixture of von Mises ##

theta2 <- rmixedvonmises(n=150, mu1=circular(pi/2),
mu2=circular(3*pi/2), kappa1=5, kappa2=5,p=0.5)

bw.nrd2 <- bw.nrd.circular(theta2)
bw.cv.mse2 <- bw.cv.mse.circular(theta2)
bw.cv.ml2 <- bw.cv.ml.circular(theta2)

## Linear plot
plot(function(x) dmixedvonmises(circular(x), mu1=circular(pi/2),
mu2=circular(3*pi/2), kappa1=5, kappa2=5, p=0.5), type="l", lwd=2,
col=1, main="mixture of von Mises", xlab=expression(theta),
```

```
ylab="Density", from=0, to=2*pi)
lines(density.circular(x=theta2, bw=bw.nrd2), plot.type='line', col=2)
lines(density.circular(x=theta2, bw=bw.cv.mse2), plot.type='line',
col=3)
lines(density.circular(x=theta2, bw=bw.cv.ml2), plot.type='line', col=4)
rug(theta2)
legend("topright", legend=c("True", "Taylor", "LSCV", "MLCV"), col=1:4, lwd=2)

## Circular plot
dmixedvonmises1 <- function(x) dmixedvonmises(circular(x), mu1=circular(pi/2),
mu2=circular(3*pi/2), kappa1=5, kappa2=5, p=0.5)
curve.circular(dmixedvonmises1, join=TRUE,
xlim=c(-1.5, 1.5), ylim=c(-1.5, 1.5), lwd=2, col=1, main="mixture of von
Mises")
lines(density.circular(x=theta2, bw=bw.nrd2), col=2)
lines(density.circular(x=theta2, bw=bw.cv.mse2), col=3)
lines(density.circular(x=theta2, bw=bw.cv.ml2), col=4)
points(theta2)
legend("topright", legend=c("True", "Taylor", "LSCV", "MLCV"), col=1:4, lwd=2)

## Example 3: mixture of von Mises and Wrapped Cauchy ##

rmixture <- function(n){
  x <- circular(sapply(runif(n), function(u) ifelse(u>0.5,
  rvonmises(n=1, mu=circular(pi),kappa=10),
  rwrappedcauchy(n=1,mu=circular(pi/2),rho=0.75))))
  return(x)
}

theta3 <- rmixture(n=150)

bw.nrd3 <- bw.nrd.circular(theta3)
bw.cv.mse3 <- bw.cv.mse.circular(theta3, lower=0.1, upper=100)
bw.cv.ml3 <- bw.cv.ml.circular(theta3, lower=0.1, upper=100)

dmixture <- function(x) (dvonmises(x, mu=circular(pi),
kappa=10)+dwrappedcauchy(x, mu=circular(pi/2), rho=0.75))/2
curve.circular(dmixture, join=TRUE, xlim=c(-1.5, 1.5), ylim=c(-1.5,
1.5), lwd=2, col=1, main="mixture of von Mises and Wrapped Normal")
lines(density.circular(x=theta3, bw=bw.nrd3), col=2)
lines(density.circular(x=theta3, bw=bw.cv.mse3), col=3)
lines(density.circular(x=theta3, bw=bw.cv.ml3), col=4)
legend("topright", legend=c("True", "Taylor", "LSCV", "MLCV"), col=1:4, lwd=2)
points(theta3)
```

| c.circular | *A method for circular object, which combines its arguments* |

Description

A method for circular object, which combines its arguments

Usage

```
## S3 method for class 'circular'
c(..., recursive = FALSE)
```

Arguments

`...`	vectors, the first of which of class `circular`.
`recursive`	logical. If 'recursive=TRUE', the function recursively descends through lists combining all their elements into a vector.

Author(s)

Claudio Agostinelli

See Also

[c](#)

Examples

```
x <- rvonmises(10, circular(0), 10)
y <- rvonmises(10, circular(0), 10, control.circular=list(units="degrees"))
z <- runif(10, 0, 20) # here you do not use circular properties,
#####but you mean it is measured in degrees
c(x, y, z) # While y is converted in radians, z is treated as it was!
```

Cardioid   *Cardioid Density Function*

Description

Density and random generation for the Cardioid circular distribution.

Usage

```
dcardioid(x, mu = circular(0), rho = 0)
rcardioid(n, mu = circular(0), rho = 0, control.circular=list())
```

Arguments

`x`	a vector. The object is coerced to class [circular](#).
`n`	number of observations.
`mu`	mean direction of the distribution. The object is coerced to class [circular](#).
`rho`	concentration parameter of the distribution. Absolute value of `rho` must be less than 0.5.
`control.circular`	
	the coordinate system used in the output of rcardioid. See [circular](#) for details.

Value

dcardioid gives the density and rcardioid generates random deviates.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 2.2.2, World Scientific Press, Singapore.

Examples

```
set.seed(1234)
resrad <- rcardioid(n=10)
set.seed(1234)
resdeg <- rcardioid(n=10, control.circular=list(units="radians", zero=pi))
max(abs(resrad - conversion.circular(resdeg, zero=0)))
```

Carthwrite   *Carthwrite's Power-of-Cosine Density Function*

Description

Density for the Carthwrite's power-of-cosine distribution.

Usage

```
dcarthwrite(x, mu, psi)
```

Arguments

x	a vector. The x and q objects are coerced to class `circular`.
mu	the location angular parameter. The object is coerced to class `circular`.
psi	the positive shape parameter.

Details

The Carthwrite's power-of-cosine distribution has density

$$f(x) = \frac{2^{(1/\psi)-1}\Gamma^2((1/\psi)+1)(1+\cos(x-\mu))^{1/\psi}}{\pi\Gamma((2/\psi)+1)},$$

for $0 \le x < 2\pi$.

Value

The density

Author(s)

Federico Rotolo

References

Carthwrite, D.E. (1963). The use of directional spectra in studying the output of a wave recorder on a moving ship. Ocean Wave Spectra , 203-218.

---

change.point　　　　　　　　　　　　　*Change Point Test*

Description

Tests for a change in mean direction, concentration, or both, given a set of directional data points.

Usage

```
change.point(x)
```

Arguments

x　　　　　　　　　　a vector. The object is coerced to class `circular`.

Details

In either context, the user can choose which statistic (max or ave) to use, and then consult the appropriate table provided in the book referenced below. The critical values for these 4 statistics are to be found in Table 11.3 (or Figure 11.3) for rmax, Table 11.4 (or Figure 11.4) for rave, Figure 11.5 for tmax and Figure 11.6 for tave.

Value

Returns a list with variables n, rho, rmax, k.r, rave, tmax, k.t, and tave. The first of these is the sample size, followed by the overall mean resultant length. Both of these are needed to enter any of the tables or nomograms (see under Details). The other values represent the change point test statistics. While rmax and rave test for a change in mean direction (with unknown concentration), tmax and tave are useful in the context of testing more generally, for a change in mean direction and/or concentration. k.r and k.t are the observation numbers for which rmax and tmax attain their maximum value and indicate the observation at which the change is most likely to have occurred, when the tables or nomograms indicate significance.

Author(s)

Claudio Agostinelli and Ulric Lund

See Also

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Chapter 11, World Scientific Press, Singapore.

circle.control	*Auxiliary for Controlling Circular Plots*

Description

Auxiliary function as user interface for circular plots. Typically only used when calling plot.circular.

Usage

```
circle.control(n = 1000, type = "l", col = 1, bg = par("bg"),
  pch = 1, cex = 1, lty = 1, lwd = 1)
```

Arguments

n	number of points used to interpolate the circle
type	1-character string giving the type of plot desired. The following values are possible, for details, see [plot]: "p" for points, "l" for lines, "o" for overplotted points and lines, "b", "c" for (empty if "c") points joined by lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
col	The color used.
bg	The color to be used for the background of the device region.
pch	Either an integer specifying a symbol or a single character to be used as the default in plotting points. See [points] for possible values and their interpretation. Note that only integers and single-character strings can be set as a graphics parameter (and not NA nor NULL).
cex	A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default.
lty	The line type. Line types can either be specified as an integer (0=blank, 1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash) or as one of the character strings "blank", "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash", where "blank" uses 'invisible lines' (i.e., does not draw them). Alternatively, a string of up to 8 characters (from c(1:9, "A":"F")) may be given, giving the length of line segments which are alternatively drawn and skipped. See section 'Line Type Specification'.
lwd	The line width, a positive number, defaulting to 1. The interpretation is device-specific, and some devices do not implement line widths less than one. (See the help on the device for details of the interpretation.)

Author(s)

Claudio Agostinelli

See Also

[plot.circular]

Examples

```
plot(rvonmises(10, circular(0), 1), control.circle=circle.control(col=2, lty=2))
```

Circular   *Package 'circular': summary information*

Description

The package 'circular' provides functions for the statistical analysis and graphics representation of circular data (observations which are angles). It originally started as a porting from S-plus to R of functions developed for the book: Circular Statistics, from "Topics in circular Statistics" (2001) S. Rao Jammalamadaka and A. SenGupta, World Scientific. Now, it has an S3 implementation and several new functions and datasets.

Version

The version level of the package is given by the command packageDescription("circular"). The most recent version of the package can be obtained from the R-Forge repository at https://r-forge.r-project.org/projects/circular/

Author

Claudio Agostinelli, Department of Mathematics University of Trento, Italy (http://datascience.maths.unitn.it/~claudio/)

Ulric Lund, Department of Statistics, California Polytechnic State University, San Luis Obispo, California, USA (https://statistics.calpoly.edu/ulric-lund)

Licence

This package and its documentation are usable under the terms of the "GNU General Public License", a copy of which is distributed with the package. While the software is freely usable, it would be appreciated if a reference is inserted in publications or other work which makes use of it; for this purpose, see the command citation("circular").

Acknowledgements

The package has evolved through several versions, developed over some years.

Many thanks to all that points out bugs, provide suggestions and comments.

The functions median and medianHS are developed together with Alessandro Gagliardi mailto:alessandro.gagliardi@unipd.it

The functions watson.wiliams.test and wallraff.test are developed by Jean-Olivier Irisson (https://www.obs-vlfr.fr/~irisson/)

The functions dcarthwrite, dgenvonmises, (d,r)katojones, djonespewsey are developed by Federico Rotolo

The function rose.diag has contribution by Hiroyoshi Arai (mailto:h_arai@aoni.waseda.jp)

The function windrose is developed by Matthew Pocernich

Dataset swallows is kindly provided by Dimitri Giunchi http://unimap.unipi.it/cercapersone/dettaglio.php?ri=2504&template=dettaglio.tpl

The function bw.circular is developed together with Eduardo Garcia Portugues https://egarpor.github.io/

If I miss to report your contribution please let me know by email at mailto:claudio.agostinelli@unitn.it

---

| circular | *Create Objects of class circular for Circular data.* |

Description

The function circular is used to create circular objects. as.circular and is.circular coerce an object to a circular and test whether an object is a circular data.

Usage

```
circular(x, type = c("angles", "directions"),
  units = c("radians", "degrees", "hours"),
  template = c("none", "geographics", "clock12", "clock24"),
  modulo = c("asis", "2pi", "pi"),
  zero = 0, rotation = c("counter", "clock"), names)
## S3 method for class 'circular'
as(x, control.circular=list(), ...)
## S3 method for class 'circular'
is(x)
## S3 method for class 'circular'
print(x, info=TRUE, ...)
```

Arguments

x	a vector or a matrix. If a data.frame is supply then it is coerced to a matrix.
type	the type of measures (Not Used Yet).
units	units of the measures.
template	how the data should be plotted. This set modulo, zero and rotation to some suitable values. For instance for 'geographics': zero=pi/2 and rotation='clock'. It is also used to set default labels on the plots.
modulo	if we need to reduce the measures to modulo.
zero	the zero of the axes (in radians, counter).
rotation	the orientation of the axes.
names	names of the data.
info	if TRUE information on the properties of the data are printed.

control.circular

> the attribute (coordinate system) used to coerced the resulting objects. See circular.

...   For as.circular an alternative way of setting the coordinate system of the resulting objects. Passed parameters to print.default for print.circular.

Value

an object of class circular. Since version 0.3-5 the previous class of the object is retain.

Author(s)

Claudio Agostinelli

See Also

conversion.circular

Examples

```
x <- circular(c(pi, pi/3, pi/4))
print(x)
is.circular(x)

x <- circular(runif(10, -pi/2, pi/2), template="geographics")
plot(x)
class(x)

x <- circular(data.frame(runif(10, -pi/2, pi/2)))
plot(x)
class(x)

cbind(x, x) # the matrix, cbind, rbind functions unclass and lost attributes!
########Use it with care.

x <- c(pi/12,2*pi+pi/12)
print(x)
x <- unique(x)
print(x)

x[1]==x[2]

all.equal(x[1], x[2])

x <- as.circular(pi, control.circular=list(units="radians", zero=pi))
y <- conversion.circular(circular(pi), zero=pi)
res <- plot(x)
points(y, col=2, plot.info=res)
```

Circular Uniform   *Circular Uniform Density Function*

Description

Density and random generation for the Circular Uniform distribution on the whole circle.

Usage

```
dcircularuniform(x)
rcircularuniform(n, control.circular=list())
```

Arguments

x   a vector. The object is not coerced to class [circular](#).

n   number of observations.

control.circular

                the attribute of the resulting object.

Value

dcircularuniform gives the density and rcircularuniform generates random deviates.

Author(s)

Claudio Agostinelli

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 2.2.1, World Scientific Press, Singapore.

Examples

```
data1 <- rcircularuniform(100, control.circular=list(units="degrees"))
plot(data1)

curve.circular(dcircularuniform, join=TRUE, xlim=c(-1.2, 1.2),
  ylim=c(-1.2, 1.2), main="Density of a Circular Uniform Distribution")
```

circular.colors   *Color Palettes for Circular*

Description

Create a vector of n contiguous colors.

Usage

```
circular.colors(n, m = 0, M = 2 * pi, offset = 0, ...)
```

Arguments

n	the number of colors (>= 1) to be in the palette.
m	the smallest angle in radians.
M	the largest angle in radians.
offset	the zero in radians.
...	further arguments passed to the function hsv.

Value

a vector of length n.

Author(s)

Claudio Agostinelli

See Also

hsv, colors

Examples

```
circular.colors(n=10, m=0, M=2*pi)
```

---

circularp   *Attributes for a Circular Object*

Description

'circularp' returns the 'circularp' attribute (or 'NULL'). 'circularp<-' sets the 'circularp' attribute.

Usage

```
circularp(x)
circularp(x) <- value
```

Arguments

x   a vector or a matrix of circular data.

value   a vector of length 6 or a list with six components: type, units, template, modulo, zero and rotation.

Details

The `circularp` attribute is a list of six elements: type, units, template, modulo, zero and rotation; see `circular` for their meaning.

Assignments are checked for consistency.

Assigning NULL removes the `circularp` attribute and any `"circular"` class of x.

Author(s)

Claudio Agostinelli

See Also

`circular`

Examples

```
x <- pi
circularp(x) # now NULL
circularp(x) <- list(type="angles", units="radians", template="none",
  modulo="asis", zero=0, rotation="counter")
circularp(x)
x
class(x) <- "circular" # now we set also the class so that print.circular is used
x
```

---

conversion.circular   *Unit of Measure Conversion for Circular Data and other conversions*

Description

Conversion for Circular Data from one coordinate/units system to another one. For back compatibility, without arguments the function converts data from degrees to radians.

Usage

```
conversion.circular(x, units = c("radians", "degrees", "hours"), type = NULL,
  template = NULL, modulo = NULL, zero = NULL, rotation = NULL)
```

Arguments

x	an object of class circular.
units	unit of the transformed data.
type	type of the transformed data. If NULL no action is performed.
template	template of the transformed data. If NULL no action is performed.
modulo	modulo of the transformed data. If NULL no action is performed.
zero	zero of the transformed data. If NULL no action is performed.
rotation	rotation of the transformed data. If NULL no action is performed.

Value

an object of class circular with the specified unit of measure, modulo, zero and rotation.

Author(s)

Claudio Agostinelli

See Also

deg and rad. If you want to set the properties of an object instead to transform it, you can use circular or circularp<-.

Examples

```
x <- rvonmises(n=10, mu=circular(0), kappa=9, control.circular=list(units="degrees"))
par(mfcol=c(2, 2))
plot(x)
y <- conversion.circular(x) # only the unit is changed (to radians) and
####### the data converted.
plot(y)
z <- conversion.circular(x, units="degrees", zero=pi) # only the zero is changed and
####### the data converted.
plot(z)
```

```
w <- conversion.circular(x, zero=pi, rotation="clock") # zero and rotation is
####### changed and the data converted.
plot(w)
```

Coope   *Coope dataset*

Description

A dataset taken from the paper of Coope (1993).

Usage

```
data(coope)
```

Format

`x.coope` and `y.coope` are vectors of length 8.

Source

Coope, I. (1993). Circle fitting by linear and non-linear least squares. Journal of Optimization Theory and Applications, 76, 381-388.

---

coord2rad   *Angles between a vector and the x-axis*

Description

From coordinates of the end point of a vector in 2 dimensions to the angle between this vector and the x-axis

Usage

```
coord2rad(x, y = NULL, control.circular = list())
```

Arguments

x   a `matrix` or a `data.frame` with two columns if y is NULL otherwise a vector.

y   a vector.

control.circular
                  the attribute of the resulting object.

Value

an object of class `circular`

Author(s)

Claudio Agostinelli and Frederick T. Wehrle

See Also

[circular](#)

Examples

```
set.seed(1234)
x <- cbind(rnorm(20), rnorm(20))
y <- coord2rad(x)
```

---

cor.circular   *Correlation Coefficient for Angular Variables*

Description

Computes a circular version of the Pearson's product moment correlation, and performs a significance test if requested.

Usage

```
cor.circular(x, y=NULL, test=FALSE)
```

Arguments

x	vector or matrix of circular data.
y	vector or matrix of circular data.
test	if test == TRUE, then a significance test for the correlation coefficient is computed.

Details

The correlation coefficient is computed like Pearson's product moment correlation for two linear variables X and Y. In the computational formula, however, (xi - xbar) and (yi - ybar) are replaced by sin(xi - xbar) and sin(yi - ybar), where xbar and ybar in the second two expressions are the mean directions of the samples.

Value

Returns a vector or a matrix of a circular version of the Pearson's product moment correlation, if test == TRUE then a list is reported with statistic and p.value, the test statistic and p-value respectively, for testing significance of the correlation coefficient.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 8.2, World Scientific Press, Singapore.

Jammalamadaka, S. and Sarma, Y. (1988). A correlation coefficient for angular variables. Statistical Theory and Data Analysis 2. North Holland: New York.

Examples

```
# Generate two circular data sets, and compute their correlation.
x <- rvonmises(n=50, mu=circular(0), kappa=3)
y <- x + rvonmises(n=50, mu=circular(pi), kappa=10)
cor.circular(x, y, test=TRUE)
```

---

curve.circular   *Draw Function Plots in a Circle*

Description

Draws a curve corresponding to the given function or expression (in x) over the interval [from,to] in a circle. Mainly used to plot circular density functions.

Usage

```
## S3 method for class 'circular'
curve(expr, from=NULL, to=NULL, n=101, add=FALSE,
  cex=1, axes=TRUE, ticks=FALSE, shrink=1, tcl=0.025,
  tcl.text=0.125, tol=0.04, uin=NULL, xlim=c(-1, 1),
  ylim=c(-1, 1), digits=2, modulo=c("2pi", "asis", "pi"),
  main=NULL, sub=NULL, xlab="", ylab="",
  control.circle=circle.control(), ...)
## S3 method for class 'function.circular'
plot(x, from=0, to=2*pi, ...)
```

Arguments

expr	an expression written as a function of x, or alternatively the name of a function which will be plotted.
x	a 'vectorizing' numeric R function.
from, to	the range over which the function will be plotted.
n	integer; the number of x values at which to evaluate.
add	logical; if TRUE add to already existing plot.
axes	logical: if TRUE axis are added to the plot.
ticks	logical: if TRUE tick - marks are added to the plot.
shrink	parameter that controls the size of the plotted circle. Default is 1. Larger values shrink the circle, while smaller values enlarge the circle.

tcl	length of the ticks.
tcl.text	The position of the axis labels.
tol	proportion of white space at the margins of plot.
uin	desired values for the units per inch parameter. If of length 1, the desired units per inch on the x axis.
xlim, ylim	the ranges to be encompassed by the x and y axes. Useful for centering the plot.
digits	number of digits used to print axis values.
modulo	the modulo used to process the data.

main, sub, xlab, ylab, cex

        graphical parameters.

control.circle	parameters passed to [plot.default](#) in order to draw the circle. The function [circle.control](#) is used to set the parameters.
...	parameters, passed to [lines.circular](#).

Details

For now, curve circular draws functions defined in radians, counterclockwise coordinate and zero at 0.

Value

A list with information on the plot: zero, rotation and next.points.

Author(s)

Claudio Agostinelli

See Also

[lines.circular](#) and [circle.control](#)

Examples

```
ff <- function(x) sqrt(x)/20
curve.circular(ff)
curve.circular(ff, to=6*pi, join=FALSE, nosort=TRUE, n=1001, modulo="asis",
  shrink=1.2)

plot.function.circular(function(x) dvonmises(x, circular(0), 10), xlim=c(-1, 2.2))
```

---

deg   *Degrees*

Description

Converts radians to degrees.

Usage

```
deg(x)
```

Arguments

x   vector or matrix of radian measurements.

Details

This function is available for compatibility with the CircStats package; please use `conversion.circular`.

Value

Returns a vector or matrix of degree measurements corresponding to the data in radians.

Author(s)

Claudio Agostinelli and Ulric Lund

See Also

`conversion.circular` and `rad`

---

density.circular   *Kernel Density Estimation for Circular Data*

Description

The function `density.circular` computes kernel density estimates with the given kernel and bandwidth for circular data.

Usage

```
## S3 method for class 'circular'
density(x, z=NULL, bw, adjust = 1, type = c("K", "L"),
  kernel = c("vonmises", "wrappednormal"), na.rm = FALSE,
  from = circular(0), to = circular(2 * pi), n = 512, K = NULL, min.k=10,
  control.circular=list(), ...)
## S3 method for class 'density.circular'
print(x, digits = NULL, ...)
```

Arguments

x	the data from which the estimate is to be computed. The object is coerced to class `circular`.
z	the points where the density is estimated. If `NULL` equally spaced points are used according to the parameters `from`, `to` and `n`.
bw	the smoothing bandwidth to be used. When the `kernel` is `vonmises` the bandwidth is equal to the concentration parameter.
adjust	the bandwidth used is actually `adjust*bw`. This makes it easy to specify values like "half the default bandwidth".
type	Not Yet Used.
kernel	a character string giving the smoothing kernel to be used. This must be one of `"vonmises"` or `"wrappednormal"`, that are kernels of type `"K"`.
na.rm	logical; if `TRUE`, missing values are removed from x. If `FALSE` any missing values cause an error.
from, to	the left and right-most points of the grid at which the density is to be estimated. The objects are coerced to class `circular`.
n	the number of equally spaced points at which the density is to be estimated.
K	number of terms to be used in approximating the density.
min.k	minimum number of terms used in approximating the density.
control.circular	
	the attribute of the resulting objects (x component).
digits	integer indicating the precision to be used.
...	further arguments passed to or from other methods.

Value

an object with class `"density.circular"` whose underlying structure is a list containing the following components.

data	original dataset.
x	the n coordinates of the points where the density is estimated. It is a circular objects with coordinate system setting using `control.circular`.
y	the estimated density values.
bw	the bandwidth used.
N	the sample size after elimination of missing values.
call	the call which produced the result.
data.name	the deparsed name of the x argument.
has.na	logical, for compatibility (always FALSE).

Author(s)

Claudio Agostinelli

References

Z.D. Bai and C.R. Rao and L.C. Zhao (1988). Kernel Estimators of Density Function of Directional Data, Journal of Multivariate Analysis, 27, 24-39.

J. Klemel\"a (2000). Estimation of densities and derivatives of densities with directional data, Journal of Multivariate Analysis, 73, 18-40.

V.R. Prayag and A.P. Gore (1990). Density Estimation for Randomly Distributed Circular Objects, Metrika, 1990, 37, 63-69.

P. Hall and G.S. Watson and J. Cabrera (1987). Kernel Density Estimation with Spherical Data, Biometrika, 74, 4, 751–762.

See Also

plot.density.circular and lines.density.circular

Examples

```
x <- rvonmises(n=100, mu=circular(pi), kappa=2)
res25 <- density(x, bw=25, control.circular=list(units="degrees"))
circularp(res25$x)
plot(res25, points.plot=TRUE, xlim=c(-1.6,1))
res50 <- density(x, bw=25, adjust=2)
lines(res50, col=2)
lines(res50, col=3, shrink=0.9) #shrink the plot wrt the function :-)
lines(res50, col=4, offset=0.5) #draw it with a reference circle of 0.5
```

---

dist.circular	*Distance Matrix Computation for Circular Data*

Description

This function computes and returns the distance matrix computed by using the specified distance measure to compute the distances between the rows of a data matrix containing circular data.

Usage

```
dist.circular(x, method = "correlation", diag = FALSE, upper = FALSE)
```

Arguments

x	a numeric matrix of class circular.
method	the distance measure to be used. This must be one of "correlation", "angularseparation", "chord", "geodesic". Any unambiguous substring can be given.
diag	logical value indicating whether the diagonal of the distance matrix should be printed by print.dist.
upper	logical value indicating whether the upper triangle of the distance matrix should be printed by print.dist.

Details

Available distance measures are (written for two vectors x and y):

correlation: $\sqrt{1-\rho}$ where ρ is the Circular Correlation coefficient defined as

$$\frac{\sum_{i=1}^{n} \sin(x_i - \mu_x) \sin(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n} \sin^2(x_i - \mu_x) \sum_{i=1}^{n} \sin^2(y_i - \mu_y)}}$$

and μ_x, μ_y are the mean direction of the two vectors

angularseparation: $\sum_{i=1}^{n} 1 - cos(x_i - y_i)$

chord: $\sum_{i=1}^{n} \sqrt{2(1 - \cos(x_i - y_i))}$

geodesic: $\sum_{i=1}^{n} \pi - |\pi - |x_i - y_i||$ where the abs(x - y) is expressed with an angle in [-pi,pi]

Missing values are allowed, and are excluded from all computations involving the rows within which they occur. Further, when Inf values are involved, all pairs of values are excluded when their contribution to the distance gave NaN or NA.
If some columns are excluded in calculating the sum is scaled up proportionally to the number of columns used. If all pairs are excluded when calculating a particular distance, the value is NA.

Value

dist.circular returns an object of class "dist".

The lower triangle of the distance matrix stored by columns in a vector, say do. If n is the number of observations, i.e., n <- attr(do, "Size"), then for $i < j <= n$, the dissimilarity between (row) i and j is do[n*(i-1) - i*(i-1)/2 + j-i]. The length of the vector is $n * (n - 1)/2$, i.e., of order n^2.

The object has the following attributes (besides "class" equal to "dist"):

Size	integer, the number of observations in the dataset.
Labels	optionally, contains the labels, if any, of the observations of the dataset.
Diag, Upper	logicals corresponding to the arguments diag and upper above, specifying how the object should be printed.
call	optionally, the [call](#) used to create the object.
method	optionally, the distance method used; resulting from [dist.circular](#)(), the ([match.arg](#)()ed) method argument.

See Also

[dist](#)

---

equal.kappa.test   *Equal Kappa Test*

Description

This function tests for the homogeneity of concentration parameters for multiple samples of directional data.

Usage

```
equal.kappa.test(x, group)
## S3 method for class 'equal.kappa.test'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	a vector of class `circular`.
group	a vector identifying the groups or samples.
digits	the number of digits to be printed.
...	additional arguments.

Details

The samples are assumed to have been drawn from von Mises populations. The null hypothesis tested is that all populations sampled have the same concentration parameter, kappa.

When the pooled data has high concentration, sample mean resultant length above 0.70, Bartlett's test is used. For less concentrated pooled data, variance-stabilizing transformations are used to improve normal approximations needed to arrive at an approximate chi-squared test statistic (see references below). For pooled sample mean resultant length below 0.45, it is possible that individually a sample may in fact have quite a large sample mean resultant length. In this case, it is possible that the variance-stabilizing transformation involving the inverse sine function is passed a value outside of -1,1. If this occurs, the function will automatically use Bartlett's test and issue a warning to that effect.

Value

An object of class `equal.kappa.test` with the following components:

kappa	concentration parameter for each sample.
kappa.all	concentration parameter of all samples combined.
rho	mean resultant length for each sample.
rho.all	mean resultant length of all samples combined.
df	degrees of freedom for chi-squared distribution.
statistic	the value of the chi-squared test statistic.
p.value	the p.value of the test statistic.
call	the `match.call` result.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 5.3, World Scientific Press, Singapore.

Mardia, K. and Jupp, P. (1999). Directional Statistics, Section 7.4, John Wiley and Sons, England.

Examples

```
x <- c(rvonmises(50, circular(0), 1), rvonmises(100, circular(pi/3), 10))
group <- c(rep(0, 50), rep(1, 100))

equal.kappa.test(x, group)
```

---

fisherB1   *B.1 Arrival times at an intensive care unit*

Description

Arrival time on a 24-hour clock of 254 patients at an intensive care unit, over a period of about 12 months.

Usage

```
data(fisherB1)
data(fisherB1c)
```

Format

`fisherB1` is a vector of 254 observations (in the format hours.minutes). `fisherB1c` contains the same observations in a circular objects (minutes are expressed as decimals).

Source

Cox, D.R. and Lewis, P.A.W. (1966) The Statistical Analysis of Series of Events. London : Methuen & CO. Ltd. pp. 254-255

See Also

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press. Pag. 239.

Examples

```
data(fisherB1c)
par(mfcol=c(1,2))
plot(fisherB1c, main="Clock 24", shrink=1.5)
plot(fisherB1c, template="clock12", main="Clock 12", shrink=1.5)
```

---

fisherB10   *B.10 Directions of desert ants*

Description

Directions of 11 long-legged desert ants (Cataglyphis fortis) after one eye on each ant was 'trained'
to learn the ant's home direction, then covered and the other eye uncovered.

Usage

```
data(fisherB10)
data(fisherB10c)
```

Format

fisherB10 is a list (in degrees). fisherB10c contains the same observations in a circular objects.

Source

Personal communication of Prof. Dr. R. Wehner to Prof. N.I. Fisher, experiment described in

R. Wehner and M. Muller (1985) Does interocular transfer occur in visual navigation by ants?
Nature, 315, 228-9.

See Also

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press. Pag. 244-245.

Examples

```
data(fisherB10c)
res <- plot(fisherB10c$set1)
points(fisherB10c$set2, col=2, plot.info=res)
points(fisherB10c$set3, col=3, plot.info=res)
```

---

fisherB11   *B.11 Movements of sea stars*

Description

Resultant directions of 22 sea stars 11 days after being displaced from their natural habitat.

Usage

```
data(fisherB11)
data(fisherB11c)
```

Format

fisherB11 a vector of 22 observations (in degrees). fisherB11c contains the same observations in a circular objects.

Source

G.J.G. Upton and B. Fingleton (1989) Spatial Data Analysis by Example. Volume 2. Categorical and Directional Data. New York: John Wiley as adapted from B. Pabst and H. Vicentini (1978) Dislocation experiments in the migrating seastar. Astropecten jonstoni. Marine Biology 48, 271-8.

See Also

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press. Pag. 245.

Examples

```
data(fisherB11c)
plot(fisherB11c, stack=TRUE, shrink=1.5)
```

---

fisherB12   *B.12: Vanishing directions of homing pigeons*

Description

Vanishing directions of 15 homing pigeons, released just over 16 kilometres Northwest of their loft.

Usage

```
data(fisherB12)
data(fisherB12c)
```

Format

fisherB12 a vector of 15 observations (in degrees). fisherB12c contains the same observations in a circular objects.

Source

Schmidt-Koenig (1963) On the role of the loft, the distance and site of release in pigeon homing (the "cross-loft experiment"). Biol. Bull. (125)154-164.

References

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press. Pag. 245.

Examples

```
data(fisherB12c)
plot(fisherB12c, stack=TRUE, shrink=1.5)
```

---

  fisherB13   *B.13: Orientations of termite mounds*

Description

Orientations of termite mounds of Amitermes laurensis at 14 sites in Cape York Penisula, North Queensland.

Usage

```
data(fisherB13)
data(fisherB13c)
```

Format

fisherB13 a list of 14 datasets (axes in degrees) at several locations. fisherB13c contains the same observations in a circular objects.

Details

Set 1: n=100, Latitude -15'43", Longitude 144'42" Set 2: n=50, Latitude -15'32", Longitude 144'17"

Source

A.V. Spain, T. Okello-Oloya and R.D. John (1983) Orientation of the termitaria of two species of Amitermes (Isoptera:Termitinae) from Northern Queensland. Aust. J. Zoo. (31):167-177.

References

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press. Pag. 246.

Examples

```
data(fisherB13c)
plot(fisherB13c$set1, stack=TRUE, shrink=1.5)
```

---

fisherB18   *B.18 Wind direction and ozone concentration.*

Description

19 measurements of wind direction 'theta' and ozone level 'x' taken at 6.00am at four-day intervals between April 18th and June 29th, 1975 at a weather station in Milwaukee.

Usage

```
data(fisherB18)
data(fisherB18c)
```

Format

fisherB18 is a data.frame of integer value. fisherB18c is a data.frame that contains the same observations, but in the first column, the data is a circular object.

Source

N.I. Fisher (1993) pag. 251. Johnson & Wehrly (1977, Table 1).

References

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press.

Examples

```
data(fisherB18)
data(fisherB18c)
par(mfcol=c(1,3))
plot(fisherB18c$theta, xlab=expression(theta))
boxplot(fisherB18c$x, xlab="x")
plot(c(fisherB18$x, fisherB18$x), c(fisherB18$theta,
  fisherB18$theta+360), xlab="x", ylab=expression(theta))
```

| fisherB2 | *B.2 Measurements of long-axis orientation of 133 feldspar laths in basalt* |

Description

Measurements of long-axis orientation of 133 feldspar laths in basalt

Usage

```
data(fisherB2)
data(fisherB2c)
```

Format

`fisherB2` is a vector of 133 observations (in degrees). `fisherB2c` contains the same observations in a circular objects.

Source

Smith, N.M. (1988) Reconstruction of the Tertiary drainage systems of the Inverell region. Unpublished B.Sc. (Hons.) thesis, Department of Geography, University of Sydney, Australia.

This dataset (set 28-6-1co.prn) was kindly supplied by Ms Nicola Smith to Prof. N.I. Fisher.

See Also

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press. Pag. 240.

Examples

```
data(fisherB2c)
plot(fisherB2c)
```

| fisherB20 | *B.20 Movements of blue periwinkles.* |

Description

Distances 'x' and directions 'theta' by small blue periwinkles, Nodilittorina unifasciata, after they had been transplanted downshore from the height at which they normally live.

Usage

```
data(fisherB20)
data(fisherB20c)
```

Format

fisherB20 is a [data.frame](#) of integer value. fisherB20c is a [data.frame](#) that contains the same observations, but in the first column, the data is a [circular](#) object.

Source

N.I. Fisher (1993) pag. 252-253. Data kindly supplied by Dr A. Underwood and Ms G. Chapman.

References

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press.

Examples

```
data(fisherB20)
data(fisherB20c)
par(mfcol=c(1,3))
plot(fisherB20c$theta, xlab=expression(theta))
boxplot(fisherB20c$x, xlab="x")
plot(c(fisherB20$x, fisherB20$x), c(fisherB20$theta,
  fisherB20$theta+360), xlab="x", ylab=expression(theta))
```

---

fisherB3   *B.3 Movements of turtles*

Description

Measurements of the directions taken by 76 turtles after treatment.

Usage

```
data(fisherB3)
data(fisherB3c)
```

Format

fisherB3 is a vector of 76 observations (in degrees). fisherB3c contains the same observations in a circular objects.

Source

Stephens, M.A. (1969) Techniques for directional data. Technical Report #150, Department of Statistics, Stanford University, Stanford, CA.

See Also

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press. Pag. 241.

Examples

```
data(fisherB3c)
plot(fisherB3c)
```

---

fisherB4   *B.4 Directional preferences of starhead topminnows*

Description

Sun compass orientations of 50 starhead topminnows, measured under heavily overcast conditions.

Usage

```
data(fisherB4)
data(fisherB4c)
```

Format

`fisherB4` is a vector of 50 observations (in degrees). `fisherB4c` contains the same observations in a circular objects.

Source

Goodyear (1970) Terrestrial and aquatic orientation in the Starhead Topminnow, Fundulus notti. Science 168, 603-5. Figure 1D.

See Also

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press. Pag. 241.

Examples

```
data(fisherB3c)
plot(fisherB3c)
```

---

fisherB5   *B.5 Measurements of long-axis orientation of 164 feldspar laths in basalt*

Description

Measurements of long-axis orientation of 164 feldspar laths in basalt

Usage

```
data(fisherB5)
data(fisherB5c)
```

Format

`fisherB5` is a vector of 164 observations (in degrees). `fisherB5c` contains the same observations in a circular objects.

Source

Smith, N.M. (1988) Reconstruction of the Tertiary drainage systems of the Inverell region. Unpublished B.Sc. (Hons.) thesis, Department of Geography, University of Sydney, Australia.

This dataset (set 24-6-5co.prn) was kindly supplied by Ms Nicola Smith to Prof. N.I. Fisher.

See Also

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press. Pag. 242.

Examples

```
data(fisherB5c)
plot(fisherB5c)
```

---

fisherB6   *B.6 Cross-bed azimuths of palaeocurrents*

Description

Set of cross-bed azimuths of palaeocurrents measured in the Belford Anticline (New South Wales).

Usage

```
data(fisherB6)
data(fisherB6c)
```

Format

fisherB6 is a list (in degrees). fisherB6c contains the same observations in a circular objects.

Source

Fisher, N.I. & Powell C. McA. (1989) Statistical analysis of two-dimensional palaeocurrent data: Methods and examples. Aust. J. Earth Sci. 36, 91-107.

See Also

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press. Pag. 242.

Examples

```
data(fisherB6c)
res <- plot(fisherB6c$set1)
points(fisherB6c$set2, col=2, plot.info=res)
points(fisherB6c$set3, col=3, plot.info=res)
```

---

fisherB7   *B.7 Movements of ants*

Description

Directions chosen by 100 ants in response to an evenly illuminated black targets placed as shown.

Usage

```
data(fisherB7)
data(fisherB7c)
```

Format

fisherB7 a vector of 100 observations (in degrees). fisherB7c contains the same observations in a circular objects.

Source

Randomly selected values from Jander, R. (1957) Die optische Richtangsorientierung der roten Waldameise (Formica rufa. L.) Z. vergl. Physiologie 40, 162-238. Figure 18A.

See Also

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press. Pag. 243.

Examples

```
data(fisherB7c)
plot(fisherB7c, zero=pi/2, rotation='clock', stack=TRUE)
```

---

fisherB8   *B.8 Orientations of pebbles*

Description

Horizontal axes of 100 outwash pebbles fromo a late Wisconsin outwash terrace along Fox river, near Cary, Illinois

Usage

```
data(fisherB8)
data(fisherB8c)
```

Format

`fisherB8` a vector of 100 observations (in degrees). `fisherB8c` contains the same observations in a circular objects.

Source

Mardia, K.V. (1972) Statistics of Directional Data. London: Academic Press. Table 1.6 adapted from Krumbein W.C. (1939) Preferred orientations of pebbles in sedimentary deposits. J. Geol. 47, 673-706.

See Also

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press. Pag. 243.

Examples

```
data(fisherB8c)
plot(fisherB8c, stack=TRUE, shrink=1.5)
```

---

fisherB9   *B.9 Dance directions of bees*

Description

Dance directions of 279 honey bees viewing a zenith patch of artificially polarised light.

Usage

```
data(fisherB9)
data(fisherB9c)
```

Format

fisherB9 a vector of 279 observations (in degrees). fisherB9c contains the same observations in a circular objects.

Source

Adapted by Prof. N.I. Fisher from R. Wehner & S. Strasser (1985) The POL area of the honey bee's eye: behavioural evidence. Physiol. Entomol. 10, 337-49. Pag. 346.

See Also

N.I. Fisher (1993) Statistical analysis of circular data. Cambridge University Press. Pag. 244.

Examples

```
data(fisherB9c)
plot(fisherB9c, stack=TRUE, shrink=1.5)
```

GenVonMises   *Generalized Von Mises Density Function*

Description

Density for the Generalized von Mises circular distribution.

Usage

```
dgenvonmises(x, mu1, mu2, kappa1, kappa2)
```

Arguments

x	a vector. The object is coerced to class `circular`.
mu1	principal direction of the distribution. The object is coerced to class `circular`.
mu2	secondary direction parameter. The object is coerced to class `circular`.
kappa1	non-negative numeric parameter of the distribution.
kappa2	non-negative numeric parameter of the distribution.

Details

The Generalized von Mises distribution has density

$$f(x) = \frac{1}{2\pi G_0(\delta, \kappa_1, \kappa_2)} \exp\{\kappa_1 \cos(x - \mu_1) + \kappa_2 \cos 2(x - \mu_2)\},$$

for $0 \le x < 2\pi$, where $\delta = (\mu_1 - \mu_2)$ and G_0 is the normalizing constant.

Value

The density

Author(s)

Federico Rotolo

References

Gatto , R. & Jammalamadaka , S.R. (2007). The generalized von Mises distribution. Statistical Methodology 4, 341-353.

Examples

```
ff <- function(x) dgenvonmises(x, mu1=circular(5*pi/4), mu2=circular(pi/4), kappa1=.3, kappa2=1)
curve.circular(ff, join=TRUE, xlim=c(-1, 1), ylim=c(-1.2, 1.2),
  main="Density of a Generalized von Mises Distribution",
  xlab=expression(paste(mu,"1=5/4",pi,", ",mu2,"=",pi/4,", ",kappa,"1=0.3, ",kappa,"2=1"))
  )
```

---

heatmap.circular   *Draw a Heat Map for circular data*

Description

A heat map is a false color image (basically image(t(x))) with a dendrogram added to the left side and to the top. Typically, reordering of the rows and columns according to some set of values (row or column means) within the restrictions imposed by the dendrogram is carried out. See also heatmap.

Usage

```
heatmap.circular(x, Rowv = NULL, Colv = if (symm) "Rowv" else NULL,
distfun = dist.circular, hclustfun = hclust,
reorderfun = function(d, w) reorder(d, w), add.expr, symm = FALSE,
revC = identical(Colv, "Rowv"), na.rm = TRUE, margins = c(5, 5),
lwid = c(1, 4), lhei = c(1, 4), ColSideColors, RowSideColors,
NAColors = "black", cexRow = 0.2 + 1/log10(nr), cexCol = 0.2 + 1/log10(nc),
labRow = NULL, labCol = NULL, main = NULL, xlab = NULL, ylab = NULL,
keep.dendro = FALSE, annotate.expr, annotate = rep(NA, 4),
verbose = getOption("verbose"), ...)
```

Arguments

x	numeric matrix of class [circular](#) of the values to be plotted.
Rowv	determines if and how the *row* dendrogram should be computed and reordered. Either a [dendrogram](#) or a vector of values used to reorder the row dendrogram or [NA](#) to suppress any row dendrogram (and reordering) or by default, [NULL](#), see 'Details' below.
Colv	determines if and how the *column* dendrogram should be reordered. Has the same options as the Rowv argument above and *additionally* when x is a square matrix, Colv = "Rowv" means that columns should be treated identically to the rows (and so if there is to be no row dendrogram there will not be a column one either).
distfun	function used to compute the distance (dissimilarity) between both rows and columns. Defaults to [dist.circular](#).
hclustfun	function used to compute the hierarchical clustering when Rowv or Colv are not dendrograms. Defaults to [hclust](#). Should take as argument a result of distfun and return an object to which [as.dendrogram](#) can be applied.
reorderfun	function(d,w) of dendrogram and weights for reordering the row and column dendrograms. The default uses [reorder.dendrogram](#).
add.expr	expression that will be evaluated after the call to image. Can be used to add components to the plot.
symm	logical indicating if x should be treated **symm**etrically; can only be true when x is a square matrix.
revC	logical indicating if the column order should be [rev](#)ersed for plotting, such that e.g., for the symmetric case, the symmetry axis is as usual.
na.rm	logical indicating whether NA's should be removed.
margins	numeric vector of length 2 containing the margins (see [par](#)(mar= *)) for column and row names, respectively.
lwid	a vector of values for the widths of columns on the device. Relative widths are specified with numeric values. Absolute widths (in centimetres) are specified with the [lcm](#)() function (see [layout](#)).
lhei	a vector of values for the heights of rows on the device. Relative and absolute heights can be specified, see lwid above.
ColSideColors	(optional) character vector of length ncol(x) containing the color names for a horizontal side bar that may be used to annotate the columns of x.
RowSideColors	(optional) character vector of length nrow(x) containing the color names for a vertical side bar that may be used to annotate the rows of x.
NAColors	the color used to plot missing values.
cexRow, cexCol	positive numbers, used as cex.axis in for the row or column axis labeling. The defaults currently only use number of rows or columns, respectively.
labRow, labCol	character vectors with row and column labels to use; these default to rownames(x) or colnames(x), respectively.
main, xlab, ylab	main, x- and y-axis titles; defaults to none.

keep.dendro   logical indicating if the dendrogram(s) should be kept as part of the result (when
                 Rowv and/or Colv are not NA).

annotate   annotation in the four external side of the figure. A positive value in a position
                 means you want annotate something in that position (1=bottom, 2=left, 3=top,
                 4=right). For instance, annotate=c(0.1, NA, NA, 1, 1) means you want to
                 annotate one thing on the bottom with dimension 0.1 and two things on right
                 each with dimension 1.

annotate.expr   must be a list of expressions with the same length as annotate. For instance for
                 annotate=c(0.1, NA, NA, 1, 1) you must have something as annotate.expr=list(expr1,
                 NA, NA, expr2, expr2) where expr1 etc. must be a valid R expression able to
                 produce a plot.

verbose   logical indicating if information should be printed.

...   additional arguments passed on to image, e.g., col specifying the colors.

Details

If either Rowv or Colv are dendrograms they are honored (and not reordered). Otherwise, dendro-
grams are computed as dd <- as.dendrogram(hclustfun(distfun(X))) where X is either x or
t(x).

If either is a vector (of 'weights') then the appropriate dendrogram is reordered according to the
supplied values subject to the constraints imposed by the dendrogram, by reorder(dd, Rowv), in
the row case. If either is missing, as by default, then the ordering of the corresponding dendrogram
is by the mean direction value of the rows/columns, i.e., in the case of rows, Rowv <- rowMeans(x,
na.rm=na.rm). If either is NULL, *no reordering* will be done for the corresponding side.

Unless Rowv = NA (or Colw = NA), the original rows and columns are reordered *in any case* to
match the dendrogram, e.g., the rows by order.dendrogram(Rowv) where Rowv is the (possibly
reorder()ed) row dendrogram.

heatmap() uses layout and draws the image in the lower right corner of a 2x2 layout. Consequen-
tially, it can **not** be used in a multi column/row layout, i.e., when par(mfrow= *) or (mfcol= *)
has been called.

Value

par(mfrow= *) or (mfcol= *) has been called.

Author(s)

Claudio Agostinelli using the code from heatmap.

See Also

dist.circular, heatmap, image, hclust

I.0 *Zeroth Order Bessel Function of the First Kind*

Description

An alias of besselI(x, nu=0).

Usage

I.0(x)

Arguments

x   non-negative numerical value at which to evaluate the Bessel function.

Value

Returns the zeroth order Bessel function of the first kind evaluated at a specified real number.

See Also

besselI.

I.1 *First Order Bessel Function of the First Kind*

Description

An alias of besselI(x, nu=1).

Usage

I.1(x)

Arguments

x   non-negative numerical value at which to evaluate the Bessel function.

Value

Returns the first order Bessel function of the first kind, evaluated at a specified real number.

See Also

besselI.

I.p   *P-th Order Bessel Function of the First Kind*

Description

An alias of besselI(x, nu=p).

Usage

```
I.p(p, x)
```

Arguments

p	positive integer order of the Bessel function.
x	non-negative numerical value at which to evaluate the Bessel function.

Value

Returns the p-th order Bessel function of the first kind, evaluated at a specified real number.

See Also

[besselI](#).

---

intersect.modal.region

Intersection between model region and a given interval.

Description

Find an estimates of the probability of the intersection between a modal region and a given interval.

Usage

```
intersect.modal.region(x, ...)
## Default S3 method:
intersect.modal.region(x, ...)
## S3 method for class 'circular'
intersect.modal.region(x, breaks, z = NULL, q = 0.95, bw,
  adjust = 1, type = c("K", "L"), kernel = c("vonmises", "wrappednormal"),
  na.rm = FALSE, step = 0.01, eps.lower = 10^(-4), eps.upper = 10^(-4), ...)
```

Arguments

x	numeric or an object of class [circular]().
breaks	a matrix with two columns. Each row specifies a sub-interval.
z	numeric or object of class [circular](). The grid were the kernel density estimate will be evaluated. If NULL equally spaced points in the interval [0,2*pi) with step step.
q	numeric in the interval [0,1]. The quantile of the modal region.
bw	the smoothing bandwidth to be used. When the kernel is vonmises the bandwidth is equal to the concentration parameter.
adjust	the bandwidth used is actually adjust*bw. This makes it easy to specify values like "half the default bandwidth".
type	Not Yet Used.
kernel	a character string giving the smoothing kernel to be used. This must be one of "vonmises" or "wrappednormal", that are kernels of type "K".
na.rm	logical; if TRUE, missing values are removed from x. If FALSE any missing values cause an error.
step	numeric. Used in the construction of the regular grid z.
eps.lower, eps.upper	
	the cut point in the density is searched in the interval [min(density)*(1+eps.lower),max(density)*(1-eps.upper)].
...	further arguments passed to the next methods.

Details

Only the version for circular data is actually implemented.

Value

For the circular method a list with the following three components

tot	the total area.
areas	information for each subinterval.
breaks	the extremes of each subinterval.

Author(s)

Claudio Agostinelli

See Also

[modal.region]()

Examples

```
x <- rvonmises(100, circular(pi), 10)
res <- intersect.modal.region(x, breaks=circular(matrix(c(pi,pi+pi/12,
  pi-pi/12, pi), ncol=2, byrow=TRUE)), bw=50)
res$tot

x <- rvonmises(100, circular(0), 10)
res <- intersect.modal.region(x, breaks=circular(matrix(c(pi,pi+pi/12),
  ncol=2)), bw=50)
res$tot

res <- intersect.modal.region(x, breaks=circular(matrix(c(pi/12,
  2*pi-pi/12), ncol=2, byrow=TRUE)), bw=50)
res$tot
```

JonesPewsey   *Jones and Pewsey Density Function*

Description

Density for the Jones and Pewsey circular distribution.

Usage

```
djonespewsey(x, mu, kappa, psi)
```

Arguments

x	a vector. The object is coerced to class [circular]().
mu	direction parameter of the distribution. The object is coerced to class [circular]().
kappa	non-negative concentration parameter of the distribution.
psi	real shape parameter.

Details

The JonesPewsey distribution has density

$$f(x) = \frac{(\cosh(\kappa\psi) + \sinh(\kappa\psi)\cos(x - \mu))^{1/\psi}}{2\pi P_{1/\psi}(\cosh(\kappa\psi))},$$

for $0 \leq x < 2\pi$, where $P_{1/\psi}(\cdot)$ is the associated Legendre function of the first kind, degree $1/\psi$ and order 0.

Value

The density

Author(s)

Federico Rotolo

References

Jones , M.C. and Pewsey, A. (2005). A family of symmetric distributions on the circle. J. Am. Statist. Assoc. 100, 1422-1428

Examples

```
ff <- function(x) djonespewsey(x, mu=circular(4), kappa=1.8, psi=-.6)
curve.circular(ff, join=TRUE, xlim=c(-1, 1), ylim=c(-1.2, 1.2),
  main="Density of a JonesPewsey Distribution",
  xlab=expression(paste(mu,"=1.3",pi,", ",kappa,"=1.8, ",psi,"=-0.6"))
  )
```

KatoJones   *Kato and Jones Density Function*

Description

Density and random generation for the Kato and Jones distribution.

Usage

```
rkatojones(n, mu, nu, r, kappa, control.circular=list())
dkatojones(x, mu, nu, r, kappa)
```

Arguments

x	the angular value the density must be computed in.
n	number of observations.
mu	the Mobius 'mu' parameter. The object is coerced to class [circular].
nu	the Mobius 'nu' parameter. The object is coerced to class [circular].
r	the Mobius 'r' parameter. It must be in [0,1).
kappa	the positive vonMises parameter.
control.circular	
	the attribute of the resulting object.

Details

The Kato and Jones distribution has density

$$f(x) = \frac{1-r^2}{2\pi \mathcal{I}_0(\kappa)} \exp\left[\frac{\kappa\{\xi\cos(x-\eta) - 2r\cos\nu\}}{1+r^2-2r\cos(x-\gamma)}\right] \quad \times \frac{1}{1+r^2-2r\cos(x-\gamma)},$$

for $0 \le x < 2\pi$, where $\gamma = \mu + \nu$, $\xi = \{r^4 + 2r^2\cos(2\nu) + 1\}^{1/2}$ and $\eta = \mu + \arg[r^2\{\cos(2\nu) + i\sin(2\nu)\} + 1]$.

Original code for random generation is by Kato, S. and Jones, M.C. and can be found at the address http://pubs.amstat.org/doi/suppl/10.1198/jasa.2009.tm08313/suppl_file/t08-313code.txt.

Value

The density. dkatojones gives the density and rkatojones generates random deviates.

Author(s)

Federico Rotolo

References

Kato , S. and Jones, M.C. (2010). A family of distributions on the circle with links to, and applications arising from, Mobius transformation. J. Am. Statist. Assoc. 105, 249-262.

Examples

```
data1 <- rkatojones(n=100, mu=circular(0), nu=circular(pi/4), r=.2, kappa=1)
plot(data1)

data1 <- rkatojones(n=100, mu=circular(pi/3), nu=circular(pi), r=.7, kappa=2.3)
plot(data1)

ff <- function(x) dkatojones(x, mu=circular(pi/3), nu=circular(pi), r=.7, kappa=2.3)
curve.circular(ff, join=TRUE, xlim=c(-1, 1), ylim=c(-1.2, 1.2),
  main="Density of a KatoJones Distribution",
  xlab=expression(paste(mu,"=",pi,"/3, ",nu,"=",pi,", r=0.7, ",kappa,"=2.3"))
  )
```

---

kuiper.test   *Kuiper's Test*

Description

Performs Kuiper's one-sample test of uniformity on the circle.

Usage

```
kuiper.test(x, alpha=0)
## S3 method for class 'kuiper.test'
print(x, digits = 4, ...)
```

Arguments

x	a vector. The object is coerced to class `circular`.
alpha	significance level of the test. Possible levels are 0.15, 0.1, 0.05, 0.025, 0.01. Alpha may be omitted or set to zero, in which case a range for the p-value of the test will be printed.
digits	integer indicating the precision to be used.
...	further arguments passed to or from other methods.

Details

Kuiper's test statistic is a rotation-invariant Kolmogorov-type test statistic. The critical values of a modified Kuiper's test statistic are used according to the tabulation given in Stephens (1970).

Value

A list with the statistic and alpha value.

Note

Kuiper's one-sample test of uniformity is performed, and the results are printed to the screen. If alpha is specified and non-zero, the test statistic is printed along with the critical value and decision. If alpha is omitted, the test statistic is printed and a range for the p-value of the test is given.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 7.2, World Scientific Press, Singapore.

Stephens, M. (1970). Use of the Kolmogorov-Smirnov, Cramer-von Mises and related statistics without extensive tables. Journal of the Royal Statistical Society, B32, 115-122.

See Also

`range.circular`, `rao.spacing.test`, `rayleigh.test` and `watson.test`

Examples

```
# Generate data from the uniform distribution on the circle.
data <- circular(runif(100, 0, 2*pi))
kuiper.test(data)
# Generate data from the von Mises distribution.
data <- rvonmises(n=100, mu=circular(0), kappa=3)
kuiper.test(data, alpha=0.01)
```

---

lines.circular   *Add Connected Line Segments to a Circular Plot*

Description

A method taking coordinates in a polar system and joining the corresponding points with line segments.

Usage

```
## S3 method for class 'circular'
lines(x, y, join = FALSE, nosort = FALSE, offset=1, shrink=1,
  plot.info = NULL, zero = NULL, rotation = NULL, modulo = NULL, ...)
```

Arguments

x	a vector of class `circular`.
y	a vector with the same length as 'x'.
join	logical: if `TRUE` the first and the last values are joined by a line.
nosort	logical: if `TRUE` the data are not sorted before join them.
offset	the radius of the circle
shrink	parameter that controls the size of the plotted function. Default is 1.
plot.info	an object from another circular graphic function.
zero	the zero of the axis.
rotation	the rotation of the axis.
modulo	the modulo applied to 'x' before sorting.
...	graphical parameters passed to [lines.default]().

Value

A list with information on the plot: zero, rotation and next.points.

Author(s)

Claudio Agostinelli

See Also

[plot.circular]()

Examples

```
x <- rvonmises(20, circular(0), 10)
y <- runif(20, 0.5, 1)

plot(x, shrink=2)
lines(x, y)
```

```
lines.density.circular
```
Add a Plot for Kernel Density Estimation for Circular Data

Description

The `lines` add a plot for `density.circular` objects.

Usage

```
## S3 method for class 'density.circular'
lines(x, type = "l", zero.line = TRUE,
  points.plot = FALSE, points.col = 1, points.pch = 1, points.cex = 1,
  plot.type = c("circle", "line"), bins = NULL, offset=1, shrink = 1,
  tcl = 0.025, sep = 0.025, join = TRUE, nosort = FALSE,
  plot.info = NULL, zero = NULL, rotation = NULL, ...)
```

Arguments

x	an object of class [density.circular]().
type	plotting parameter with useful default.
zero.line	logical; if TRUE, add a base line at $y = 0$. Used when `plot.type` is "line".
points.plot	logical; if TRUE original data are added to the plot.
points.col, points.pch, points.cex	
	parameters used to draw the points.
plot.type	type of the plot.
bins	number of ticks to plot.
offset	the radius of the circle
shrink	parameter that controls the size of the plotted function. Default is 1.
tcl	length of the ticks.
sep	constant used to specify the distance between stacked points. Default is 0.025; smaller values will create smaller spaces.
join	logical: should the first and the last point joined.
nosort	logical: should the data sort before plotting. Defaults is to sort.

plot.info	an object from `plot.circular` that contains information on the `zero`, the `rotation` and `next.points`.
zero	the zero of the plot. Ignored if `plot.info` is provided.
rotation	the rotation of the plot. Ignored if `plot.info` is provided.
...	further parameters passed to `lines.default`.

Value

A list with information on the plot: zero, rotation and next.points and, if available, the coordinates x and y.

Author(s)

Claudio Agostinelli

See Also

`density.circular` and `plot.density.circular`

Examples

```
set.seed(1234)
x <- rvonmises(n=100, mu=circular(pi), kappa=2)
y <- rvonmises(n=100, mu=circular(pi/2), kappa=2)
resx <- density(x, bw=25)
res <- plot(resx, points.plot=TRUE, xlim=c(-1.5,1), ylim=c(-1.1, 1.5))
resy <- density(y, bw=25)
lines(resy, points.plot=TRUE, col=2, points.col=2, plot.info=res)
```

---

lm.circular   *Circular-Circular and Circular-Linear Regression*

Description

Fits a regression model for a circular dependent and circular independent variable or for a circular dependent and linear independent variables.

Usage

```
lm.circular(..., type=c("c-c", "c-l"))
lm.circular.cc(y, x, order = 1, level = 0.05, control.circular = list())
lm.circular.cl(y, x, init = NULL, verbose = FALSE, tol = 1e-10,
  control.circular = list())
## S3 method for class 'lm.circular.cl'
print(x, digits = max(3, getOption("digits") - 3),
  signif.stars= getOption("show.signif.stars"), ...)
```

Arguments

...	arguments passed to `lm.circular.cc` or to `lm.circular.cl` depending on the value of `type`.
type	if `type=="c-c"` then `lm.circular.cc` is called otherwise `lm.circular.cl` is called.
y	vector of data for the dependent circular variable.
x	vector of data for the independent circular variable if `type="c-c"` or `lm.circular.cc` is used otherwise a matrix or a vector containing the independent linear variables.
order	order of trigonometric polynomial to be fit. Order must be an integer value. By default, order=1. Used if `type="c-c"`.
level	level of the test for the significance of higher order trigonometric terms. Used if `type="c-c"`.
control.circular	
	the attribute of the resulting objects (`fitted`, `residuals` components in the case of `type=="c-c"` and `mu` and `se.mu`) otherwise.
init	a vector with initial values of length equal to the columns of `x`.
verbose	logical: if TRUE messages are printed while the function is running.
tol	the absolute accuracy to be used to achieve convergence of the algorithm.
digits	the number of digits to be printed.
signif.stars	logical; if TRUE, P-values are additionally encoded visually as "significance stars" in order to help scanning of long coefficient tables. It defaults to the show.signif.stars slot of [options]().

Details

If `type=="c-c"` or `lm.circular.cc` is called directly a trigonometric polynomial of x is fit against the cosine and sine of y. The order of trigonometric polynomial is specified by order. Fitted values of y are obtained by taking the inverse tangent of the predicted values of sin(y) divided by the predicted values of cos(y). Details of the regression model can be found in Sarma and Jammalamadaka (1993).

If `type=="c-l"` or `lm.circular.cl` is called directly, this function implements the homoscedastic version of the maximum likelihood regression model proposed by Fisher and Lee (1992). The model assumes that a circular response variable theta has a von Mises distribution with concentration parameter kappa, and mean direction related to a vector of linear predictor variables according to the relationship: mu + 2*atan(beta'*x), where mu and beta are unknown parameters, beta being a vector of regression coefficients. The function uses Green's (1984) iteratively reweighted least squares algorithm to perform the maximum likelihood estimation of kappa, mu, and beta. Standard errors of the estimates of kappa, mu, and beta are estimated via large-sample asymptotic variances using the information matrix. An estimated circular standard error of the estimate of mu is then obtained according to Fisher and Lewis (1983, Example 1).

Value

If type=="c-c" or lm.circular.cc is called directly an object of class lm.circular.cc is returned with the following components:

call	the match.call result.
rho	square root of the average of the squares of the estimated conditional concentration parameters of y given x.
fitted	fitted values of the model of class circular.
data	matrix whose columns correspond to x and y.
residuals	circular residuals of the model of class circular.
coefficients	matrix whose entries are the estimated coefficients of the model. The first column corresponds to the coefficients of the model predicting the cosine of y, while the second column contains the estimates for the model predicting the sine of y. The rows of the matrix correspond to the coefficients according to increasing trigonometric order.
p.values	p-values testing whether the (order + 1) trigonometric terms are significantly different from zero.
A.k	is mean of the cosines of the circular residuals.
kappa	assuming the circular residuals come from a von Mises distribution, kappa is the MLE of the concentration parameter.

If type=="c-l" or lm.circular.cl is called directly an object of class lm.circular.cc is returned with the following components:

call	the match.call result.
x	the independent variables.
y	the dependent variable.
mu	the circular mean of the dependent variable of class circular.
se.mu	an estimated standard error of the circular mean with the same units of measure used for mu.
kappa	the concentration parameter for the dependent variable.
se.kappa	an estimated standard error of the concentration parameter.
coefficients	the estimated coefficients.
cov.coef	covariance matrix of the estimated coefficients.
se.coef	standard errors of the estimated coefficients.
log.lik	log-likelihood.
t.values	values of the t statistics for the coefficients.
p.values	p-values of the t statistics. Approximated values using Normal distribution.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Fisher, N. and Lee, A. (1992). Regression models for an angular response. Biometrics, 48, 665-677.

Fisher, N. and Lewis, T. (1983). Estimating the common mean direction of several circular or spherical distributions with different dispersions. Biometrika, 70, 333-341.

Green, P. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. Journal of the Royal Statistical Society, B, 46, 149-192.

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 8.3, World Scientific Press, Singapore.

Sarma, Y. and Jammalamadaka, S. (1993). Circular Regression. Statistical Science and Data Analysis, 109-128. Proceeding of the Thrid Pacific Area Statistical Conference. VSP: Utrecht, Netherlands.

Examples

```
# Generate a data set of dependent circular variables.
x <- circular(runif(50, 0, 2*pi))
y <- atan2(0.15*cos(x) + 0.25*sin(x), 0.35*sin(x)) +
  rvonmises(n=50, mu=circular(0), kappa=5)

# Fit a circular-circular regression model.
circ.lm <- lm.circular(y, x, order=1)
# Obtain a crude plot of the data and fitted regression line.
plot.default(x, y)
circ.lm$fitted[circ.lm$fitted>pi] <- circ.lm$fitted[circ.lm$fitted>pi] - 2*pi
points.default(x[order(x)], circ.lm$fitted[order(x)], type='l')

# Fit a circular-linear regression model and show predictions.
set.seed(1234)
x <- cbind(rnorm(10), rep(1, 10))
x <- cbind(rnorm(10), rep(1,10))
y <- circular(2*atan(c(x%*%c(5,1))))+rvonmises(10, mu=circular(0), kappa=100)
lm.circular(y=y, x=x, init=c(5,1), type='c-l', verbose=TRUE)
plot(y)
lmC <- lm.circular(y=y, x=x, init=c(5,1), type='c-l', verbose=TRUE)
p <- circular(lmC$mu+2*atan(x%*%lmC$coefficients))
points(p, col=2, pch= "+")
```

| lsfit.circle | *Fit a 2D circle to an (x,y) dataset* |

Description

Fit a 2D circle to an (x,y) dataset using LS.

Usage

```
lsfit.circle(x, y, init = NULL, units = c("radians", "degrees"),
  template = c("none", "geographics"),
  modulo = c("asis", "2pi", "pi"), zero = 0,
  rotation = c("counter", "clock"), ...)
## S3 method for class 'lsfit.circle'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	either a matrix with two columns or a vector.
y	if x is a vector then y must be a vector with the same length.
init	initial values of the parameters. A vector of length 3 with the following components: radius of the circle, x-coordinate of the center, y-coordinate of the center. If NULL the vector is set to c(max(c(abs(x-mean(x)), abs(y-mean(y)))), mean(x), mean(y).
units	the units used in defining the angles between observations and the center of the circle. See [circular].
template	the template used in defining the angles between observations and the center of the circle. See [circular].
modulo	the modulo used in defining the angles between observations and the center of the circle. See [circular].
zero	the zero used in defining the angles between observations and the center of the circle. See [circular].
rotation	the rotation used in defining the angles between observations and the center of the circle. See [circular].
...	further parameters passed to the optim function.
digits	the number of digits to be printed.

Details

lsfit.circle uses the optim function to minimize the sum of the squared residuals between the observations and the optimally fitting circle.

Value

An object of class lsfit.circle.

coefficients	a vector of length 3 with the estimated radius and coordinate of the center of the fitted circle.
x	the x-coordinate.
y	the y-coordinate.
x.centered	the x-coordinate re-centered at the center of the circle.
y.centered	the y-coordinate re-centered at the center of the circle.

angles	angles of the observations with respect to the center coordinate of class `circular`.
radius	the distance between the observations and the center coordinate
convergence	value from the function optim.
optim	the output from the function optim.
call	the [match.call]{} result.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Coope, I. (1993). Circle fitting by linear and non-linear least squares. Journal of Optimization Theory and Applications, 76, 381-388.

Examples

```
data(coope)
res <- lsfit.circle(x=x.coope, y=y.coope)
res

plot(res)

par(mfcol=c(1,2))
plot(res$angles)
hist(res$radius)

plot(circular(0), type="n", xlim=c(-5.2, 5.2), ylim=c(-5.2, 5.2),
  xlab="The Radius of the circle \n is measured from the base line of the axes.")
lines(x=res$angles, y=res$radius, join=TRUE, type="b")
ff <- function(x) sqrt((res$coefficients[1]*cos(x))^2+(res$coefficients[1]*sin(x))^2)
curve.circular(ff, add=TRUE, join=TRUE, nosort=FALSE, col=2)

windrose(x=res$angles, y=res$radius)
```

---

mean.circular	*Mean Direction*

Description

Returns the mean direction of a vector of circular data.

Usage

```
## S3 method for class 'circular'
mean(x, na.rm=FALSE, control.circular=list(), ...)
```

Arguments

x	a vector. The object is coerced to class [circular]().
na.rm	logical, indicating if [NA]()'s should be omitted.
control.circular	
	the attribute of the resulting object.
...	further arguments passed to or from other methods.

Details

Each observation is treated as a unit vector, or point on the unit circle. The resultant vector of the observations is found, and the direction of the resultant vector is returned. An [NA]() is returned if the resultant length (see [rho.circular]()) is less than [.Machine]()

Value

Returns the mean direction of the data as an object of class circular with the attribute given by control.circular or from x if missed in control.circular.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 1.3, World Scientific Press, Singapore.

See Also

[var.circular](), [summary.circular](), [mle.vonmises](), [rho.circular]() and [.Machine]().

Examples

```
# Compute the mean direction of a random sample of observations.
x <- circular(runif(50, circular(0), pi))
mean(x)
```

---

meandeviation	*A measure of deviation for Circular Data*

Description

Returns a measure of spread associated with the circular median of a vector of circular data.

Usage

```
meandeviation(x, na.rm = FALSE)
```

Arguments

x	a vector. The object is coerced to class [circular]().
na.rm	logical, indicating if [NA]()'s should be omitted.

Details

See equation (2.33) at pag. 36 in Fisher (1993) for its definition. In the case the circular median is not defined, that is, every angle is a median axis, the mean deviation is not reported and set to NA.

Value

Returns a measure of spread associated with the circular median of a vector of circular data.

Author(s)

Claudio Agostinelli and Alessandro Gagliardi

References

N.I. Fisher (1993) Statistical Analysis of Circular Data, Cambridge University Press.

See Also

[median.circular](), [sd.circular](), [angular.variance](), [angular.deviation](), [mean.circular](), [rho.circular]() and [summary.circular]().

Examples

```
x <- rvonmises(n=100, mu=circular(0), kappa=1)
meandeviation(x)
```

---

median.circular   *Median Direction*

Description

Sample median direction for a vector of circular data

Usage

```
## S3 method for class 'circular'
median(x, na.rm = FALSE, ...)
```

Arguments

x	a vector. The object is coerced to class [circular]().
na.rm	logical, indicating if [NA]()'s should be omitted.
...	NotYetUsed.

Details

The Definition in equations 2.32 & 2.33 from N.I. Fisher's 'Statistical Analysis of Circular Data', Cambridge Univ. Press 1993. is implemented. Since version 0.4-4, the algorithm (not the definition) for the calculation of the median is changed. For a measure of spread associated to the circular median use function `meandeviation`.

Value

A scalar with the circular median value.

The median is returned as an object of class `circular`.

Author(s)

Claudio Agostinelli and Alessandro Gagliardi

References

N.I. Fisher (1993) Statistical Analysis of Circular Data, Cambridge University Press.

R.Y. Liu and K. Singh (1992) Ordering Directional Data: Concepts of Data Depth on Circles and Spheres, The Annals of Statistics, vol. 20, n. 3, 1468-1484.

See Also

`meandeviation`, `mean.circular`, `var.circular`, `summary.circular`, `rho.circular` and `medianHL.circular`.

Examples

```
# Compute the median direction of a random sample of observations.
x <- circular(runif(50, circular(0), pi))
median(x) #only the median is returned
meandeviation(x) #mean deviation is reported
```

---

medianHL.circular   *Median using Hodges-Lehmann estimate.*

Description

Sample median for a vector of data using Hodges-Lehmann estimate and Sample median direction measure for a vector of circular data using Hodges-Lehmann estimate.

Usage

```
medianHL(x, na.rm=FALSE, ...)
## Default S3 method:
medianHL(x, na.rm=FALSE,
       method=c("HL1","HL2","HL3"), prop=NULL,...)
## S3 method for class 'circular'
medianHL(x, na.rm=FALSE,
       method=c("HL1","HL2","HL3"), prop=NULL,...)
```

Arguments

x	a vector. For the function medianHL.circular the object is coerced to class circular.
na.rm	logical, indicating if NA's should be omitted.
method	The method used to calculate the median, see details below.
prop	The proportion of pairs that are sampled. If NULL all combinations are used. It must be a number in the interval (0,1) or NULL.
...	further arguments passed to the next method.

Details

The algorithm is as follows:

The algorithm will create pairs of elements of the vector x.

It will calculate the circular mean on those pairs.

It will calculate the circular median on these averages.

The type of pairs considered are controlled by method:

if method is "HL1" are considered unordered pairs without replications and repetition in the number of (n*(n-1))/2 pairs;

if method is "HL2" are considered unordered pairs without replications in the number of (n*(n+1))/2 pairs;

if method is "HL3" all pairs are considered in the number of n^2.

If prop is not NULL, the algorithm will consider a subsample following the rules specified by method, however, the number of pairs considered is prop * (number of pairs defined by method).

For more details see Bennett Sango Otieno, 'An Alternative Estimate of Preferred Direction for Circular Data', Virginia Tech (2002) pag. 27-28 and 46-47.

Value

For medianHL.circular the median is returned as an object of class circular with the attribute given by those of x. An attributes medians reports all the averages which are minimizer of the circular median function.

Author(s)

Claudio Agostinelli and Alessandro Gagliardi.

References

Bennett Sango Otieno, An Alternative Estimate of Preferred Direction for Circular Data, Virginia Tech (July 2002).

Bennett Sango Otieno and Christine M. Anderson-Cook,Measures of preferred direction for environmental and ecological circular data, Springer (June 2004).

See Also

[mean.circular](), [median.circular]().

Examples

```
# Compute the median direction of a random sample of observations.
  x <- circular(runif(50, circular(0), pi))
# Calculate the three medians for each method without \code{prop} argument.
  medianHL.circular(x,method="HL1")
  medianHL.circular(x,method="HL2")
  medianHL.circular(x,method="HL3")
```

---

minusPiPlusPi   *return angles in the [-pi,pi] interval.*

Description

return angles in the (-pi,pi] interval.

Usage

```
minusPiPlusPi(x)
```

Arguments

x   an object of class [circular]().

Value

a [circular]() object with values in the interval (-pi,pi].

Author(s)

Claudio Agostinelli and Alessandro Gagliardi

Examples

```
x <- circular(c(0, 90, 180, 270), units="degrees")
minusPiPlusPi(x)
```

mixedvonmises   *Mixture of von Mises Distributions*

Description

Density and random generation for the mixed von Mises circular distribution.

Usage

```
dmixedvonmises(x, mu1, mu2, kappa1, kappa2, prop)
rmixedvonmises(n, mu1, mu2, kappa1, kappa2, prop, control.circular = list())
pmixedvonmises(q, mu1, mu2, kappa1, kappa2, prop, from=NULL, tol = 1e-020)
```

Arguments

x, q	a vector. The object is coerced to class [circular]().
n	number of observations.
mu1	mean direction of one of the two von Mises distributions as a circular object.
mu2	mean direction of the other von Mises distribution as a circular object.
kappa1	concentration parameter of one of the two von Mises distributions.
kappa2	concentration parameter of the other von Mises distribution.
prop	mixing proportion.
from	if NULL is set equal to 0 (Notice the difference from the corresponding vonmises distribution). This is the value from which the pmixedvonmises is evaluated. It should be a circular object.
tol	the precision in evaluating the distribution function or the quantile.
control.circular	
	the attribute of the resulting object.

Value

dmixedvonmises gives the density, pmixedvonmises gives the distribution function and rmixedvonmises generates random deviates.

Author(s)

Claudio Agostinelli and Ulric Lund

See Also

[dvonmises]() [pvonmises]() and [rvonmises]()

Examples

```
x <- rmixedvonmises(n=100, mu1=circular(0), mu2=circular(pi), kappa1=15,
  kappa2=15, prop=0.5)
plot(x)
```

---

mle.vonmises   *von Mises Maximum Likelihood Estimates*

Description

Computes the maximum likelihood estimates for the parameters of a von Mises distribution: the mean direction and the concentration parameter.

Usage

```
mle.vonmises(x, mu=NULL, kappa=NULL, bias=FALSE, control.circular=list())
## S3 method for class 'mle.vonmises'
print(x,
    digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	a vector. The object is coerced to class circular.
mu	if NULL the maximum likelihood estimate of the mean direction is calculated. If provided it is coerced to a class circular.
kappa	if NULL the maximum likelihood estimate of the concentration parameter is calculated.
bias	logical, if TRUE, the estimate for kappa is computed with a bias corrected method. Default is FALSE, i.e. no bias correction.
control.circular	
	the attribute of the resulting objects (mu)
digits	integer indicating the precision to be used.
...	further arguments passed to or from other methods.

Details

Best and Fisher (1981) show that the MLE of kappa is seriously biased when both sample size and mean resultant length are small. They suggest a bias-corrected estimate for kappa when n < 16.

Value

Returns a list with the following components:

`call`	the `match.call` result.
`mu`	the estimate of the mean direction or the value supplied as an object of class `circular`.
`kappa`	the estimate of the concentration parameter or the value supplied
`se.mu`	the standard error for the estimate of the mean direction (0 if the value is supplied) in the same units of `mu`.
`se.kappa`	the standard error for the estimate of the concentration parameter (0 if the value is supplied).
`est.mu`	TRUE if the estimator is reported.
`est.kappa`	TRUE if the estimator is reported.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 4.2.1, World Scientific Press, Singapore.

Best, D. and Fisher N. (1981). The bias of the maximum likelihood estimators of the von Mises-Fisher concentration parameters. Communications in Statistics - Simulation and Computation, B10(5), 493-502.

See Also

`mean.circular` and `mle.vonmises.bootstrap.ci`

Examples

```
x <- rvonmises(n=50, mu=circular(0), kappa=5)
mle.vonmises(x) # estimation of mu and kappa
mle.vonmises(x, mu=circular(0)) # estimation of kappa only
```

---

mle.vonmises.bootstrap.ci

Bootstrap Confidence Intervals

Description

Generates simple bootstrap confidence intervals for the parameters of a von Mises distribution: the mean direction mu, and the concentration parameter kappa.

Usage

```
mle.vonmises.bootstrap.ci(x, mu = NULL, bias = FALSE, alpha = 0.05,
          reps = 1000, control.circular = list())
## S3 method for class 'mle.vonmises.bootstrap.ci'
print(x, ...)
```

Arguments

x	vector of angular measurements as a `circular` object.
mu	If `NULL` the value is estimated. This value is used in the bootstrap replications for kappa.
bias	logical, if `TRUE`, the replication estimates for kappa are computed with a bias corrected method. See `mle.vonmises`. Default is FALSE, i.e. no bias correction.
alpha	parameter determining level of confidence intervals. 1-alpha confidence intervals for mu and kappa are computed. By default, 95% confidence intervals are generated.
reps	number of resampled data sets to use. Default is 1000.
control.circular	
	the attribute of the resulting objects (mu, mu.ci).
...	arguments passed to `print.default`.

Details

Percentile confidence intervals are computed by resampling from the original data set `reps` times. For each resampled data set, the MLE's of mu and kappa are computed. The bootstrap confidence intervals are the alpha/2 and 1-alpha/2 percentiles of the `reps` MLE's computed for each resampled data set.

Value

A list is returned with the following components:

mu.ci	limits of the confidence interval for mu as a `circular` object.
kappa.ci	limits of the confidence interval for kappa.
mu	estimate of mu as a `circular` object.
kappa	estimate of kappa.

Author(s)

Claudio Agostinelli and Ulric Lund

See Also

`mle.vonmises`

Examples

```
x <- rvonmises(n=25, mu=circular(0), kappa=3)
x.bs <- mle.vonmises.bootstrap.ci(x, alpha=.10)
par(mfcol=c(1,2))
rose.diag(x.bs$mu, bins=30, main=expression(mu))
hist(x.bs$kappa, main=expression(kappa))
```

---

mle.wrappedcauchy *Wrapped Cauchy Maximum Likelihood Estimates*

Description

Computes the maximum likelihood estimates for the parameters of a Wrapped Cauchy distribution: mean and concentration parameter.

Usage

```
mle.wrappedcauchy(x, mu = NULL, rho = NULL, tol = 1e-15,
        max.iter = 100, control.circular = list())
## S3 method for class 'mle.wrappedcauchy'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	a vector. The object is coerced to class [circular](#).
mu	if NULL the maximum likelihood estimate of the mean direction is calculated otherwise it is coerced to an object of class circular.
rho	if NULL the maximum likelihood estimate of the concentration parameter is calculated.
tol	precision of the estimation.
max.iter	maximum number of iterations.
control.circular	
	the attribute of the resulting objects (mu)
digits	integer indicating the precision to be used.
...	further arguments passed to or from other methods.

Value

Returns a list with the following components:

call	the [match.call](#) result.
mu	the estimate of the mean direction or the value supplied as an object of class circular.
rho	the estimate of the concentration parameter or the value supplied
convergence	TRUE if convergence is achieved.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 4.2.1, World Scientific Press, Singapore.

See Also

[mean.circular](#)

Examples

```
x <- rwrappedcauchy(n=50, mu=circular(0), rho=0.5)
mle.wrappedcauchy(x) # estimation of mu and rho
mle.wrappedcauchy(x, mu=circular(0)) # estimation of rho only
```

---

mle.wrappednormal   *Wrapped Normal Maximum Likelihood Estimates*

Description

Computes the maximum likelihood estimates for the parameters of a Wrapped Normal distribution: mean and concentration parameter.

Usage

```
mle.wrappednormal(x, mu = NULL, rho = NULL, sd = NULL, K = NULL,
  tol = 1e-05, min.sd = 1e-3, min.k = 10, max.iter = 100,
  verbose = FALSE, control.circular=list())
## S3 method for class 'mle.wrappednormal'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	a vector. The object is coerced to class [circular](#).
mu	if NULL the maximum likelihood estimate of the mean direction is calculated, otherwise the value is coerced to an object of class circular.
rho	if NULL the maximum likelihood estimate of the concentration parameter is calculated.
sd	standard deviation of the (unwrapped) normal. Used as an alternative parametrization.
K	number of terms to be used in approximating the density.
tol	precision of the estimation.

`min.sd`	minimum value should be reached by the search procedure for the standard deviation parameter.
`min.k`	minimum number of terms used in approximating the density.
`max.iter`	maximum number of iterations.
`verbose`	logical, if TRUE information on the convergence process are printed.
`control.circular`	
	the attribute of the resulting objects (mu)
`digits`	integer indicating the precision to be used.
`...`	further arguments passed to or from other methods.

Value

Returns a list with the following components:

`call`	the [match.call](#) result.
`mu`	the estimate of the mean direction or the value supplied as an object of class `circular`.
`rho`	the estimate of the concentration parameter or the value supplied
`sd`	the estimate of the standard deviation or the value supplied.
`est.mu`	TRUE if the estimator is reported.
`est.rho`	TRUE if the estimator is reported.
`convergence`	TRUE if the convergence is achieved.

Author(s)

Claudio Agostinelli with a bug fix by Ana Nodehi

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 4.2.1, World Scientific Press, Singapore.

See Also

[mean.circular](#)

Examples

```
x <- rwrappednormal(n=50, mu=circular(0), rho=0.5)
mle.wrappednormal(x) # estimation of mu and rho (and sd)
mle.wrappednormal(x, mu=circular(0)) # estimation of rho (and sd) only
```

---

modal.region	*Modal regions*

Description

Evaluate the modal regions for a data set. Only the version for circular data is implemented.

Usage

```
modal.region(x, ...)
## Default S3 method:
modal.region(x, ...)
## S3 method for class 'circular'
modal.region(x, z=NULL, q=0.95, bw, adjust = 1,
  type = c("K", "L"), kernel = c("vonmises", "wrappednormal"),
  na.rm = FALSE, step=0.01, eps.lower=10^(-4), eps.upper=10^(-4), ...)
```

Arguments

x	numeric or an object of class [circular]().
z	numeric or object of class [circular](). The grid were the kernel density estimate will be evaluated. If NULL equally spaced points in the interval [0,2*pi) with step step.
q	numeric in the interval [0,1]. The quantile of the modal region.
bw	the smoothing bandwidth to be used. When the kernel is vonmises the bandwidth is equal to the concentration parameter.
adjust	the bandwidth used is actually adjust*bw. This makes it easy to specify values like "half the default bandwidth".
type	Not Yet Used.
kernel	a character string giving the smoothing kernel to be used. This must be one of "vonmises" or "wrappednormal", that are kernels of type "K".
na.rm	logical; if TRUE, missing values are removed from x. If FALSE any missing values cause an error.
step	numeric. Used in the construction of the regular grid z.
eps.lower, eps.upper	
	the cut point in the density is searched in the interval [min(density)*(1+eps.lower),max(density)*(1-eps.upper)].
...	further arguments passed to the next methods.

Details

Only the version for circular data is actually implemented.

Value

A list of class `modal.region.circular` with the following elements

zeros	extremes of modal regions, possible as a matrix
areas	a list with two components: `tot` with the total (area under the density) probability, which should approximately equal to q and `areas` with the probability of each modal region.
density	the object from function `density.circular`.
q	the modal region order as in input.
level	the cut point at the density scale.

Author(s)

Claudio Agostinelli

References

L.G.R. Oliveira-Santos, C.A. Zucco and C. Agostinelli (2013) Using conditional circular kernel density functions to test hypotheses on animal circadian activity. Animal Behaviour, 85(1) 269-280.

See Also

[totalvariation.circular](#)

Examples

```
x <- rvonmises(100, circular(pi), 10)
res <- modal.region(x, bw=50)
plot(res)
```

---

ncfrog   *Northern Cricket Frog*

Description

In an experiment due to Ferguson et al. (1967) a number of northern cricket frogs (Acris crepitans) were collected from the mud flats of an abandoned stream meandering near Indianola, Mississippi, and taken to a test pen lying to the north west of the collection point. After 30 hours of enclosure within a dark environmental chamber, 14 of them were released and the directions taken by these frogs recorded. 0 degrees means North.

Usage

```
data(ncfrog)
```

Format

ncfrog is a vector of 14 observations (in degrees). ncfrog.rad contains the same observations in radians (pi/180).

Source

Collett, D. (1980) Outliers in Circular Data *Applied Statistics* **29**, 1, 50–57.

See Also

Ferguson, D.E, Landreth, H.F. and McKeown, J.P. (1967) Sun compass orientation of the northern cricket frog, *Acris crepitans*. *Anim. Behav.*, **14**, 45–53.

---

pigeons   *Initial orientation of displaced homing pigeons*

Description

This data set has 108 rows and 2 columns. The observations are the vanishing bearings of homing pigeons displaced and released at two unfamiliar locations. The data are pooled with respect to the home direction (home direction set in 360 grades).

Usage

data(pigeons)

Format

This data frame contains the following columns:

treatment , a factor with levels: *c*, control pigeon (unmanipulated); *v1*, pigeons subjected to bilateral section of the ophthalmic branch of the trigeminal nerve; *on*, pigeons subjected to bilateral section of the olfactory nerve

bearing , vanishing bearing of each bird in degrees

References

Gagliardo A., Ioale' P., Savini M., and Wild M. (2008). Navigational abilities of homing pigeons deprived of olfactory or trigeminally mediated magnetic information when young. *J. Exp. Biol.*, **211**:2046–2051.

---

plot.circular   *Circular Data Plot*

Description

Creates a plot of circular data points on the current graphics device. Data points are either plotted as points on the unit circle, or the range of the circle is divided into a specified number of bins, and points are stacked in the bins corresponding to the number of observations in each bin.

Usage

```
## S3 method for class 'circular'
plot(x, pch = 16, cex = 1, stack = FALSE,
  axes = TRUE, start.sep=0, sep = 0.025, shrink = 1,
  bins = NULL, ticks = FALSE, tcl = 0.025, tcl.text = 0.125,
  col = NULL, tol = 0.04, uin = NULL,
  xlim = c(-1, 1), ylim = c(-1, 1), digits = 2, units = NULL,
  template = NULL, zero = NULL, rotation = NULL,
  main = NULL, sub=NULL, xlab = "", ylab = "",
  control.circle=circle.control(), ...)
```

Arguments

x	a vector, matrix or data.frame. The object is coerced to class [circular](#).
pch	point character to use. See help on [par](#).
cex	point character size. See help on [par](#).
stack	logical; if TRUE, points are stacked on the perimeter of the circle. Otherwise, all points are plotted on the perimeter of the circle. Default is FALSE.
axes	logical; if TRUE axes are plotted according to properties of x.
start.sep	constant used to specify the distance between the center of the point and the axis.
sep	constant used to specify the distance between stacked points, if stack==TRUE or in the case of more than one dataset. Default is 0.025; smaller values will create smaller spaces.
shrink	parameter that controls the size of the plotted circle. Default is 1. Larger values shrink the circle, while smaller values enlarge the circle.
bins	if stack==TRUE, bins is the number of arcs to partition the circle with.
ticks	logical; if TRUE ticks are plotted according to the value of bins.
tcl	length of the ticks.
tcl.text	The position of the axis labels.
col	color of the points. The values are recycled if needed.
tol	proportion of white space at the margins of plot.

uin	desired values for the units per inch parameter. If of length 1, the desired units per inch on the x axis.
xlim, ylim	the ranges to be encompassed by the x and y axes. Useful for centering the plot.
digits	number of digits used to print axis values.
main, sub, xlab, ylab	
	title, subtitle, x label and y label of the plot.
units	the units used in the plot.
template	the template used in the plot.
zero	the zero used in the plot.
rotation	the rotation used in the plot.
control.circle	parameters passed to `plot.default` in order to draw the circle. The function `circle.control` is used to set the parameters.
...	further parameters passed to `points.default`.

Details

When there are many closely distributed observations, stacking is recommended. When stacking the points, if there are many points in a particular bin, it may be necessary to shrink the plot of the circle so that all points fit. This is controlled with the parameter shrink. Generally the parameter sep does not need adjustment, however, when shrinking the plot, or for a very large number of observations, it may be helpful. Since version 0.3-9 the intervals are on the form [a,b).

Value

A list with information on the plot: zero, rotation and next.points.

Note

some codes from eqscplot in MASS is used.

Author(s)

Claudio Agostinelli and Ulric Lund

See Also

axis.circular, ticks.circular, points.circular, lines.circular, rose.diag, windrose and curve.circular.

Examples

```
# Generate 100 observations from a von Mises distribution.
# with mean direction 0 and concentration 3.
data.vm <- rvonmises(n=100, mu=circular(0), kappa=3)

# Plot data set. All points do not fit on plot.
plot(data.vm, stack=TRUE, bins=150)
```

```
# Shrink the plot so that all points fit.
plot(data.vm, stack=TRUE, bins=150, shrink=1.5)

# Recentering the figure in a different place
plot(data.vm, stack=TRUE, bins=150, xlim=c(-1,1.2), ylim=c(-1,0))
```

---

plot.density.circular   *Plot Method for Kernel Density Estimation for Circular Data*

Description

The `plot` method for `density.circular` objects.

Usage

```
## S3 method for class 'density.circular'
plot(x, main=NULL, sub=NULL, xlab=NULL, ylab="Density circular", type="l",
  zero.line=TRUE, points.plot=FALSE, points.col=1, points.pch=1,
  points.cex=1, plot.type=c("circle", "line"), axes=TRUE, ticks=FALSE,
 bins=NULL, offset=1, shrink=1, tcl=0.025, tcl.text = 0.125, sep=0.025, tol=0.04,
  digits=2, cex=1, uin=NULL, xlim=NULL, ylim=NULL, join=FALSE, nosort=FALSE,
  units=NULL, template=NULL, zero=NULL, rotation=NULL,
  control.circle=circle.control(), ...)
```

Arguments

x	an object of class [density.circular]().
main, sub, xlab, ylab, type	
	plotting parameters with useful defaults.
zero.line	logical; if TRUE, add a base line at $y = 0$. Used when `plot.type` is "line".
points.plot	logical; if TRUE original data are added to the plot.
points.col, points.pch, points.cex	
	parameters used to draw the points.
plot.type	type of the plot: "line": linear plot, "circle": circular plot.
axes	logical; if TRUE axis are drawn.
ticks	logical; if TRUE ticks are drawn.
bins	number of ticks to plot.
offset	the radius of the circle
shrink	parameter that controls the size of the plotted function. Default is 1.
tcl	length of the ticks.
tcl.text	The position of the axis labels.
sep	constant used to specify the distance between stacked points. Default is 0.025; smaller values will create smaller spaces.

tol	proportion of white space at the margins of plot
digits	number of digits used to print axis values.
cex	point character size. See help on `par`.
uin	desired values for the units per inch parameter. If of length 1, the desired units per inch on the x axis.
xlim, ylim	the ranges to be encompassed by the x and y axes. Useful for centering the plot.
join	logical: should the first and the last point joined.
nosort	logical: should the data sort before plotting. Defaults is to sort.
units	units measure used in the plot. If NULL the value is taken from the attribute of object 'x' from the argument 'x', i.e. x$x.
template	template used in the plot. If NULL the value is taken from the attribute of object 'x' from the argument 'x', i.e. x$x.
zero	position of the zero used in the plot. If NULL the value is taken from the attribute of object 'x' from the argument 'x', i.e. x$x.
rotation	rotation used in the plot. If NULL the value is taken from the attribute of object 'x' from the argument 'x', i.e. x$x.
control.circle	parameters passed to `plot.default` in order to draw the circle. The function `circle.control` is used to set the parameters.
...	further parameters passed to `plot.default`.

Value

A list with information on the plot: zero, rotation and next.points.

Author(s)

Claudio Agostinelli

See Also

`density.circular`, `lines.density.circular`, `plot.circular`, `lines.circular` and `curve.circular`.

Examples

```
set.seed(1234)
x <- rvonmises(n=100, mu=circular(pi), kappa=2)
res25x <- density(x, bw=25)
plot(res25x, points.plot=TRUE, xlim=c(-1.5,1))
res50x <- density(x, bw=25, adjust=2)
lines(res50x, col=2)

resp25x <- plot(res25x, points.plot=TRUE, xlim=c(-1, 1.3), ylim=c(-1.5,1.2),
  template="geographics", main="Plotting density estimate for two data set")
y <- rvonmises(n=100, mu=circular(pi/2), kappa=2,
  control.circular=list(template="geographics"))
res25y <- density(y, bw=25)
lines(res25y, points.plot=TRUE, plot.info=resp25x, col=2, points.col=2)
```

```
plot(res25x, plot.type="line", points.plot=TRUE, xlim=c(-1, 1.3), ylim=c(-1.5,1.2),
  template="geographics", main="Plotting density estimate for two data set")
lines(res25y, plot.type="line", points.plot=TRUE, col=2, points.col=2)
```

---

plot.edf *Plot Circular Empirical Distribution Function*

Description

Plots the empirical distribution function of a circular data set.

Usage

```
## S3 method for class 'edf'
plot(x, type = "s", xlim = c(0, 2 * pi), ylim = c(0, 1), ...)
## S3 method for class 'edf'
lines(x, type = "s", ...)
```

Arguments

x	vector of circular data measured.
type, xlim, ylim	plotting parameters with useful defaults. xlim is in radians.
...	optional graphical parameters. See help section on [par](#).

Details

The vector of data is taken modulo 2*pi, and then the linear ranks are used to generate an empirical distribution function.

Note

Creates a plot or adds a plot (lines.edf) of the empirical distribution function of the circular data vector.

Author(s)

Claudio Agostinelli and Ulric Lund

See Also

[plot.ecdf](#), [curve.circular](#) and [par](#).

Examples

```
# Compare the edf's of two simulated sets of data.
data1 <- rvonmises(n=10, mu=circular(0), kappa=3)
data2 <- rvonmises(n=10, mu=circular(0), kappa=1)
plot.edf(data1, xlab="Data", ylab="EDF", main="Plots of Two EDF's")
lines.edf(data2, lty=2, col=2)

#You can use standard ecdf and plot.ecdf functions
ff <- function(x, data) {
     x <- x
     data <- data
     temp <- ecdf(data)
     temp(x)
}
plot(function(x) ff(x, data=data1), from=0, to=2*pi-3*.Machine$double.eps)

#Or curve.circular
plot.function.circular(function(x) ff(x, data=data1), from=0,
  to=(2*pi-3*.Machine$double.eps), join=FALSE, nosort=TRUE, xlim=c(-2,2),
  ylim=c(-2,2), modulo="asis", main="Empirical Distribution Function",
  n=2001, tcl.text=0.25)

res <- plot.function.circular(function(x) ff(x, data=data2), from=0,
  to=(2*pi-3*.Machine$double.eps), join=FALSE, nosort=TRUE, modulo="asis",
  add=TRUE, col=2, n=2001)

res1 <- points(data1, plot.info=res)
points(data2, plot.info=res1, col=2, sep=0.05)

legend(-1.9, 1.9, legend=c("data1", "data2"), col=c(1,2), lty=c(1,1))
```

---

plot.lsfit.circle   *Plot method for lsfit.circle function*

Description

This is a plot method for objects of class lsfit.circle.

Usage

```
## S3 method for class 'lsfit.circle'
plot(x, add = FALSE, main = NULL, xlim = NULL, ylim = NULL,
  xlab = NULL, ylab = NULL, uin, tol = 0.04, plus.cex = 1, ...)
```

Arguments

x	an object of class lsfit.circle.
add	logical: if TRUE the plot is superimposed on the active device.

main	a main title for the plot.
xlim	the x limits (min,max) of the plot.
ylim	the y limits of the plot.
xlab	a label for the x axis.
ylab	a label for the x axis.
uin	desired values for the units per inch parameter. If of length 1, the desired units per inch on the x axis.
tol	proportion of white space at the margins of plot.
plus.cex	dimension of the cross in the center of the circle.
...	further arguments passed to the next method.

Author(s)

Claudio Agostinelli and Ulric Lund

See Also

lsfit.circle

Examples

```
data(coope)
res <- lsfit.circle(x=x.coope, y=y.coope)
plot(res)
```

---

points.circular	*Add Points to a Circular Plot*

Description

Add points to a plot of circular data points on the current graphics device.

Usage

```
## S3 method for class 'circular'
points(x, pch = 16, cex = 1, stack = FALSE,
  start.sep=0, sep = 0.025,
  shrink = 1, bins = NULL, col = NULL, next.points = NULL,
  plot.info = NULL, zero = NULL, rotation = NULL, ...)
```

Arguments

x	a vector, matrix or data.frame. The object is coerced to class [circular](#).
pch	point character to use. See help on [par](#).
cex	point character size. See help on par.
stack	logical: if TRUE, points are stacked on the perimeter of the circle. Otherwise, all points are plotted on the perimeter of the circle. Default is FALSE.
start.sep	constant used to specify the distance between the center of the point and the axis.
sep	constant used to specify the distance between stacked points, if stack==TRUE or in the case of more than one dataset. Default is 0.025; smaller values will create smaller spaces.
shrink	parameter that controls the size of the plotted circle. Default is 1. Larger values shrink the circle, while smaller values enlarge the circle.
bins	if stack==TRUE, bins is the number of arcs to partition the circle with.
col	color of the points. The values are recycled if needed.
next.points	if stack=FALSE, the distance from the circle the next dataset is plotted. Ignored if plot.info is provided.
plot.info	an object from [plot.circular](#) that contains information on the zero, the rotation and next.points.
zero	the zero of the plot. Ignored if plot.info is provided.
rotation	the rotation of the plot. Ignored if plot.info is provided.
...	further parameters passed to [points.default](#).

Details

When there are many closely distributed observations, stacking is recommended. When stacking the points, if there are many points in a particular bin, it may be necessary to shrink the plot of the circle so that all points fit. This is controlled with the parameter shrink. Generally the parameter sep does not need adjustment, however, when shrinking the plot, or for a very large number of observations, it may be helpful. Since version 0.3-9 the intervals are on the form [a,b).

Value

A list with information on the plot: zero, rotation and next.points.

Author(s)

Claudio Agostinelli

See Also

[plot.circular](#) and [lines.circular](#).

Examples

```
data.1 <- rvonmises(n=100, mu=circular(0), kappa=3)
data.2 <- rvonmises(n=100, mu=circular(pi/3), kappa=3)
res <- plot(data.1, stack=FALSE, col=1)
points(data.2, plot.info=res, col=2)
```

---

pp.plot   *von Mises Probability-Probability Plot*

Description

Plots the empirical distribution of a data set against the best fitting von Mises distribution function.

Usage

```
pp.plot(x, ref.line = TRUE, tol=1e-20,  xlab = "von Mises Distribution",
  ylab = "Empirical Distribution", control.circular = list(), ...)
```

Arguments

x	a vector. The object is coerced to class `circular`.
ref.line	logical, if TRUE a 45 degree reference line is added to the plot. Default is TRUE.
tol	parameter passed to `pvonmises`.
xlab, ylab	labels of the axis.
control.circular	
	the attribute of the resulting object.
...	parameters passed to the `plot.default` function.

Details

The maximum likelihood estimates of the parameters of the von Mises distribution are computed from the given data set. The empirical distribution function is plotted against a von Mises distribution function with parameters given by the MLEs computed.

Value

a list with the estimated mean and concentration parameter for a von Mises distribution.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 10.2, World Scientific Press, Singapore.

See Also

mle.vonmises

Examples

```
x <- rvonmises(n=25, mu=circular(0), kappa=3)
pp.plot(x)
x <- c(rvonmises(n=20, mu=circular(0), kappa=7),
  rvonmises(n=20, mu=circular(pi), kappa=7))
pp.plot(x)
```

---

pp.unif.plot   *Uniform Circular Probability-Probability Plot*

Description

Plots the empirical distribution of a data set against a uniform circular distribution function.

Usage

```
pp.unif.plot(x, ref.line = TRUE, frac = NULL,  xlab = "Uniform Distribution",
  ylab = "Empirical Distribution", col = NULL,
  col.inf = NULL, col.sup = NULL, ...)
```

Arguments

x	a vector. The object is coerced to class circular.
ref.line	logical, if TRUE a 45 degree reference line is added to the plot. Default is TRUE.
frac	a number in the [0,1] interval or NULL.
xlab, ylab	labels of the axis.
col	color of the points.
col.inf, col.sup	
	color of the fraction of the points replicated in the left bottom and right upper corner of the plot.
...	parameters passed to the plot.default function.

Author(s)

Claudio Agostinelli

See Also

pp.plot for the von Mises distribution.

Examples

```
x <- rvonmises(n=25, mu=circular(0), kappa=3)
pp.unif.plot(x)
pp.unif.plot(x, frac=0.2)
```

---

projected bivariate normal on the circle

Projected bivariate normal on the circle

Description

The projected normal distribution provides a flexible distribution for circular data, e.g., asymmetry and possible bimodality.

Usage

```
dpnorm(x, mu, sigma, log = FALSE)
rpnorm(n, mu, sigma, control.circular=list())
```

Arguments

x	a vector. The x and q objects are coerced to class [circular]().
n	number of observations.
mu	the mean vector of the bivariate normal.
sigma	the 2x2 variance and covariance matrix of the bivariate normal.
log	logical. If TRUE the log of the density is reported.
control.circular	
	the attribute of the resulting object.

Value

dpnorm gives the density, rpnorm generates random deviates.

Author(s)

Claudio Agostinelli

References

S.R. Jammalamadaka and A. SenGupta (2001). Topics in Circular Statistics, Section 2.2.4, World Scientific Press, Singapore. K.V. Mardia (1972). Statistics of Directional Data. Academic Press. London and New York. F. Wang and A.E. Gelfand (2013). Directional data analysis under the general projected normal distribution. Stat Methodol. 10(1):113-127. doi:10.1016/j.stamet.2012.07.005.

Examples

```
data1 <- rpnorm(100, mu=c(0,0), sigma=diag(2),
  control.circular=list(units="degrees")) # Uniform on the circle
plot(data1)

ff <- function(x) dpnorm(x, mu=c(0,0), sigma=diag(2)) # Uniform on the circle
curve.circular(ff, join=TRUE,
  main="Density of a Projected Normal Distribution \n mu=(0,0), sigma=diag(2)")

ff <- function(x) dpnorm(x, mu=c(1,1), sigma=diag(2)) # Unimodal
curve.circular(ff, join=TRUE, xlim=c(-1, 2.3),
  main="Density of a Projected Normal Distribution \n mu=(1,1), sigma=diag(2)")

sigma <- matrix(c(1,0.9,0.9,1), nrow=2)
ff <- function(x) dpnorm(x, mu=c(0.5,0.5), sigma=sigma) # Bimodal
curve.circular(ff, join=TRUE, xlim=c(-1, 2.3),
  main="Density of a Projected Normal Distribution \n mu=(0.5,0.5), rho=0.9")
```

---

quantile.circular   *Sample Circular Quantiles*

Description

The function `quantile.circular` produces sample circular quantiles corresponding to the given probabilities for a circular data set.

Usage

```
## S3 method for class 'circular'
quantile(x, probs = seq(0, 1, 0.25), na.rm=FALSE, names = TRUE, type = 7, ...)
```

Arguments

x	numeric circular vector whose sample quantiles are wanted. NA and NaN values are not allowed in numeric vectors unless na.rm is TRUE.
probs	numeric vector of probabilities with values in $[0, 1]$. (Values up to '2e-14' outside that range are accepted and moved to the nearby endpoint.)
na.rm	logical; if true, any NA and NaN's are removed from x before the quantiles are computed.
names	logical; if true, the result has a names attribute. Set to FALSE for speedup with many probs.
type	an integer between 1 and 9 selecting one of the nine quantile algorithms detailed below to be used.
...	further arguments passed to or from other methods. Like quantile and so on.

Details

A vector of length `length(probs)` is returned; if `names = TRUE`, it has a `names` attribute.

`NA` and `NaN` values in `probs` are propagated to the result.

The algorithm will proceed how described below: 1) Linearize the circular observations. 2) Calculate the linear median like type establish. 3) The value it will transformed in circular.

Types

See description on documentation of `quantile`.

Author(s)

Claudio Agostinelli and Alessandro Gagliardi.

Examples

```
x <- rvonmises(1001, mu=circular(pi), kappa=5)
quantile.circular(x) # Extremes & Quartiles by default
```

---

rad	*Radians*

Description

Converts degrees to radians.

Usage

```
rad(x)
```

Arguments

x   vector or matrix of degree measurements.

Details

This function is available for compatibility with the CircStats package, please use `conversion.circular`.

Value

Returns a vector or matrix of radian measurements corresponding to the data in degrees.

Author(s)

Claudio Agostinelli and Ulric Lund

See Also

`conversion.circular` and `deg`

---

range.circular   *Circular Range*

Description

Computes the circular range of a data set and performs a test of uniformity if specified.

Usage

```
## S3 method for class 'circular'
range(x, test=FALSE, na.rm = FALSE, finite = FALSE,
  control.circular=list(), ...)
```

Arguments

x	a vector. The object is coerced to class `circular`.
test	logical flag: if TRUE then the test of uniformity is performed; otherwise the test is not performed. Default is FALSE.
na.rm	logical, indicating if `NA`'s should be omitted.
finite	logical, indicating if all non-finite elements should be omitted.
control.circular	
	the attribute of the resulting object.
...	further parameter passed from/to the method.

Details

The circular range is the shortest arc on the circle containing the entire set of data. The p-value is computed using the exact distribution of the circular range under the hypothesis of uniformity, details can be found in Mardia and Jupp (1999) pag. 107.

Value

Returns the circular range as a `circular` object. If the significance test is requested the p-value of the test is returned as p.value.

Author(s)

Claudio Agostinelli and Ulric Lund

References

K.V. Mardia and P.E. Jupp (1999) Directional Statistics, Wiley.

See Also

`kuiper.test`, `rao.spacing.test`, `rayleigh.test` and `watson.test`.

Examples

```
data <- rvonmises(n=50, mu=circular(0), kappa=2)
range(data, test=TRUE)
data <- circular(runif(50, 0, 2*pi))
range(data, test=TRUE)
```

---

rao.spacing.test   *Rao's Spacing Test of Uniformity*

Description

Performs Rao's spacing test of uniformity.

Usage

```
rao.spacing.test(x, alpha=0)
## S3 method for class 'rao.spacing.test'
print(x, digits = 4, ...)
```

Arguments

x	a vector. The object is coerced to class `circular`.
alpha	numeric value specifying the significance level of the test. The default value is 0, in which case, a range for the p-value will be returned. Valid significance levels are 0.10, 0.05, 0.01 and 0.001.
digits	integer indicating the precision to be used.
...	further arguments passed to or from other methods.

Details

If alpha is specified, critical values are determined (using the `print` function) from a table of simulated critical points (see reference below); in this case the `print` function return a further value accepted which is `TRUE` if the null hypothesis is accepted and `FALSE` otherwise. If alpha is not specified, a range for the p-value is determined using the table of simulated critical points in the `print` function but not reported.

Value

a list with the statistic, alpha and the number of observations.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 7.4, World Scientific Press, Singapore.

Rao, J.S. (1976). Some tests based on arc-lengths for the circle. Sankhya, The Indian Journal of Statistics, Serial B(4), 38, 329-338.

Russell, G.S. and Levitin, D.J. (1995). An expanded table of probability values for Rao's Spacing Test. Communications in Statistics - Simulation and Computation, 24, 4, 879-888.

See Also

range.circular, kuiper.test, rayleigh.test and watson.test

Examples

```
x <- circular(runif(200, 0, 2*pi))
rao.spacing.test(x)

res <- print(rao.spacing.test(x, alpha=0.1))
res$accepted

x <- rvonmises(100, circular(0), 20)
rao.spacing.test(x)
```

---

rao.table	*Table for Rao's Spacing Test of Uniformity*

Description

Table for Rao's spacing test of uniformity

Usage

```
data(rao.table)
```

Author(s)

Ulric Lund

See Also

rao.spacing.test

---

rao.test   *Rao's Tests for Homogeneity*

Description

Performs Rao's test for homogeneity on k populations of angular data.

Usage

```
rao.test(..., alpha=0)
## S3 method for class 'rao.test'
print(x, digits = 4, ...)
```

Arguments

...	a sequence of [circular](#) for the rao.test and further arguments passed to or from other methods for the print.rao.test function.
alpha	numeric value specifying the significance level of the test. Default is 0, in which case p-values for the test statistic is printed.
x	an object from the rao.test.
digits	integer indicating the precision to be used.

Details

Critical values and p-values are determined according to the chi-squared approximation of the test statistic.

Value

A list with the statistic and p.value for the mean and the dispersion and the value of alpha.

Note

The test is performed, and the results are written to the screen. Test results are given for both the test of equality of polar vectors, and of dispersions. If alpha is specified, the test statistic is printed, along with the level critical value. If alpha is not specified, a p-value for the test is printed.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 7.6.1, World Scientific Press, Singapore.

Rao, J.S. (1967). Large sample tests for the homogeneity of angular data, Sankhya, Ser, B., 28.

Examples

```
x <- rvonmises(100, circular(0), kappa=10)
y <- rvonmises(100, circular(0), kappa=10)

rao.test(x, y)
```

---

rayleigh.test   *Rayleigh Test of Uniformity*

Description

Performs a Rayleigh test of uniformity, assessing the significance of the mean resultant length. The alternative hypothesis is a unimodal distribution with unknown mean direction and unknown mean resultant length if mu is NULL otherwise the alternative hypothesis is a unimodal distribution with a specified mean direction and unknown mean resultant length.

Usage

```
rayleigh.test(x, mu = NULL)
## S3 method for class 'rayleigh.test'
print(x, digits=4, ...)
```

Arguments

x	a vector. The object is coerced to class circular.
mu	Specified mean direction in alternative hypothesis as a circular object.
digits	integer indicating the precision to be used.
...	further arguments passed to or from other methods.

Value

Returns a list with three components: the mean resultant length, statistic, the p-value of the test statistic, p.value and the value of the alternative mean direction mu.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Sections 3.3.2 and 3.4.1, World Scientific Press, Singapore.

See Also

range.circular, kuiper.test, rao.spacing.test and watson.test

Examples

```
x <- rvonmises(n=25, mu=circular(pi), kappa=2)
# General alternative
rayleigh.test(x)
# Specified alternative
rayleigh.test(x, mu=circular(0))
```

---

rho.circular   *Mean Resultant Length*

Description

Returns the mean resultant length of a vector of circular data.

Usage

```
rho.circular(x, na.rm = FALSE)
```

Arguments

x   a vector. The object is coerced to class circular.
na.rm   logical, indicating if NA's should be omitted.

Details

Each observation is treated as a unit vector, or point on the unit circle. The resultant vector of the observations is found, and the length of the resultant vector divided by the sample size is returned.

Value

Returns the mean resultant length of data.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 1.3, World Scientific Press, Singapore.

See Also

mean.circular, var.circular, summary.circular and mle.vonmises.

Examples

```
# Compute the mean resultant length of a random sample of observations.
data <- circular(runif(100, 0, 2*pi))
rho.circular(data)
```

---

rose.diag   *Rose Diagram*

Description

Creates a rose diagram of a circular data set on the current graphics device.

Usage

```
rose.diag(x, pch = 16, cex = 1, axes = TRUE, shrink = 1,
  bins = NULL, upper = TRUE, ticks = TRUE, tcl = 0.025, tcl.text = 0.125,
  radii.scale = c("sqrt", "linear"), border=NULL, col=NULL, tol = 0.04,
  uin = NULL, xlim = c(-1, 1), ylim = c(-1, 1), prop = 1, digits = 2,
  plot.info = NULL, units = NULL, template = NULL, zero = NULL,
  rotation = NULL, main = NULL, sub = NULL, xlab = "", ylab = "",
  add = FALSE, control.circle = circle.control(), ...)
```

Arguments

x	a vector, matrix or data.frame. The object is coerced to class circular.
pch	point character to use. See help on [par](par).
cex	point character size. See help on [par](par).
axes	logical: if TRUE axes are plotted according to properties of x.
shrink	parameter that controls the size of the plotted circle. Default is 1. Larger values shrink the circle, while smaller values enlarge the circle.
bins	number of arcs to partition the circle with.
upper	therose diagram cells are "upper"-closed intervals.
ticks	logical: if TRUE ticks are plotted according to the value of bins.
tcl	length of the ticks.
tcl.text	the position of the axis labels.
radii.scale	make possible to choose sector radius form: square-root of relative frequency (sqrt, default) or conventional linear scale (linear).
border	the color to draw the border. The default, NULL, means to use par("fg"). Use border = NA to omit borders.
col	the color for filling the rose diagram. The default, NULL, is to leave rose diagram unfilled. color of the points. The values are recycled if needed.
tol	proportion of white space at the margins of plot.
uin	desired values for the units per inch parameter. If of length 1, the desired units per inch on the x axis.
xlim, ylim	the ranges to be encompassed by the x and y axes. Useful for centering the plot.
prop	numerical constant determining the radii of the sectors. By default, prop = 1.
digits	number of digits used to print axis values.

plot.info	an object from `plot.circular` that contains information on the `zero`, the `rotation` and `next.points`.
units	the `units` used in the plot. If `NULL` the `units` of the first component of 'x' is used.
template	the template of the plot. Ignored if `plot.info` is provided.
zero	the zero of the plot. Ignored if `plot.info` or `template` are provided.
rotation	the rotation of the plot. Ignored if `plot.info` or `template` are provided.
main, sub, xlab, ylab	
	title, subtitle, x label and y label of the plot.
add	add the rose diag to an existing plot.
control.circle	parameters passed to `plot.default` in order to draw the circle. The function `circle.control` is used to set the parameters.
...	further parameters passed to `polygon`.

Details

The circumference of the circle is split into groups, the number of groups specified by bins. For each group, a sector is drawn. The radii of the sectors are by default equal to the square root of the relative frequencies of observations in each group. This ensures that the area of the sector is proportional to the group frequency. The length of the radii can be controlled by varying the parameter prop. Since version 0.3-9 the intervals are on the form [a,b).

Value

a list with information on the plot: zero, rotation and next.points.

Note

some codes from `eqscplot` in MASS is used. Since version 0.4-1 the meaning of the `col` parameter is changed.

Author(s)

Claudio Agostinelli, Ulric Lund and Hiroyoshi Arai

See Also

`plot.circular`

Examples

```
# Generate uniform data and create several rose diagrams.
# Some optional parameters may be needed to optimize plots.
x <- circular(runif(50, 0, 2*pi))
rose.diag(x, bins = 18, main = 'Uniform Data')
points(x)

# Generate von Mises data and create several rose diagrams.
```

```
x <- rvonmises(n=50, mu=circular(0), kappa=5, control.circular=list(zero=pi/4))
y <- rose.diag(x, bins=18) # Points fall out of bounds.
points(x, plot.info=y, stack=TRUE)
y <- rose.diag(x, bins=18, prop=1.5, shrink=1.5) # Adjust optional parameters to fit
######## all points on plot.
points(x, plot.info=y, stack=TRUE)

# Add the rose diag to a plot
plot(x)
rose.diag(x, bins=12, add=TRUE, col=2)

# Examples on using radii.scale and prop with a dummy dataset where
# highest proportion is 50% in bin 2
x <- c(2, 2, 2, 2, 5, 5, 10, 20)
circ.x <- circular::circular(x, units = "hours", template = "clock24")
old_par <- par(mfrow = c(2, 2))
rose.diag(circ.x, bins=24, main="radii.scale=linear, prop=1",
          radii.scale="linear", prop=1)
rose.diag(circ.x, bins=24, main = "radii.scale=linear, prop=2",
          radii.scale="linear", prop=2)
rose.diag(circ.x, bins=24, main = "radii.scale=sqrt, prop=1",
          radii.scale="sqrt", prop=1)
rose.diag(circ.x, bins=24, main = "radii.scale=sqrt, prop=sqrt(2)",
          radii.scale="sqrt", prop=sqrt(2))
par(old_par)
```

---

rstable   *Random Generation from the Stable Family of Distributions*

Description

Returns random deviates from the stable family of probability distributions.

Usage

```
rstable(n, scale = 1, index = stop("no index arg"), skewness = 0)
```

Arguments

n	sample size.
index	number from the interval (0, 2]. An index of 2 corresponds to the normal, 1 to the Cauchy. Smaller values mean longer tails.
skewness	number giving the modified skewness (see Chambers et al., 1976). Negative values correspond to skewness to the left (the median is smaller than the mean, if it exists), and positive values correspond to skewness to the right (the median is larger than the mean). The absolute value of skewness should not exceed 1.
scale	the scale of the distribution.

Details

This function return random variates from the Levy skew stable distribution with index=α, scale=c and skewness=β. The skewness parameter must lie in the range [-1,1] while the index parameter must lie in the range (0,2]. The Levy skew stable probability distribution is defined by a Fourier transform,

$$p(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} dt \exp(-itx - |ct|^\alpha (1 - i\beta sign(t) \tan(\pi\alpha/2)))$$

When $\alpha = 1$ the term $\tan(\pi\alpha/2)$ is replaced by $-(2/\pi) \log|t|$. For $\alpha = 2$ the distribution reduces to a Gaussian distribution with $\sigma = \sqrt{2}scale$ and the skewness parameter has no effect. For $\alpha < 1$ the tails of the distribution become extremely wide. The symmetric distribution corresponds to $\beta = 0$.

The Levy alpha-stable distributions have the property that if N alpha-stable variates are drawn from the distribution $p(c, \alpha, \beta)$ then the sum $Y = X_1 + X_2 + \cdots + X_N$ will also be distributed as an alpha-stable variate, $p(N^{1/\alpha}c, \alpha, \beta)$.

There is no explicit solution for the form of $p(x)$ and there are no density, probability or quantile functions supplied for this distribution.

Value

random sample from the specified stable distribution.

Author(s)

Claudio Agostinelli

References

Chambers, J. M., Mallows, C. L. and Stuck, B. W. (1976). A Method for Simulating Stable Random Variables. Journal of the American Statistical Association 71, 340-344.

Logaeve, M. (1977). Probability Theory I. (fourth edition) Springer-Verlag, New York.

See Also

rnorm, rcauchy.

Examples

```
hist(rstable(200, 1.5, .5)) #fairly long tails, skewed right
```

---

rwrappedstable   *Random Generation from the Wrapped Stable Distribution*

Description

Generates pseudo-random numbers from a wrapped stable distribution.

Usage

```
rwrappedstable(n, scale=1, index, skewness, control.circular=list())
```

Arguments

n	number of random numbers to generate.
scale	the scale of the distribution.
index	number from the interval (0, 2]. An index of 2 corresponds to the normal, 1 to the Cauchy. Smaller values mean longer tails.
skewness	number giving the modified skewness. Negative values correspond to skewness to the left (the median is smaller than the mean, if it exists), and positive values correspond to skewness to the right (the median is larger than the mean). The absolute value of skewness should not exceed 1.
control.circular	
	the attribute of the resulting object.

Details

n random numbers are generated from a stable distribution with with parameters index, skewness and scale. The function returns these values modulo 2*pi.

Value

Returns a vector of n independent random numbers generated from a wrapped stable distribution.

Author(s)

Claudio Agostinelli

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 2.2.8, World Scientific Press, Singapore.

---

sd   *Standard Deviation*

Description

The sd function from the **base** is replace by a new method in order to report the standard deviation of circular data appropriately. sd.default is an alias of the original function sd see sd. The behavior would be the same for objects which are not from class data.frame and circular (in the last case the standard deviation is define as in Mardia (1972)

$$\sqrt{-2\ln r}$$

where r is the mean resultant length of the data, see sd.circular for more details). The method for data.frame will apply the sd function to each columns.

Usage

```
sd(x, ...)
## Default S3 method:
sd(x, na.rm = FALSE, ...)
## S3 method for class 'data.frame'
sd(x, ...)
```

Arguments

x   a numeric vector, matrix or data frame.

na.rm   logical. Should missing values be removed?

...   further arguments passed to or from other methods.

See Also

sd, sd.circular, var.circular and summary.circular.

---

sd.circular   *Circular Standard Deviation*

Description

Returns the circular standard deviation of a vector of circular data which is defined as the square root of minus 2 times the log of the mean resultant length divided by the number of observations.

Usage

```
## S3 method for class 'circular'
sd(x, na.rm = FALSE, ...)
```

Arguments

x	a vector. The object is coerced to class [circular]().
na.rm	logical, indicating if [NA]()'s should be omitted.
...	further arguments passed to or from other methods.

Details

Computes the circular standard deviation as defined by Mardia (1972)

$$\sqrt{-2 \ln r}$$

where r is the mean resultant length of the data.

Value

Returns the circular standard deviation.

Author(s)

Claudio Agostinelli and Jean-Olivier Irisson

References

Mardia, K.V. (1972) Statistics of Directional Data. Academic Press, London, sec. 26.5, p. 617

Fisher, N.I. (1993) Statistical analysis of circular data. Cambridge University Press.

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 1.3, World Scientific Press, Singapore.

Zar, J H (2010). Biostatistical analysis. Prentice Hall. sec. 26.5, p. 617

See Also

[var.circular](), [angular.deviation](), [mean.circular](), [rho.circular]() and [summary.circular]().

Examples

```
# Compute the circular standard deviation of a random
# sample of observations from a von Mises distribution
x <- rvonmises(n=100, mu=circular(0, units="degrees"), kappa=10)
sd(x)
```

---

summary.circular   *Circular Summary Statistics*

Description

Computes circular summary statistics including the sample size, mean direction and mean resultant length and quartiles.

Usage

```
## S3 method for class 'circular'
summary(object,
    digits = max(3, getOption("digits") - 3), ...)
```

Arguments

object	an object of class `circular`.
digits	digits to be used in printing.
...	parameters passed to `summary.matrix` if needed.

Details

Each observation is treated as a unit vector or a point on the unit circle. The resultant vector of the observations is found, and the direction of the resultant vector is returned as well as its length divided by the sample size.

Value

Returns a vector with the sample size, the sample mean direction and the sample mean resultant length.

Author(s)

Claudio Agostinelli, David Andel and Alessandro Gagliardi

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 1.3, World Scientific Press, Singapore.

See Also

`mean.circular`, `median.circular`, `quantile.circular`, `var.circular`, `mle.vonmises`, `rho.circular`.

Examples

```
# Compute summary statistics of a random sample of observations.
data <- circular(runif(50, 0, pi))
summary(data)
summary(data.frame(data, runif(50, 0, pi)))
```

---

swallows   *Orientation of juvenile barn swallows*

Description

The _swallows_ dataset has 114 rows and 2 columns. The observations are the headings of juvenile barn swallows (_Hirundo rustica_) tested in orientation cages (Emlen funnels) during autumn migration under simulated overcast conditions.

Usage

```
data(swallows)
```

Format

A data frame with 114 observations on the following 2 variables.

treatment a factor with levels control (control group: local magnetic field) and shifted (shifted magnetic field, magnetic North = geographical West)

heading a numeric vector: modal heading of each bird

Source

Giunchi, D., and Baldaccini N. E. (2004) Orientation of juvenile barn swallows (Hirundo rustica) tested in Emlen funnels during autumn migration. Behav. Ecol. Sociobiol. (56):124-131.

Examples

```
data(swallows)

swallows <- split(swallows$heading, swallows$treatment)
swallows <- lapply(swallows, function(x) circular(x, units='degrees', template='geographics'))

plot(swallows[[1]])
points(swallows[[2]], col=2)
legend('topright', legend=c('control', 'shifted'), pch=c(19,19), col=c(1,2))
```

---

ticks.circular   *Draw Tick-Marks in a Circular Plot*

Description

Draw tick-marks in a circular plot.

Usage

```
ticks.circular(x, template = c("none", "geographics"), zero = NULL,
  rotation = NULL, tcl = 0.025, col = NULL, ...)
```

Arguments

x	the points at which tick-marks are to be drawn.
template	either none or geographics.
zero	the zero of the plot (in radians).
rotation	the rotation of the plot.
col	color for the tick marks. If NULL, function uses 'par("col.axis")'.
tcl	The length of tick marks.
...	parameters passed to line.default.

Author(s)

Claudio Agostinelli

See Also

[plot.circular](#) and [axis.circular](#).

---

totalvariation.circular

Conditional total variation distance between two circular samples.

Description

The total variation distance between two circular samples is evaluated conditional on a circular modal region.

Usage

```
totalvariation.circular(x, y, z = NULL, q = 0.95, bw, adjust = 1,
type = c("K", "L"), kernel = c("vonmises", "wrappednormal"),
na.rm = FALSE, step = 0.001, eps.lower = 10^(-4), eps.upper = 10^(-4), ...)
```

Arguments

x	numeric or an object of class [circular](#).
y	numeric or an object of class [circular](#).
z	numeric or object of class [circular](#). The grid were the kernel density estimate will be evaluated. If NULL equally spaced points in the interval [0,2*pi) with step step.
q	numeric in the interval [0,1]. The quantile of the modal region.
bw	the smoothing bandwidth to be used. When the kernel is vonmises the bandwidth is equal to the concentration parameter.
adjust	the bandwidth used is actually adjust*bw. This makes it easy to specify values like "half the default bandwidth".
type	Not Yet Used.
kernel	a character string giving the smoothing kernel to be used. This must be one of "vonmises" or "wrappednormal", that are kernels of type "K".
na.rm	logical; if TRUE, missing values are removed from x. If FALSE any missing values cause an error.
step	numeric. Used in the construction of the regular grid z.
eps.lower, eps.upper	
	the cut point in the density is searched in the interval [min(density)*(1+eps.lower),max(density)*(1-eps.upper)].
...	further arguments passed to the modal.region.circular function. Not used at present.

Value

A list of class totalvariation.circular with the following components

tv	the (conditional) total variation.
ovl	the (conditional) overlapping coefficient.
q	the order of the modal regions.
bw	the bandwidth value as in input.
modal.x	an object of class [modal.region.circular](#) for the x data set.
modal.y	an object of class [modal.region.circular](#) for the y data set.
density.x	an object of class [density.circular](#) for the x data set.
density.y	an object of class [density.circular](#) for the y data set.
density	a function which report the positive part of the difference between the estimated density of the two data sets.

Author(s)

Claudio Agostinelli

References

L.G.R. Oliveira-Santos, C.A. Zucco and C. Agostinelli (2013) Using conditional circular kernel density functions to test hypotheses on animal circadian activity. Animal Behaviour, 85(1) 269-280.

See Also

modal.region.circular

Examples

```
x <- rvonmises(100, circular(pi), 10)
y <- rvonmises(100, circular(pi+pi/8), 10)
res <- totalvariation.circular(x,y,bw=50)
plot(res)
```

---

triangular	*Triangular Density Function*

Description

Density and random generation for the Triangular circular distribution.

Usage

```
dtriangular(x, rho)
rtriangular(n, rho, control.circular=list())
```

Arguments

x	a vector. The object is coerced to class circular.
n	number of observations.
rho	concentration parameter of the distribution. rho must be between 0 and $4/pi^2$.
control.circular	
	the attribute of the resulting object.

Value

dtriangular gives the density and rtriangular generates random deviates.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 2.2.3, World Scientific Press, Singapore.

Examples

```
data1 <- rtriangular(100, 0.3, control.circular=list(units="degrees"))
plot(data1)

ff <- function(x) dtriangular(x, rho=0.3)
curve.circular(ff, shrink=1.2, join=TRUE)
```

---

trigonometric.moment   *Trigonometric Moments*

Description

Computes the specified order trigonometric moment for a set of directional data points.

Usage

```
trigonometric.moment(x, p = 1, center = FALSE, control.circular = list())
```

Arguments

x   a vector of class `circular`.

p   order of trigonometric moment to be computed. Default is for a first order
                   trigonometric moment.

center   logical, whether to compute centered moments or not. Default is to compute an
                   uncentered moment.

control.circular
                   the attribute of the resulting object `mu`.

Value

Returns a list with variables mu, rho, cos, sin, p, n, call, respectively the pth trigonometric moment's
direction, resultant length, real and imaginary components, the order, the number of observations
and the call.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 1.3, World
Scientific Press, Singapore.

See Also

[var.circular](#), [mean.circular](#), [summary.circular](#), [mle.vonmises](#) and [rho.circular](#)

Examples

```
x <- rvonmises(100, circular(0), 5)
trigonometric.moment(x, control.circular=list(units="degrees"))
```

---

turtles   *Arrival directions of displaced sea turtles*

Description

The _turtles_ dataset has 10 rows and 2 columns. The observations are the directions from which 10 green sea turtles approached their nesting island (Ascension Island, South Atlantic Ocean) after having been displaced to open-sea sites.

Usage

```
data(turtles)
```

Format

A data frame with 10 observations on the following 2 variables.

id a numeric vector: the turtle ID

arrival a numeric vector: the direction of arrival to Ascension Island

Source

Luschi, P., Akesson, S., Broderick, A. C., Glen, F., Godley, B. J., Papi F., and Hays, G. C. (2001) Testing the navigational abilities of ocean migrants: displacement experiments on green sea turtles (_Chelonia mydas_). Behav. Ecol. Sociobiol. (50):528-534.

Examples

```
data(turtles)
turtles[,2] <- circular(turtles[,2], units='degrees', template='geographics')
plot(turtles[,2])
```

---

unique.circular   *Extract Unique Elements from a circular vector*

Description

unique.circular returns a circular vector but with duplicate elements removed.

Usage

```
## S3 method for class 'circular'
unique(x, ...)
```

Arguments

x   a vector or a data frame or an array or NULL.

...   parameters passed to unique.default

Details

This is a method for circular object. See the documentation of unique.

Value

An object of the same type of x, but if an element is equal to one with a smaller index, it is removed.

See Also

unique

Examples

```
x <- rvonmises(10, circular(0), 10)
unique(x)
```

---

var   *Variance*

Description

The var function from the **stats** is replace by a new method in order to report the variance of circular data appropriately. var.default is an alias of the original function var see cor. The behavior would be the same for objects which are not from class data.frame and circular (in the last case the variance is define as one minus the mean resultant length divided by the sample size of data, see var.circular for more details). The method for data.frame will apply the var function to each columns.

Usage

```
var(x, ...)
## Default S3 method:
var(x, y = NULL, na.rm = FALSE, use, ...)
## S3 method for class 'data.frame'
var(x, ...)
```

Arguments

x	a numeric vector, matrix or data frame.
y	NULL (default) or a vector, matrix or data frame with compatible dimensions to x. The default is equivalent to y = x (but more efficient).
na.rm	logical. Should missing values be removed?
use	an optional character string giving a method for computing covariances in the presence of missing values. This must be (an abbreviation of) one of the strings "all.obs", "complete.obs" or "pairwise.complete.obs".
...	further arguments passed to or from other methods.

See Also

[cor](), [var.circular](), [rho.circular]() and [summary.circular]().

---

var.circular	*A measure of variance for Circular Data*

Description

Returns one minus the mean resultant length divided by the sample size of a vector of circular data.

Usage

```
## S3 method for class 'circular'
var(x, na.rm = FALSE, ...)
```

Arguments

x	a vector. The object is coerced to class [circular]().
na.rm	logical, indicating if [NA]()'s should be omitted.
...	further arguments passed to or from other methods.

Value

Returns one minus the mean resultant length divided by the sample size of data.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Mardia, K.V. (1972) Statistics of Directional Data. Academic Press, London.

Fisher, N.I. (1993) Statistical analysis of circular data. Cambridge University Press.

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 1.3, World Scientific Press, Singapore.

See Also

`sd.circular`, `angular.variance`, `mean.circular`, `rho.circular` and `summary.circular`.

Examples

```
x <- rvonmises(n=100, mu=circular(0), kappa=1)
var(x)
```

---

vonMises   *von Mises Density Function*

Description

Density, distribution function, random generation and quantiles for the von Mises circular distribution.

Usage

```
rvonmises(n, mu, kappa, control.circular=list())
dvonmises(x, mu, kappa, log)
pvonmises(q, mu, kappa, from=NULL, tol = 1e-020)
qvonmises(p, mu = circular(0), kappa=NULL, from=NULL, tol = .Machine$double.eps^(0.6),
  control.circular = list(), ...)
```

Arguments

x, q, p	a vector. The x and q objects are coerced to class `circular`.
n	number of observations.
mu	mean direction of the distribution. The object is coerced to class `circular`.
kappa	non-negative numeric value for the concentration parameter of the distribution.
log	logical; if TRUE, probabilities p are given as log(p).
from	if NULL is set to $mu - pi$. This is the value from which the pvonmises and qvonmises are evaluated. It should be a `circular` object.
tol	the precision in evaluating the distribution function or the quantile.

control.circular

> the attribute of the resulting object.

... parameters passed to integrate.

Value

dvonmises gives the density, pvonmises gives the distribution function, rvonmises generates random deviates and qvonmises provides quantiles.

Since version 0.3-5 the random deviates are generated using a C code.

Author(s)

Claudio Agostinelli, Ulric Lund and Harry Southworth

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 2.2.4, World Scientific Press, Singapore.

Examples

```
data1 <- rvonmises(100, circular(0), 10, control.circular=list(units="degrees"))
plot(data1)

ff <- function(x) dvonmises(x, mu=circular(pi), kappa=10)
curve.circular(ff, join=TRUE, xlim=c(-2.3, 1),
  main="Density of a VonMises Distribution \n mu=pi, kappa=10")

ff <- function(x) pvonmises(x, mu=circular(pi), kappa=10)
curve.circular(ff, join=FALSE, xlim=c(-2, 2), ylim=c(-2, 1),
  to=(2*pi-3*.Machine$double.eps), modulo="asis", nosort=TRUE,
  main="Probability of a VonMises Distribution \n mu=pi, kappa=10")

plot(function(x) qvonmises(x, mu=circular(0), kappa=5), from=0, to=1)
##curve do not work!
plot(function(x) qvonmises(x, mu=circular(pi), kappa=5), from=0, to=1)
plot(function(x) qvonmises(x, mu=circular(pi), kappa=5, from=circular(pi/2)), from=0, to=1)
```

---

wallraff.test   *Wallraff Test of Angular Distances*

Description

Performs the Wallraff test of angular distances or angular dispersion around the mean.

Usage

```
wallraff.test(x, ...)

## Default S3 method:
wallraff.test(x, group, ref=NULL, ...)

## S3 method for class 'list'
wallraff.test(x, ref=NULL, ...)

## S3 method for class 'formula'
wallraff.test(formula, data, ref=NULL, ...)
```

Arguments

x	a vector of angles (coerced to class [circular](#)) or a list of such angles. When x is a list, its elements are taken as the samples to be compared.
group	a vector or factor object giving the group for the corresponding elements of x. Ignored if x is a list
formula	a formula of the form lhs ~ rhs where lhs is a vector of angles and rhs a vector or factor giving the corresponding groups.
data	an optional data.frame containing the variables in the formula [formula](#).
ref	a vector of angles used as reference to compute the angular distances from, in each group. It should contain as many elements as there are groups, *in the same order*.
	If x is a list, the order is the order of the elements of the list.
	In the default or formula interfaces, if the grouping vector is a factor, the order is the order of its levels; if the grouping vector is not a factor, it is coerced as such but with levels in the order of their appearance in the original vector. In this case a warning is issued to make sure the order of ref is correct.
	If ref has less elements than the number of groups (typically one: a common reference for all groups), it is recycled to match the number of groups.
	If ref is NULL (the default), the mean angle of each group will be used as reference. In this situation, the Wallraff test becomes a comparison of angular dispersion around the mean.
...	further arguments passed to or from other methods.

Details

The Wallraff test of angular distances between two or more groups is performed and the results are printed. The null hypothesis is that distances are equal across groups.

The test proceeds by computing the angular distances from a reference angle, in each group. The angular distance between two angles is the circular range and is computed with [range.circular](#). Then the distances are compared with a usual rank sum test (Kruskal-Wallis, [kruskal.test](#)). When there are only two groups, the Wilcoxon-Mann-Whitney test could be used but [wilcox.test](#) without continuity correction for the p-value is equivalent to [kruskal.test](#) so only [kruskal.test](#) is used here.

The Wallraff test is most frequently used to compare angular dispersion around the mean, between samples. In this case, the reference angle is the mean angle of each sample. This is the default here, when no reference angles are provided.

All angles should be of class circular and will be coerced as such with the default parameters if they are not. An exception are the reference angles in ref. For ease of use, those can be only numeric and are then considered to be in the same angular reference as x.

Value

A list with class "htest" containing the following components:

statistic	the chi-squared statistic from kruskal.test.
parameter	the degrees of freedom for the chi-squared statistic.
p.value	the p-value for the test.
method	a character string containing the name of the test.
data.name	a character string giving the name(s) of the data.

Author(s)

Jean-Olivier Irisson

References

Batschelet, E (1981). Circular Statistics in Biology. chap. 6.10, p. 124

Zar, J H (2010). Biostatistical analysis. sec. 27.7-8, p. 643

See Also

kruskal.test for the Kruskal-Wallis rank sum test used on the angular distances.

wilcox.test for the two samples alternative to the Kruskal-Wallis test.

Examples

```
# Homing of pigeons
# Example used in Batschelet (1981)
data <- list(
  control = circular(c(70, 80, 80, 85, 85, 90, 95, 95),
        units="degrees", template="geographics"),
  experimental = circular(c(5, 5, 15, 55, 55, 65, 105, 120, 340),
            units="degrees", template="geographics")
)

# compare the angular dispersion between the two groups
wallraff.test(data)

# compare the homing performance
# home azimuth is 40 degrees for both groups
wallraff.test(data,
              ref = circular(c(40, 40), units="degrees",
```

```
                                          template="geographics")
            )
# we could have more simply used
wallraff.test(data, ref=40)
# because ref is automatically repeated and considered
# in the same circular reference as the data
```

---

watson.test   *Watson's Test*

Description

Performs a Watson's goodness of fit test for the von Mises or circular uniform distribution.

Usage

```
watson.test(x, alpha=0, dist=c("uniform", "vonmises"))
## S3 method for class 'watson.test'
print(x, digits = 4, ...)
```

Arguments

x	a vector. The object is coerced to class [circular]().
alpha	significance level of the test. Valid levels are 0.01, 0.05, 0.1. This argument may be omitted, in which case, a range for the p-value will be returned.
dist	distribution to test for. The default is the uniform distribution. To test for the von Mises distribution, set dist to "vonmises".
digits	integer indicating the precision to be used.
...	further arguments passed to or from other methods.

Details

If dist = "uniform", Watson's one-sample test for the circular uniform distribution is performed, and the results are printed. If alpha is specified and non-zero, the test statistic is printed along with the critical value and decision. If alpha is omitted, the test statistic is printed and a range for the p-value of the test is given.

If dist = "vonmises", estimates of the population parameters are used to evaluate the von Mises distribution function at all data points, thereby arriving at a sample of approximately uniformly distributed data, if the original observations have a von Mises distribution. The one-sample Watson test is then applied to the transformed data as above.

Value

a list with the statistic, alpha, the number of observations, the distribution and 'row' which is used by print.watson.test to evaluate the p-value.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 7.2, World Scientific Press, Singapore.

Stephens, M. (1970). Use of the Kolmogorov-Smirnov, Cramer-von Mises and related statistics without extensive tables. Journal of the Royal Statistical Society, B32, 115-122.

See Also

[range.circular](#), [kuiper.test](#), [rao.spacing.test](#) and [rayleigh.test](#)

Examples

```
# Generate data from the uniform distribution on the circle.
x <- circular(runif(100, 0, 2*pi))
watson.test(x)
# Generate data from a von Mises distribution.
x <- rvonmises(n=50, mu=circular(0), kappa=4)
watson.test(x, alpha=0.05, dist="vonmises")
```

---

watson.two.test	*Watson's Two-Sample Test of Homogeneity*

Description

Performs Watson's test for homogeneity on two samples of circular data.

Usage

```
watson.two.test(x, y, alpha=0)
## S3 method for class 'watson.two.test'
print(x, digits=4, ...)
```

Arguments

x	a vector. The object is coerced to class [circular](#).
y	a vector. The object is coerced to class [circular](#).
alpha	significance level of the test. Valid levels are 0.001, 0.01, 0.05, 0.1. This argument may be omitted, in which case, a range for the p-value will be returned.
digits	integer indicating the precision to be used.
...	further arguments passed to or from other methods.

Details

Watson's two-sample test of homogeneity is performed, and the results are printed. If alpha is specified and non-zero, the test statistic is printed along with the critical value and decision. If alpha is omitted, the test statistic is printed and a range for the p-value of the test is given.

Critical values for the test statistic are obtained using the asymptotic distribution of the test statistic. It is recommended to use the obtained critical values and ranges for p-values only for combined sample sizes in excess of 17. Tables are available for smaller sample sizes and can be found in Mardia (1972) for instance.

Value

a list with statistic, alpha and the number of observations of the first and second sample.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 7.5, World Scientific Press, Singapore.

Examples

```
# Perform a two-sample test of homogeneity on two
# simulated data sets.
data1 <- rvonmises(n=20, mu=circular(0), kappa=3)
data2 <- rvonmises(n=20, mu=circular(pi), kappa=2)
watson.two.test(data1, data2, alpha=0.05)
watson.two.test(data1, data2)
```

---

watson.wheeler.test   *Watson-Williams Test of Homogeneity of Means*

Description

Performs the Watson-Wheeler test for homogeneity on two or more samples of circular data.

Usage

```
watson.wheeler.test(x, ...)

## Default S3 method:
watson.wheeler.test(x, group, ...)

## S3 method for class 'list'
watson.wheeler.test(x, ...)
```

```
## S3 method for class 'formula'
watson.wheeler.test(formula, data, ...)
```

Arguments

x	a vector of angles (coerced to class [circular](#)) or a list of such angles.
group	a vector or factor object giving the groups for the corresponding elements of x. Ignored if x is a list
formula	a formula of the form lhs ~ rhs where lhs is a vector of angles and rhs a vector or factor giving the corresponding groups.
data	an optional data.frame containing the variables in the formula [formula](#).
...	further arguments passed to or from other methods.

Details

The Watson-Wheeler (or Mardia-Watson-Wheeler, or uniform score) test is a non-parametric test to compare two or several samples. The difference between the samples can be in either the mean or the variance.

The *p*-value is estimated by assuming that the test statistic follows a chi-squared distribution. For this approximation to be valid, all groups must have at least 10 elements.

In the default method, x is a vector of angles and group must be a vector or factor object of the same length as x giving the group for the corresponding elements of x.

If x is a list, its elements are taken as the samples to be compared.

In the [formula](#) method, the angles and grouping elements are identified as the left and right hand side of the formula respectively.

All angles should be of class [circular](#) and will be coerced as such if they are not.

Value

A list with class "htest" containing the following components:

statistic	W, the statistic of the test, which is approximately distributed as a chi-squared.
parameter	the degrees of freedom for the chi-squared approximation of the statistic.
p.value	the p-value for the test.
method	a character string containing the name of the test.
data.name	a character string giving the name(s) of the data.

Author(s)

Jean-Olivier Irisson

References

Batschelet, E (1981). Circular Statistics in Biology. chap 6.3, p. 104

Zar, J H (1999). Biostatistical analysis. section 27.5, p. 640

Examples

```
# Example used in Zar (1999)
x1 <- circular(c(35, 45, 50, 55, 60, 70, 85, 95, 105, 120),
  units="degrees", template="geographics")
x2 <- circular(c(75, 80, 90, 100, 110, 130, 135, 140, 150, 160, 165),
  units="degrees", template="geographics")

watson.wheeler.test(list(x1,x2))
```

---

watson.williams.test   *Watson-Williams Test of Homogeneity of Means*

Description

Performs the Watson-Williams test for homogeneity of means between several samples of circular data.

Usage

```
watson.williams.test(x, ...)

## Default S3 method:
watson.williams.test(x, group, ...)

## S3 method for class 'list'
watson.williams.test(x, ...)

## S3 method for class 'formula'
watson.williams.test(formula, data, ...)
```

Arguments

x	a vector of angles (coerced to class [circular](#)) or a list of such angles.
group	a vector or factor object giving the group for the corresponding elements of x. Ignored if x is a list
formula	a formula of the form lhs ~ rhs where lhs is a vector of angles and rhs a vector or factor giving the corresponding groups.
data	an optional data.frame containing the variables in the formula [formula](#).
...	further arguments passed to or from other methods.

Details

The Watson-Williams test for the homogeneity of means between two or more groups is performed and the results are printed. The null hypothesis is that means are equal across groups.

The assumptions are that: (1) the samples are drawn from populations with a von Mises distribution; (2) the parameter of concentration has the same value in all populations; (3) this parameter is

sufficiently large (i.e. > 1). Assumptions 2 and 3 are checked and a warning is issued if they are not met.

In the default method, x is a vector of angles and group must be a vector or factor object of the same length as x giving the group for the corresponding elements of x.

If x is a list, its elements are taken as the samples to be compared.

In the `formula` method, the angles and grouping elements are identified as the left and right hand side of the formula respectively.

All angles should be of class `circular` and will be coerced as such if they are not.

Value

A list with class `"htest"` containing the following components:

statistic	the F statistic of the test.
parameter	the degrees of freedom for the F statistic.
p.value	the p-value for the test.
estimate	a vector of the means of each group.
method	a character string containing the name of the test.
data.name	a character string giving the name(s) of the data.

Author(s)

Jean-Olivier Irisson

References

Batschelet, E (1981). Circular Statistics in Biology. chap. 6.2, p. 99

Mardia, KV and Jupp, PE (2000). Directional statistics. p. 135

Examples

```
# Ant orientation from Duelli and Wehner (1973)
# Example used in Batschelet (1981)
data <- list(
  exp = circular(rep(c(-20, -10, 0), c(1,7,2)),
        units="degrees", template="geographics"),
  control = circular(rep(c(-10, 0, 10, 20), c(3,3,3,1)),
           units="degrees", template="geographics")
)

watson.williams.test(data)
```

---

weighted.mean.circular

Weighted Mean Direction

Description

Returns the weighetd mean direction of a vector of circular data.

Usage

```
## S3 method for class 'circular'
weighted.mean(x, w, na.rm=FALSE,
  control.circular=list(), ...)
```

Arguments

x	a vector. The object is coerced to class [circular]().
w	a numerical vector of weights the same length as x giving the weights to use for elements of x.
na.rm	logical, indicating if [NA]()'s should be omitted.
control.circular	
	the attribute of the resulting object.
...	further arguments passed to or from other methods.

Details

Each observation is treated as a unit vector, or point on the unit circle. The resultant vector of the observations is found, and the direction of the resultant vector is returned. An [NA]() is returned if the weighted resultant length is less than [.Machine]().

If w is missing then all elements of x are given the same weight, otherwise the weights coerced to numeric by as.numeric and normalized to sum to one.

Missing values in w are not handled specially and so give a missing value as the result. However, zero weights are handled specially and the corresponding x values are omitted from the computation.

Value

Returns the weighted mean direction of the data as an object of class circular with the attribute given by control.circular or from x if missed in control.circular.

Author(s)

Claudio Agostinelli

See Also

[mean.circular]()

Examples

```
# Compute the weighted mean direction of a random sample of observations.
x <- circular(runif(50, circular(0), pi))
w <- runif(50, 0, 1)
weighted.mean(x, w)
```

---

wind   *Col De La Roa wind direction*

Description

In a place named "Col de la Roa" in the Italian Alps there is a meteorological station that records via data-logger several parameters. Measures are made every 15 minutes, in this dataset we report the wind direction recorded every day from January 29, 2001 to March 31, 2001 from 3.00am to 4.00am included. Which means 5 observations every day for a total of 310 measures.

Usage

```
data(wind)
```

Format

This data frame contains one variables (wind direction) in radians.

Source

http://www.tesaf.unipd.it/SanVito/dati.htm

References

C. Agostinelli (2007) Robust estimation for circular data, Computational Statistics and Data Analysis, 51(12), 5867-5875, doi = doi:10.1016/j.csda.2006.11.002

Examples

```
data(wind)
wind <- circular(wind, template='geographics')
par(mfcol=c(1,2))
plot(wind)
plot(density(wind, bw=40), main='')
```

---

windrose   *Windrose Generator*

Description

This function creates a windrose used to visualize the direction and magnitude of wind. The pedals of a windrose indicate the proportion of time wind comes from a given direction. Bands on the windrose indicate the proportions of winds of each magnitude.

Usage

```
windrose(x, y=NULL, breaks=NULL, bins=12, increment = 10,
  main='Wind Rose', cir.ind = 0.05, fill.col=NULL, plot.mids=TRUE,
  mids.size=1.2, osize=0.1, axes=TRUE, ticks=TRUE, tcl=0.025,
  tcl.text=-0.15, cex=1, digits=2, units=NULL,
  template=NULL, zero=NULL, rotation=NULL,
  num.ticks=12, xlim=c(-1.2, 1.2), ylim=c(-1.2, 1.2),
  uin, tol=0.04, right=FALSE, shrink=NULL,
  label.freq=FALSE, calm=c("0", "NA"), ...)
```

Arguments

x	a vector contains direction or a two columns data frame, where the first component is the direction and the second the magnitude. The vector or the first column in the case of data frame is coerced to class `circular`.
y	a vector contains magnitude. If 'y' is not NULL and 'x' is a dataframe, only the first column of 'x' is used for direction.
breaks	the extremes of the pedals. The biggest value (in 2*pi) is recycled for building the first pedal. The vector is coerced to class `circular` but only the units is used.
bins	Number of pedals. Ignored if 'breaks' is not NULL.
increment	Grouping size of magnitude. These are the bins of the magnitudes displayed on each pedal.
main	Title for plot.
cir.ind	Percent intervals expressed on each circle if the pedals are equally spaced, otherwise values of density
fill.col	colors used to fill the pedals for each magnitude. The colors are recycled if necessary. The default is to use 'blue' and 'red'.
plot.mids	plot lines at the midpoints of the pedals.
mids.size	length of the lines for midpoints.
osize	radius of the circle draws at the center of the plot.
axes	if TRUE axes are added to the plot. The function `axis.circular` is used.
ticks	if TRUE ticks are added to the plot. The function `ticks.circular` is used.

`tcl`	length of the ticks.
`tcl.text`	The position of the axis labels.
`cex`	point character size. See help on [`par`]().
`digits`	number of digits used to print axis values and other numbers.
`units`	the units used in the plot.
`template`	the template used in the plot.
`zero`	the zero used in the plot.
`rotation`	the rotation used in the plot.
`num.ticks`	number of tick marks draw.
`tol`	proportion of white space at the margins of plot
`uin`	desired values for the units per inch parameter. If of length 1, the desired units per inch on the x axis.
`xlim, ylim`	the ranges to be encompassed by the x and y axes. Useful for centering the plot.
`right`	logical; if TRUE, the pedals are right-closed (left open) intervals.
`shrink`	maximum length of the pedals, it can be used to plot several graphics with the same scale.
`label.freq`	logical; if TRUE, the relative frequencies are used in the magnitude instead of intensities, when the breaks are equally spaced.
`calm`	"0" or "NA", see details below.
`...`	further parameters ignored for now.

Details

Following the convention of the National Weather Service, winds with a direction of 0 are considered calm, while winds with a direction of 360 degrees (2*pi radians) are assumed to be from the north. Calm winds are excluded from the wind rose creation. We allow, in direction, to use NA to indicate calm wind (argument `calm`).

This wind rose preserve areas of pedals, that is counts are proportional to the area of the pedals rather than to the length of the pedals. This is also for the slides created for the magnitudes.

Value

`x`	directions
`y`	magnitudes
`table`	Matrix output of the counts of wind direction and magnitude. Columns are in the same units as the data, according to step size, and rows are based on the increment size.
`number.obs`	Total number of observations.
`number.calm`	The number of calm observations omitted from the wind rose plot.
`breaks`	extremes of the pedals.
`mids`	midpoints of pedals.
`call`	the [`match.call`]() result.

Note

some codes from [eqscplot](#) in 'MASS' is used.

Author(s)

Matt Pocernich <pocernic@rap.ucar.edu>, ported in the package 'circular' by Claudio Agostinelli

Examples

```
# Random distribution of direction and magnitude in degrees

dir <- circular(runif(100, 0, 360), units="degrees")
mag <-  rgamma(100, 15)
sample <- data.frame(dir=dir, mag=mag)

par(mfrow=c(2,2))
res <- windrose(sample)
## we join two pedals and keep the same shrink (scale of the plot)
breaks <-circular(seq(0, 2 * pi, by = pi/6))
breaks <- breaks[-2]
windrose(sample, breaks=breaks, main="The same but with two pedals joined",
  shrink=res$shrink)
## change the rotation
sample <- data.frame(dir=circular(dir, units="degrees", rotation="clock"), mag=mag)
windrose(sample, breaks=breaks, main="Change the rotation", shrink=res$shrink)
## use geographics template
sample <- data.frame(dir=circular(dir, units="degrees", template="geographics"),
  mag=mag)
windrose(sample, breaks=breaks, main="Use the template 'geographics'",
  shrink=res$shrink)

## do the same plot but in radians
dir <- conversion.circular(dir)
windrose(x=dir, y=mag, xlim=c(-1.3, 1.3))

## magnify some part of the plot
windrose(x=dir, y=mag, xlim=c(0, 1.3))
```

---

wrappedCauchy   *Wrapped Cauchy Density Function*

Description

Density, and random generation for the wrapped Cauchy circular distribution.

Usage

```
dwrappedcauchy(x, mu = circular(0), rho = exp(-1))
rwrappedcauchy(n, mu = circular(0), rho = exp(-1), control.circular=list())
```

Arguments

x	a vector. The object is coerced to class [circular]().
n	number of observations.
mu	mean direction of the distribution as a `circular` object.
rho	concentration parameter of the distribution. `rho` must be in the interval from 0 to 1.
control.circular	
	the attribute of the resulting object.

Value

dwrappedcauchy gives the density and rwrappedcauchy generates random deviates.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 2.2.7, World Scientific Press, Singapore.

Examples

```
data1 <- rwrappedcauchy(100, mu=circular(0), rho=0.7,
  control.circular=list(units="degrees"))
plot(data1)

ff <- function(x) dwrappedcauchy(x, mu=circular(pi), rho=0.7)
curve.circular(ff, join=TRUE, xlim=c(-2, 1),
  main="Density of a Wrapped Cauchy Distribution \n mu=pi, rho=0.7")
```

---

wrappedNormal	*Wrapped Normal Density Function*

Description

Density, and random generation for the wrapped normal circular distribution.

Usage

```
rwrappednormal(n, mu = circular(0), rho = NULL, sd = 1,
  control.circular = list())
dwrappednormal(x, mu = circular(0), rho = NULL, sd = 1,
  K = NULL, min.k = 10)
pwrappednormal(q, mu = circular(0), rho = NULL, sd = 1,
  from = NULL, K = NULL, min.k = 10, ...)
qwrappednormal(p, mu = circular(0), rho = NULL, sd = 1,
  from = NULL, K = NULL, min.k = 10, tol = .Machine$double.eps^(0.6),
  control.circular = list(), ...)
```

Arguments

x, q	vector of quantiles. The object is coerced to class [circular](#).
p	vector of probabilities.
n	number of observations.
mu	mean direction of the distribution as a circular object.
rho	concentration parameter of the distribution. rho must be in the interval from 0 to 1.
sd	standard deviation of the (unwrapped) normal distribution.
from	if NULL is set to $mu - pi$. This is the value from which the pwrappednormal and qwrappednormal are evaluated. It should be a circular object.
K	number of terms to be used in approximating the density.
min.k	minimum number of terms used in approximating the density.
tol	passed to [uniroot](#).
control.circular	the attribute of the resulting object.
...	parameters passed to [integrate](#).

Value

dwrappednormal gives the density and rwrappednormal generates random deviates, pwrappednormal gives the distribution function, and qwrappednormal provides quantiles.

Author(s)

Claudio Agostinelli and Ulric Lund

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 2.2.7, World Scientific Press, Singapore.

Examples

```
data1 <- rwrappednormal(100, mu=circular(0), rho=0.7,
  control.circular=list(units="degrees"))
plot(data1)

ff <- function(x) dwrappednormal(x, mu=circular(pi), rho=0.7)
curve.circular(ff, join=TRUE, xlim=c(-1.5, 1),
  main="Density of a Wrapped Normal Distribution \n mu=pi, rho=0.7")

ff <- function(x) pwrappednormal(x, mu=circular(pi), rho=0.7)
curve.circular(ff, join=FALSE, xlim=c(-2, 2), ylim=c(-2, 2),
  to=(2*pi-3*.Machine$double.eps), modulo="asis", nosort=TRUE,
  main="Probability of a Wrapped Normal Distribution \n mu=pi,
  rho=0.7, from=0")

ff <- function(x) pwrappednormal(x, mu=circular(pi), rho=0.7, from=circular(pi))
curve.circular(ff, join=FALSE, xlim=c(-2, 2), ylim=c(-2, 2), from=-pi,
  to=(pi-3*.Machine$double.eps), modulo="asis", nosort=TRUE,
  main="Probability of a Wrapped Normal Distribution \n mu=pi,
  rho=0.7, from=pi")

plot(qwrappednormal, from=0, to=1)
plot(function(x) qwrappednormal(p=x, mu=circular(pi)), from=0, to=1)
```

[.circular   *Extract or Replace Parts of a Circular Object*

Description

Operators act on vectors and matrices to extract or replace subsets, methods for Circular Data.

Usage

```
## S3 method for class 'circular'
x[i, ...]
```

Arguments

x   object from which to extract elements.

i, ...   elements to extract or replace.

Author(s)

Claudio Agostinelli

Examples

```
x <- circular(matrix(rwrappednormal(n=100, mu=circular(0)), nrow=5))
dim(x)
x[1,]
x[,1]
x[,1, drop=FALSE]
```

Index