

# Package ‘circular’

April 17, 2009

**Title** Circular Statistics

**LazyLoad** yes

**LazyData** yes

**Version** 0.3-8

**Author** Ulric Lund <ulund@calpoly.edu>, Claudio Agostinelli <claudio@unive.it>

**Maintainer** Claudio Agostinelli <claudio@unive.it>

**Date** 2007-08-15

**Depends** boot

**Description** Circular Statistics, from “Topics in circular Statistics” (2001) S. Rao Jammalamadaka and A. SenGupta, World Scientific.

**License** GPL-2

**Repository** CRAN

**Date/Publication** 2007-08-25 14:33:25

## R topics documented:

A1 . . . . .	3
A1inv . . . . .	4
aov.circular . . . . .	5
as.data.frame.circular . . . . .	7
axialvonMises . . . . .	7
axis.circular . . . . .	8
c.circular . . . . .	9
Cardioid . . . . .	10
change.point . . . . .	11
Circular . . . . .	12
circularp . . . . .	13
conversion.circular . . . . .	14

Coope . . . . .	15
cor.circular . . . . .	16
curve.circular . . . . .	17
deg . . . . .	18
density.circular . . . . .	19
equal.kappa.test . . . . .	21
I.0 . . . . .	22
I.1 . . . . .	23
I.p . . . . .	23
kuiper.test . . . . .	24
lines.circular . . . . .	25
lines.density.circular . . . . .	26
lm.circular . . . . .	27
lsfit.circle . . . . .	30
mean.circular . . . . .	32
mixedvonmises . . . . .	33
mle.vonmises . . . . .	34
mle.vonmises.bootstrap.ci . . . . .	36
mle.wrappedcauchy . . . . .	37
mle.wrappednormal . . . . .	38
ncfrog . . . . .	40
plot.circular . . . . .	41
plot.density.circular . . . . .	42
plot.edf . . . . .	44
plot.lsfit.circle . . . . .	46
points.circular . . . . .	47
pp.plot . . . . .	48
rad . . . . .	49
range.circular . . . . .	50
rao.spacing.test . . . . .	51
rao.table . . . . .	52
rao.test . . . . .	53
rayleigh.test . . . . .	54
rho.circular . . . . .	55
rose.diag . . . . .	56
rstable . . . . .	58
rwrappedstable . . . . .	59
summary.circular . . . . .	60
ticks.circular . . . . .	61
triangular . . . . .	62
trigonometric.moment . . . . .	63
unique.circular . . . . .	64
var . . . . .	64
var.circular . . . . .	65
vonMises . . . . .	66
watson.test . . . . .	67
watson.two.test . . . . .	68
windrose . . . . .	70



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](#), [bessell](#).

**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)))
```

---

aov.circular                      *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 <code>circular</code> .
group	a vector identifying the groups or samples.
kappa	the common value of the concentration parameter. Used when <code>method</code> 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 <code>mu</code> and <code>mu.all</code> . See <a href="#">circular</a> 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:

<code>mu</code>	mean direction for each sample with class <code>circular</code> .
<code>mu.all</code>	mean direction of all samples combined with class <code>circular</code> .
<code>kappa</code>	concentration parameter for each sample.
<code>kappa.all</code>	concentration parameter for all samples combined.
<code>rho</code>	mean resultant length for each sample.
<code>rho.all</code>	mean resultant length for all samples combined.
<code>method</code>	the test statistic used.
<code>df</code>	degrees of freedom.
<code>statistic</code>	the value of the test statistic.
<code>p.value</code>	the p.value of the test statistic.
<code>call</code>	<code>match.call()</code> .
<code>SSE</code>	Sum of squares used in F-test.
<code>MSE</code>	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")
```

---

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

---

**Description**

This function is a method of `as.data.frame` for a circular object.

**Usage**

```
as.data.frame.circular(x, row.names = NULL, optional = FALSE, ...)
```

**Arguments**

x	object of class <code>circular</code> .
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.
...	

**Author(s)**

Claudio Agostinelli

---

```
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 <code>circular</code> .
mu	mean direction of the distribution. The object is coerced to class <code>circular</code>
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))))
```

---

c.circular

*A method for circular object, which combines its arguments*

---

**Usage**

```
c.circular(..., recursive = FALSE)
```

**Arguments**

...	vectors, the first of which of class <code>circular</code> .
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
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**

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

---

change.point	<i>Change Point Test</i>
--------------	--------------------------

---

**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.

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"), temp
as.circular(x, control.circular=list(), ...)
is.circular(x)
## S3 method for class 'circular':
print(x, info=TRUE, ...)
```

**Arguments**

<code>x</code>	a vector or a matrix. If a data.frame is supply then it is corced to a matrix.
<code>type</code>	the type of measures (Not Used Yet).
<code>units</code>	units of the measures.
<code>template</code>	how the data should be plotted (Until now only 'geographics' style is available). This set <code>modulo</code> , <code>zero</code> and <code>rotation</code> to some suitable values. For instance for 'geographics': <code>zero=pi/2</code> and <code>rotation='clock'</code> . It is also used to set default labels on the plots.
<code>modulo</code>	if we need to reduce the measures to modulo.
<code>zero</code>	the zero of the axes (in radians, counter).
<code>rotation</code>	the orientation of the axes.
<code>names</code>	names of the data.
<code>info</code>	if TRUE information on the properties of the data are printed.
<code>control.circular</code>	the attribute (coordinate system) used to coerced the resulting objects. See <a href="#">circular</a> .
<code>...</code>	For <code>as.circular</code> an alternative way of setting the coordinate system of the resulting objects. Passed parameters to <code>print.default</code> for <code>print.circular</code> .

**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 ca

```
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)
```

---

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,
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"), type = NULL, template = NU
```

**Arguments**

<code>x</code>	an object of class <code>circular</code> .
<code>units</code>	unit of the transformed data.
<code>type</code>	type of the transformed data. If <code>NULL</code> no action is performed.
<code>template</code>	template of the transformed data. If <code>NULL</code> no action is performed.
<code>modulo</code>	modulo of the transformed data. If <code>NULL</code> no action is performed.
<code>zero</code>	zero of the transformed data. If <code>NULL</code> no action is performed.
<code>rotation</code>	rotation of the transformed data. If <code>NULL</code> 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.circular` and `y.circular` 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.

---

<code>cor.circular</code>	<i>Correlation Coefficient for Angular Variables</i>
---------------------------	--

---

**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**

<code>x</code>	vector or matrix of circular data.
<code>y</code>	vector or matrix of circular data.
<code>test</code>	if <code>test == TRUE</code> , 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,  $(x_i - \bar{x})$  and  $(y_i - \bar{y})$  are replaced by  $\sin(x_i - \bar{x})$  and  $\sin(y_i - \bar{y})$ , where  $\bar{x}$  and  $\bar{y}$  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

```
curve.circular(expr, from = NULL, to = NULL, n = 101, add = FALSE, cex = 1, axes =
plot.function.circular(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, xlab, ylab, cex` graphical 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](#)

### 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**

<code>x</code>	the data from which the estimate is to be computed. The object is coerced to class <code>circular</code> .
<code>z</code>	the points where the density is estimated. If <code>NULL</code> equally spaced points are used according to the parameters <code>from</code> , <code>to</code> and <code>n</code> .
<code>bw</code>	the smoothing bandwidth to be used. When the kernel is <code>vonmises</code> the bandwidth is equal to the concentration parameter.
<code>adjust</code>	the bandwidth used is actually <code>adjust*bw</code> . This makes it easy to specify values like “half the default bandwidth”.
<code>type</code>	Not Yet Used.
<code>kernel</code>	a character string giving the smoothing kernel to be used. This must be one of “ <code>vonmises</code> ” or “ <code>wrappednormal</code> ”, that are kernels of type “ <code>K</code> ”.

<code>na.rm</code>	logical; if TRUE, missing values are removed from <code>x</code> . If FALSE any missing values cause an error.
<code>from, to</code>	the left and right-most points of the grid at which the density is to be estimated. The objects are coerced to class <code>circular</code> .
<code>n</code>	the number of equally spaced points at which the density is to be estimated.
<code>K</code>	number of terms to be used in approximating the density.
<code>min.k</code>	minimum number of terms used in approximating the density.
<code>control.circular</code>	the attribute of the resulting objects ( <code>x</code> component).
<code>digits</code>	integer indicating the precision to be used.
<code>...</code>	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.

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

---

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 <code>circular</code> .
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:

<code>kappa</code>	concentration parameter for each sample.
<code>kappa.all</code>	concentration parameter of all samples combined.
<code>rho</code>	mean resultant length for each sample.
<code>rho.all</code>	mean resultant length of all samples combined.
<code>df</code>	degrees of freedom for chi-squared distribution.
<code>statistic</code>	the value of the chi-squared test statistic.
<code>p.value</code>	the p.value of the test statistic.
<code>call</code>	<code>match.call()</code>

**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)
```

---

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](#).

---

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 <a href="#">circular</a> .
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**

```
lines.circular(x, y, join = FALSE, nosort = FALSE, plot.info = NULL, zero = NULL, r
```

**Arguments**

x	a vector of class <code>circular</code> .
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.
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 <a href="#">lines.default</a> .

**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)
```

**Arguments**

<code>x</code>	an object of class <code>density.circular</code> .
<code>type</code>	plotting parameter with useful default.
<code>zero.line</code>	logical; if TRUE, add a base line at $y = 0$ . Used when <code>plot.type</code> is "line".
<code>points.plot</code>	logical; if TRUE original data are added to the plot.
<code>points.col</code> , <code>points.pch</code> , <code>points.cex</code>	parameters used to draw the points.
<code>plot.type</code>	type of the plot.
<code>bins</code>	number of ticks to plot.
<code>shrink</code>	parameter that controls the size of the plotted circle. Default is 1. Larger values shrink the circle, while smaller values enlarge the circle.
<code>tcl</code>	length of the ticks.
<code>sep</code>	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 <code>plot.circular</code> that contains information on the zero, the rotation and <code>next.points</code> .
zero	the zero of the plot. Ignored if <code>plot.info</code> is provided.
rotation	the rotation of the plot. Ignored if <code>plot.info</code> is provided.
...	further parameters passed to <code>lines.default</code> .

**Value**

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

**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 =
## S3 method for class 'lm.circular.cl':
print(x, digits = max(3, getOption("digits") - 3), signif.stars= getOption("show.si
```

**Arguments**

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

**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:  $\mu + 2 \cdot \text{atan}(\beta \cdot x)$ , 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:

<code>call</code>	<code>match.call()</code> .
<code>rho</code>	square root of the average of the squares of the estimated conditional concentration parameters of <code>y</code> given <code>x</code> .
<code>fitted</code>	fitted values of the model of class <code>circular</code> .
<code>data</code>	matrix whose columns correspond to <code>x</code> and <code>y</code> .
<code>residuals</code>	circular residuals of the model of class <code>circular</code> .
<code>coefficients</code>	matrix whose entries are the estimated coefficients of the model. The first column corresponds to the coefficients of the model predicting the cosine of <code>y</code> , while the second column contains the estimates for the model predicting the sine of <code>y</code> . The rows of the matrix correspond to the coefficients according to increasing trigonometric order.
<code>p.values</code>	p-values testing whether the $(\text{order} + 1)$ trigonometric terms are significantly different from zero.
<code>A.k</code>	is mean of the cosines of the circular residuals.
<code>kappa</code>	assuming the circular residuals come from a von Mises distribution, <code>kappa</code> is the MLE of the concentration parameter.
<code>call</code>	<code>match.call()</code> .
<code>x</code>	the independent variables.
<code>y</code>	the dependent variable.
<code>mu</code>	the circular mean of the dependent variable of class <code>circular</code> .
<code>se.mu</code>	an estimated standard error of the circular mean with the same units of measure used for <code>mu</code> .
<code>kappa</code>	the concentration parameter for the dependent variable.
<code>se.kappa</code>	an estimated standard error of the concentration parameter.
<code>coefficients</code>	the estimated coefficients.
<code>cov.coef</code>	covariance matrix of the estimated coefficients.
<code>se.coef</code>	standard error of the estimated coefficients.
<code>log.lik</code>	log-likelihood.
<code>t.values</code>	values of the t statistics for the coefficients.
<code>p.values</code>	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.
set.seed(1234)
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)
```

---

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**

<code>x</code>	either a matrix with two columns or a vector.
<code>y</code>	if <code>x</code> is a vector then <code>y</code> must be a vector with the same length.
<code>init</code>	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 $c(\max(c(\text{abs}(x - \text{mean}(x)), \text{abs}(y - \text{mean}(y))))), \text{mean}(x), \text{mean}(y)$ .
<code>units</code>	the <code>units</code> used in defining the angles between observations and the center of the circle. See <a href="#">circular</a> .
<code>template</code>	the <code>template</code> used in defining the angles between observations and the center of the circle. See <a href="#">circular</a> .
<code>modulo</code>	the <code>modulo</code> used in defining the angles between observations and the center of the circle. See <a href="#">circular</a> .
<code>zero</code>	the <code>zero</code> used in defining the angles between observations and the center of the circle. See <a href="#">circular</a> .
<code>rotation</code>	the <code>rotation</code> used in defining the angles between observations and the center of the circle. See <a href="#">circular</a> .
<code>...</code>	further parameters passed to the <code>optim</code> function.
<code>digits</code>	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`.

<code>coefficients</code>	a vector of length 3 with the estimated radius and coordinate of the center of the fitted circle.
<code>x</code>	the x-coordinate.
<code>y</code>	the y-coordinate.
<code>x.centered</code>	the x-coordinate re-centered at the center of the circle.
<code>y.centered</code>	the y-coordinate re-centered at the center of the circle.
<code>angles</code>	angles of the observations with respect to the center coordinate of class <code>\circular</code> .
<code>radius</code>	the distance between the observations and the center coordinate
<code>convergence</code>	value from the function <code>optim</code> .
<code>optim</code>	the output from the function <code>optim</code> .
<code>call</code>	<code>match.call()</code> .

**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 ci
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	<i>Mean Direction</i>
---------------	-----------------------

---

## 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 <code>circular</code> .
na.rm	logical, indicating if <code>NA</code> '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.
data <- circular(runif(50, circular(0), pi))
mean(data)
```

---

mixedvonmises

*Mixture of von Mises Distributions*

---

**Description**

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

**Usage**

```
dmixedvonmises(x, mu1, mu2, kappal, kappa2, p)
rmixedvonmises(n, mu1, mu2, kappal, kappa2, p, control.circular = list())
```

**Arguments**

<code>x</code>	a vector. The object is coerced to class <code>circular</code> .
<code>n</code>	number of observations.
<code>mu1</code>	mean direction of one of the two von Mises distributions as a <code>circular</code> object.
<code>mu2</code>	mean direction of the other von Mises distribution as a <code>circular</code> object.
<code>kappal</code>	concentration parameter of one of the two von Mises distributions.
<code>kappa2</code>	concentration parameter of the other von Mises distribution.
<code>p</code>	mixing proportion.
<code>control.circular</code>	the attribute of the resulting object.

**Value**

`dmixedvonmises` gives the density and `rmixedvonmises` generates random deviates.

**Author(s)**

Claudio Agostinelli and Ulric Lund

**See Also**

[dvonmises](#) and [rvonmises](#)

**Examples**

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

---

<code>mle.vonmises</code>	<i>von Mises Maximum Likelihood Estimates</i>
---------------------------	---

---

**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**

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

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

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

**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:

<code>mu.ci</code>	limits of the confidence interval for $\mu$ as a <code>circular</code> object.
<code>kappa.ci</code>	limits of the confidence interval for $\kappa$ .
<code>mu</code>	estimate of $\mu$ as a <code>circular</code> object.
<code>kappa</code>	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 <code>circular</code> .
mu	if <code>NULL</code> the maximum likelihood estimate of the mean direction is calculated otherwise it is coerced to an object of class <code>circular</code> .
rho	if <code>NULL</code> 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 ( <code>mu</code> )
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().
mu	the estimate of the mean direction or the value supplied as an object of class <code>circular</code> .
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=li
## S3 method for class 'mle.wrappednormal':
print(x, digits = max(3, getOption("digits") - 3), ...)
```

**Arguments**

<code>x</code>	a vector. The object is coerced to class <code>circular</code> .
<code>mu</code>	if <code>NULL</code> the maximum likelihood estimate of the mean direction is calculated, otherwise the value is coerced to an object of class <code>circular</code> .
<code>rho</code>	if <code>NULL</code> the maximum likelihood estimate of the concentration parameter is calculated.
<code>sd</code>	standard deviation of the (unwrapped) normal. Used as an alternative parametrization.
<code>K</code>	number of terms to be used in approximating the density.
<code>tol</code>	precision of the estimation.
<code>min.sd</code>	minimum value should be reached by the search procedure for the standard deviation parameter.
<code>min.k</code>	minimum number of terms used in approximating the density.
<code>max.iter</code>	maximum number of iterations.
<code>verbose</code>	logical, if <code>TRUE</code> information on the convergence process are printed.
<code>control.circular</code>	the attribute of the resulting objects ( <code>mu</code> )
<code>digits</code>	integer indicating the precision to be used.
<code>...</code>	further arguments passed to or from other methods.

**Value**

Returns a list with the following components:

<code>call</code>	the <code>match.call()</code> .
<code>mu</code>	the estimate of the mean direction or the value supplied as an object of class <code>circular</code> .
<code>rho</code>	the estimate of the concentration parameter or the value supplied
<code>sd</code>	the estimate of the standard deviation or the value supplied.
<code>est.mu</code>	<code>TRUE</code> if the estimator is reported.
<code>est.rho</code>	<code>TRUE</code> if the estimator is reported.
<code>convergence</code>	<code>TRUE</code> if the convergence is achieved.

**Author(s)**

Claudio Agostinelli

**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
```

---

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.

---

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, sep = 0.025, shrink = 1, bins = 1, ticks = FALSE, tcl = 0, tcl.text = "none", col = "black", tol = 0.05, uin = 1, xlim = NULL, ylim = NULL, digits = 3, main = NULL, xlab = NULL, ylab = NULL, units = "degrees", template = "none", zero = 0, rotation = 0, ...)
```

### Arguments

x	a vector, matrix or data.frame. The object is coerced to class <code>circular</code> .
pch	point character to use. See help on <code>par</code> .
cex	point character size. See help on <code>par</code> .
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.
sep	constant used to specify the distance between stacked points, if <code>stack==TRUE</code> 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 <code>stack==TRUE</code> , 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, xlab, ylab	title, 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.
...	further parameters passed to <code>plot.default</code> .

**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.

**Value**

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

**Note**

some codes from [eqsplot](#) 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, 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=TRUE, bins=NULL, 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, ...)
```

**Arguments**

<code>x</code>	an object of class <code>density.circular</code> .
<code>main</code> , <code>xlab</code> , <code>ylab</code> , <code>type</code>	plotting parameters with useful defaults.
<code>zero.line</code>	logical; if TRUE, add a base line at $y = 0$ . Used when <code>plot.type</code> is "line".
<code>points.plot</code>	logical; if TRUE original data are added to the plot.
<code>points.col</code> , <code>points.pch</code> , <code>points.cex</code>	parameters used to draw the points.
<code>plot.type</code>	type of the plot: "line": linear plot, "circle": circular plot.
<code>axes</code>	logical; if TRUE axis are drawn.
<code>ticks</code>	logical; if TRUE ticks are drawn.
<code>bins</code>	number of ticks to plot.
<code>shrink</code>	parameter that controls the size of the plotted circle. Default is 1. Larger values shrink the circle, while smaller values enlarge the circle.
<code>tcl</code>	length of the ticks.
<code>tcl.text</code>	The position of the axis labels.
<code>sep</code>	constant used to specify the distance between stacked points. Default is 0.025; smaller values will create smaller spaces.
<code>tol</code>	proportion of white space at the margins of plot
<code>digits</code>	number of digits used to print axis values.
<code>cex</code>	point character size. See help on <code>par</code> .
<code>uin</code>	desired values for the units per inch parameter. If of length 1, the desired units per inch on the x axis.
<code>xlim</code> , <code>ylim</code>	the ranges to be encompassed by the x and y axes. Useful for centering the plot.
<code>join</code>	logical: should the first and the last point joined.
<code>nosort</code>	logical: should the data sort before plotting. Defaults is to sort.
<code>units</code>	units measure used in the plot. If NULL the value is taken from the attribute of object 'x' from the argument 'x', i.e. <code>x\$x</code> .
<code>template</code>	template used in the plot. If NULL the value is taken from the attribute of object 'x' from the argument 'x', i.e. <code>x\$x</code> .
<code>zero</code>	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. <code>x\$x</code> .

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.

... 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="geogr
y <- rvonmises(n=100, mu=circular(pi/2), kappa=2, control.circular=list(template="geographic
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
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)

res <- plot.function.circular(function(x) ff(x, data=data2), from=0, to=(2*pi-3*.Machine$double.eps)

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 = NU
```

### Arguments

<code>x</code>	an object of class <code>lsfit.circle</code> .
<code>add</code>	logical: if <code>TRUE</code> the plot is superimposed on the active device.
<code>main</code>	a main title for the plot.
<code>xlim</code>	the x limits (min,max) of the plot.
<code>ylim</code>	the y limits of the plot.
<code>xlab</code>	a label for the x axis.
<code>ylab</code>	a label for the x axis.
<code>uin</code>	desired values for the units per inch parameter. If of length 1, the desired units per inch on the x axis.
<code>tol</code>	proportion of white space at the margins of plot.
<code>plus.cex</code>	dimension of the cross in the center of the circle.
<code>...</code>	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

```
points.circular(x, pch = 16, cex = 1, stack = FALSE, 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 <code>circular</code> .
pch	point character to use. See help on <code>par</code> .
cex	point character size. See help on <code>par</code> .
stack	logical: if <code>TRUE</code> , points are stacked on the perimeter of the circle. Otherwise, all points are plotted on the perimeter of the circle. Default is <code>FALSE</code> .
sep	constant used to specify the distance between stacked points, if <code>stack==TRUE</code> 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 <code>stack==TRUE</code> , 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 <code>stack=FALSE</code> , the distance from the circle the next dataset is plotted. Ignored if <code>plot.info</code> is provided.
plot.info	an object from <code>plot.circular</code> that contains information on the zero, the rotation and <code>next.points</code> .
zero	the zero of the plot. Ignored if <code>plot.info</code> is provided.
rotation	the rotation of the plot. Ignored if <code>plot.info</code> is provided.
...	further parameters passed to <code>points.default</code> .

## 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.

**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 = "Em
```

**Arguments**

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

---

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**

```
range.circular(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`

`...`                  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.

**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**

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

**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 <code>circular</code> .
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**

<code>...</code>	a sequence of <code>circular</code> for the <code>rao.test</code> and further arguments passed to or from other methods for the <code>print.rao.test</code> function.
<code>alpha</code>	numeric value specifying the significance level of the test. Default is 0, in which case p-values for the test statistic is printed.
<code>x</code>	an object from the <code>rao.test</code> .
<code>digits</code>	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**

<code>x</code>	a vector. The object is coerced to class <code>circular</code> .
<code>mu</code>	Specified mean direction in alternative hypothesis as a <code>circular</code> object.
<code>digits</code>	integer indicating the precision to be used.
<code>...</code>	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	<i>Mean Resultant Length</i>
--------------	------------------------------

---

**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 <code>circular</code> .
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,
  ticks = TRUE, tcl = 0.025, tcl.text = 0.125, col = NULL, tol = 0.04, uin = NU
  xlim = c(-1, 1), ylim = c(-1, 1), prop = 1, digits = 2, plot.info = NULL,
  units = NULL, template = NULL, zero = NULL, rotation = NULL,
  main = "", xlab = "", ylab = "", add = FALSE, ...)
```

### Arguments

x	a vector, matrix or data.frame. The object is coerced to class <code>circular</code> .
pch	point character to use. See help on <code>par</code> .
cex	point character size. See help on <code>par</code> .
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.
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.
prop	numerical constant determining the radii of the sectors. By default, <code>prop = 1</code> .
digits	number of digits used to print axis values.
plot.info	an object from <code>plot.circular</code> that contains information on the zero, the rotation and <code>next.points</code> .
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 <code>plot.info</code> is provided.
zero	the zero of the plot. Ignored if <code>plot.info</code> or <code>template</code> are provided.
rotation	the rotation of the plot. Ignored if <code>plot.info</code> or <code>template</code> are provided.

```

main, xlab, ylab      title, x label and y label of the plot.
add                  add the rose diag to an existing plot.
...                  futher parameters passed to lines.default.

```

### 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.

### Value

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

### Note

some codes from `eqsplot` in MASS is used.

### Author(s)

Claudio Agostinelli and Ulric Lund

### 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 poi
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)

```

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  $index=\alpha$ ,  $scale=c$  and  $skewness=\beta$ . 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  $Y = X_1 + X_2 + \dots + X_N$  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.
- Loĝaeĵe, 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.

---

summary.circular    *Circular Summary Statistics*

---

**Description**

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

**Usage**

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

**Arguments**

object            an object of class `circular`.  
...                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 and David Anzel

**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](#), [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)))
```

---

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)
```

**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](#).

---

`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**

<code>x</code>	a vector of class <code>circular</code> .
<code>p</code>	order of trigonometric moment to be computed. Default is for a first order trigonometric moment.
<code>center</code>	logical, whether to compute centered moments or not. Default is to compute an uncentered moment.
<code>control.circular</code>	the attribute of the resulting object <code>mu</code> .

**Value**

Returns a list with variables `mu`, `rho`, `cos`, `sin`, `p`, `n`, `call`, respectively the `p`th 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"))
```

---

`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 **base** is replaced by a new method in order to report the variance of circular data appropriately. `var.default` is an alias of the original function `var`. The behavior would be the same for objects which are not from `class data.frame` and `circular` (in the last case the variance is defined 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 column.

**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**

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

---

var.circular	<i>A measure of variance for Circular Data</i>
--------------	--

---

**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, only.var = TRUE, ...)
```

**Arguments**

x	a vector. The object is coerced to class <a href="#">circular</a> .
na.rm	logical, indicating if NA's should be omitted.
only.var	logical, if FALSE the mean result length is reported as well.
...	further arguments passed to or from other methods.

**Value**

Returns one minus the mean resultant length divided by the sample size of data if `only.var` is TRUE, otherwise a vector with the following components: the number of observations, the mean resultant length, the mean resultant length divided by the sample size, and one minus the mean resultant length divided by the sample size.

**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](#), [rho.circular](#) and [summary.circular](#).

**Examples**

```
# Compute the mean resultant length of a random sample of observations.
x <- rvonmises(n=100, mu=circular(0), kappa=1)
var(x)
```

---

vonMises

*von Mises Density Function*

---

**Description**

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

**Usage**

```
dvonmises(x, mu, kappa)
pvonmises(q, mu, kappa, tol = 1e-020)
rvonmises(n, mu, kappa, control.circular=list())
```

**Arguments**

<code>x, q</code>	a vector. The object is coerced to class <a href="#">circular</a> .
<code>n</code>	number of observations.
<code>mu</code>	mean direction of the distribution. The object is coerced to class <a href="#">circular</a> .
<code>kappa</code>	non-negative numeric value for the concentration parameter of the distribution.
<code>control.circular</code>	the attribute of the resulting object.
<code>tol</code>	the precision in evaluating the distribution function.

**Value**

`dvonmises` gives the density, `pvonmises` gives the distribution function and `rvonmises` generates random deviates.

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 m

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)
```

---

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**

<code>x</code>	a vector. The object is coerced to class <code>circular</code> .
<code>alpha</code>	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.
<code>dist</code>	distribution to test for. The default is the uniform distribution. To test for the von Mises distribution, set <code>dist</code> to "vonmises".
<code>digits</code>	integer indicating the precision to be used.
<code>...</code>	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 <code>circular</code> .
y	a vector. The object is coerced to class <code>circular</code> .
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)
```

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,
         ticks=TRUE, tcl=0.025, tcl.text=-0.15, cex = 1, digits=2, num.ticks=1,
         ylim=c(-1.2, 1.2), uin, tol=0.04, right=FALSE, shrink=NULL, label.fre
```

**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 <code>circular</code> .
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 <code>circular</code> 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 <code>axis.circular</code> is used.
ticks	if TRUE ticks are added to the plot. The function <code>ticks.circular</code> is used.
tcl	length of the ticks.
tcl.text	The position of the axis labels.
cex	point character size. See help on <code>par</code> .

<code>digits</code>	number of digits used to print axis values and other numbers.
<code>num.ticks</code>	number of tick marks draw.
<code>tol</code>	proportion of white space at the margins of plot
<code>uin</code>	desired values for the units per inch parameter. If of length 1, the desired units per inch on the x axis.
<code>xlim, ylim</code>	the ranges to be encompassed by the x and y axes. Useful for centering the plot.
<code>right</code>	logical; if TRUE, the pedals are right-closed (left open) intervals.
<code>shrink</code>	maximum length of the pedals, it can be used to plot several graphics with the same scale.
<code>label.freq</code>	logical; if TRUE, the relative frequencies are used in the magnitude instead of intensities, when the breaks are equally spaced.
<code>calm</code>	"0" or "NA", see details below.
<code>...</code>	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

<code>x</code>	directions
<code>y</code>	magnitudes
<code>table</code>	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.
<code>number.obs</code>	Total number of observations.
<code>number.calm</code>	The number of calm observations omitted from the wind rose plot.
<code>breaks</code>	extremes of the pedals.
<code>mids</code>	midpoints of pedals.
<code>call</code>	the <code>match.call</code> 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 <code>circular</code> .
n	number of observations.
mu	mean direction of the distribution as a <code>circular</code> 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")
```

---

 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**

<code>x</code> , <code>q</code>	vector of quantiles. The object is coerced to class <code>circular</code> .
<code>p</code>	vector of probabilities.
<code>n</code>	number of observations.
<code>mu</code>	mean direction of the distribution as a <code>circular</code> object.
<code>rho</code>	concentration parameter of the distribution. <code>rho</code> must be in the interval from 0 to 1.

<code>sd</code>	standard deviation of the (unwrapped) normal distribution.
<code>from</code>	if <code>NULL</code> is set to $\mu - \pi$ . This is the value from which the <code>pwrappednormal</code> and <code>qwrappednormal</code> are evaluated. It should be a <code>circular</code> object.
<code>K</code>	number of terms to be used in approximating the density.
<code>min.k</code>	minimum number of terms used in approximating the density.
<code>tol</code>	passed to <code>uniroot</code> .
<code>control.circular</code>	the attribute of the resulting object.
<code>...</code>	parameters passed to <code>integrate</code> .

### Value

`dwrappednormal` gives the density and `rwrappednormal` 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 <- 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")

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))

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))

curve(qwrappednormal, 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**

<code>x</code>	object from which to extract elements.
<code>i, ...</code>	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

- \*Topic **array**
  - [.circular, 74
- \*Topic **classes**
  - as.data.frame.circular, 5
- \*Topic **datasets**
  - Coope, 14
  - ncfrog, 39
  - rao.table, 51
- \*Topic **distribution**
  - axialvonMises, 6
  - Cardioid, 8
  - density.circular, 18
  - mixedvonmises, 32
  - rstable, 57
  - rwrappedstable, 58
  - triangular, 61
  - vonMises, 65
  - wrappedCauchy, 71
  - wrappedNormal, 72
- \*Topic **dplot**
  - lines.density.circular, 25
  - plot.density.circular, 41
- \*Topic **hplot**
  - axis.circular, 6
  - curve.circular, 15
  - lines.circular, 24
  - plot.circular, 40
  - plot.edf, 43
  - plot.lsfite.circle, 45
  - points.circular, 46
  - pp.plot, 47
  - rose.diag, 55
  - ticks.circular, 60
  - windrose, 69
- \*Topic **htest**
  - change.point, 9
  - equal.kappa.test, 20
  - kuiper.test, 23
  - mle.vonmises, 33
  - mle.vonmises.bootstrap.ci, 35
  - mle.wrappedcauchy, 36
  - mle.wrappednormal, 37
  - range.circular, 49
  - rao.spacing.test, 50
  - rao.test, 52
  - rayleigh.test, 53
  - watson.test, 66
  - watson.two.test, 67
- \*Topic **logic**
  - unique.circular, 63
- \*Topic **manip**
  - c.circular, 8
  - unique.circular, 63
- \*Topic **math**
  - A1, 1
  - A1inv, 2
  - deg, 17
  - I.0, 21
  - I.1, 22
  - I.p, 22
  - rad, 48
  - trigonometric.moment, 62
- \*Topic **methods**
  - as.data.frame.circular, 5
- \*Topic **misc**
  - Circular, 10
  - circularp, 12
  - conversion.circular, 13
- \*Topic **models**
  - aov.circular, 3
  - lm.circular, 26
  - lsfit.circle, 29
- \*Topic **regression**
  - lm.circular, 26
- \*Topic **smooth**
  - density.circular, 18
- \*Topic **univar**
  - cor.circular, 14

- mean.circular, 31
- range.circular, 49
- rho.circular, 54
- summary.circular, 59
- var, 63
- var.circular, 64
- .Machine, 31, 32
- [.circular, 74
- A1, 1, 3
- Alinv, 2, 2
- aov.circular, 3
- as.circular (*Circular*), 10
- as.data.frame, 5
- as.data.frame.circular, 5
- axialvonMises, 6
- axis.circular, 6, 41, 60, 69
- besselI, 2, 3, 21, 22
- c, 8
- c.circular, 8
- Cardioid, 8
- change.point, 9
- Circular, 10
- circular, 4–6, 9, 11–13, 23, 30–33, 36, 38, 40, 46, 47, 49, 50, 52–55, 59, 61, 63–66, 68, 69, 71, 72
- circular (*Circular*), 10
- circularp, 12
- circularp<-, 13
- circularp<-(*circularp*), 12
- class, 63
- conversion.circular, 11, 13, 17, 48, 49
- Coope, 14
- cor.circular, 14
- curve.circular, 15, 41, 43, 44
- data.frame, 63
- daxialvonmises (*axialvonMises*), 6
- dcardioid (*Cardioid*), 8
- deg, 13, 17, 49
- density.circular, 18, 25, 26, 42, 43
- dmixedvonmises (*mixedvonmises*), 32
- dtriangular (*triangular*), 61
- dvonmises, 33
- dvonmises (*vonMises*), 65
- dwrappedcauchy (*wrappedCauchy*), 71
- dwrappednormal (*wrappedNormal*), 72
- eqsplot, 41, 56, 70
- equal.kappa.test, 20
- I.0, 21
- I.1, 22
- I.p, 22
- integrate, 73
- is.circular (*Circular*), 10
- kuiper.test, 23, 49, 51, 53, 67
- lines.circular, 16, 24, 41, 43, 47
- lines.default, 24, 26, 56
- lines.density.circular, 19, 25, 43
- lines.edf (*plot.edf*), 43
- lm.circular, 26
- lsfit.circle, 29, 45
- match.call, 70
- mean.circular, 31, 34, 37, 39, 54, 60, 62, 65
- mixedvonmises, 32
- mle.vonmises, 3, 32, 33, 35, 36, 48, 54, 60, 62
- mle.vonmises.bootstrap.ci, 34, 35
- mle.wrappedcauchy, 36
- mle.wrappednormal, 37
- NA, 31, 49, 54, 64
- ncfrog, 39
- options, 27
- par, 40, 42, 44, 46, 55, 69
- plot.circular, 7, 25, 26, 40, 43, 46, 47, 55, 56, 60
- plot.default, 40, 43, 47
- plot.density.circular, 19, 26, 41
- plot.ecdf, 44
- plot.edf, 43
- plot.function.circular (*curve.circular*), 15
- plot.lsfit.circle, 45
- points.circular, 41, 46
- points.default, 46
- pp.plot, 47
- print.aov.circular (*aov.circular*), 3
- print.circular (*Circular*), 10
- print.default, 35

print.density.circular  
     (*density.circular*), 18  
 print.equal.kappa.test  
     (*equal.kappa.test*), 20  
 print.kuiper.test (*kuiper.test*),  
     23  
 print.lm.circular.cl  
     (*lm.circular*), 26  
 print.lsfite.circle  
     (*lsfite.circle*), 29  
 print.mle.vonmises  
     (*mle.vonmises*), 33  
 print.mle.vonmises.bootstrap.ci  
     (*mle.vonmises.bootstrap.ci*),  
     35  
 print.mle.wrappedcauchy  
     (*mle.wrappedcauchy*), 36  
 print.mle.wrappednormal  
     (*mle.wrappednormal*), 37  
 print.rao.spacing.test  
     (*rao.spacing.test*), 50  
 print.rao.test (*rao.test*), 52  
 print.rayleigh.test  
     (*rayleigh.test*), 53  
 print.watson.test (*watson.test*),  
     66  
 print.watson.two.test  
     (*watson.two.test*), 67  
 pvonmises, 47  
 pvonmises (*vonMises*), 65  
 pwrappednormal (*wrappedNormal*), 72  
  
 qwrappednormal (*wrappedNormal*), 72  
  
 rad, 13, 17, 48  
 range.circular, 24, 49, 51, 53, 67  
 rao.spacing.test, 24, 49, 50, 51, 53, 67  
 rao.table, 51  
 rao.test, 52  
 rayleigh.test, 24, 49, 51, 53, 67  
 rcardioid (*Cardioid*), 8  
 rcauchy, 58  
 rho.circular, 31, 32, 54, 60, 62, 64, 65  
 rmixedvonmises (*mixedvonmises*), 32  
 rnorm, 58  
 rose.diag, 41, 55  
 rstable, 57  
 rtriangular (*triangular*), 61  
 rvonmises, 33  
  
 rvonmises (*vonMises*), 65  
 rwrappedcauchy (*wrappedCauchy*), 71  
 rwrappednormal (*wrappedNormal*), 72  
 rwrappedstable, 58  
  
 summary.circular, 32, 54, 59, 62, 64, 65  
 summary.matrix, 59  
  
 ticks.circular, 7, 41, 60, 69  
 triangular, 61  
 trigonometric.moment, 62  
  
 unique, 63  
 unique.circular, 63  
 unique.default, 63  
 uniroot, 73  
  
 var, 63, 63, 64  
 var.circular, 32, 54, 60, 62, 63, 64, 64  
 vonMises, 65  
  
 watson.test, 24, 49, 51, 53, 66  
 watson.two.test, 67  
 windrose, 41, 69  
 wrappedCauchy, 71  
 wrappedNormal, 72  
  
 x.coope (*Coope*), 14  
 y.coope (*Coope*), 14