Package 'cylcop'

August 6, 2021

Title Circular-Linear Copulas with Angular Symmetry for Movement Data

Version 0.1.0

Date 2021-08-05

Maintainer Florian Hodel <florian.hodel@yahoo.com>

Description Classes (S4) of circular-linear, symmetric copulas with corresponding methods, extending the 'copula' package. These copulas are especially useful for modelling correlation in discrete-time movement data. Methods for density, (conditional) distribution, random number generation, bivariate dependence measures and fitting parameters using maximum likelihood and other approaches. The package also contains methods for visualizing movement data and copulas.

License GPL (>= 2)

Encoding UTF-8

Depends R (>= 3.5),

Imports circular, stats, extraDistr, purrr, dplyr (>= 0.7.0), copula, stringr, rlang, methods, mvtnorm, GoFKernel, MASS, data.table, infotheo, ggplot2, utils, rgl, viridis, plotly, cowplot, movMF, Rdpack

RdMacros Rdpack

RoxygenNote 7.1.1

Collate 'cyl_cop_class.R' 'Ccond.R' 'aaaglobal.R' 'correlation.R' 'cyl_cubsec.R' 'cyl_quadsec.R' 'cyl_rect_combine.R' 'cyl_rot_combine.R' 'cyl_vonmises.R' 'cylcop-package.R' 'density.R' 'fit_cop_corr.R' 'fit_cop_mle.R' 'fit_margin.R' 'mixedvonmises.R' 'opt_auto.R' 'plotting_functions.R' 'simulate_trajectory.R' 'utils.R' 'wrappedcauchy.R' 'zzz.R'

NeedsCompilation no

Author Florian Hodel [aut, cre] (<https://orcid.org/0000-0002-0099-1006>)

Repository CRAN

Date/Publication 2021-08-06 08:30:06 UTC

R topics documented:

angstep2xy — *Calculate the Next Position in a Trajectory from a Turn Angle and a Step Length*

Description

The xy-coordinates of a position in 2-D space is calculated from the angle between that position and the 2 previous ones in the trajectory and the distance between that position and the previous one.

Usage

```
angstep2xy(angle, steplength, prevp1, prevp2)
```

Arguments

angle	numeric value of the turn angle or a circular object, either in $[0, 2\pi)$ or in $[-\pi, \pi)$
steplength	numeric value giving the distance between the position and the previous one.
prevp1	numeric vector holding the x and y coordinates of the previous position.
prevp2	numeric vector holding the x and y coordinates of the position before the previous one.

Value

The function returns a numeric vector holding the x and y coordinates of the position

Examples

```
angstep2xy(1.5*pi, 2, prevp1 = c(1, 4), prevp2 = c(2, 7.5))
angstep2xy(-0.5*pi, 2, c(1, 4), c(2, 7.5))
```

bearing — *Compass Bearing of a Line Between 2 Points*

Description

The angle between a line between 2 points in Euclidean 2-D space and the line from (0,0) to (0,1) is calculated. In other words, the compass bearing of a line between 2 points where north is 0. Angles increase in clockwise direction.

Usage

```
bearing(point1, point2, fullcirc = TRUE)
```

Arguments

point1	numeric vector holding the x and y coordinates of the first point.
point2	numeric vector holding the x and y coordinates of the second point.
fullcirc	logical value indicating whether the output should be an angle on $[0, 2\pi)$ or $[-\pi, \pi)$.

Value

If fullcirc = FALSE, the function returns a numeric value (angle) from the interval $[-\pi, \pi)$.
If fullcirc = TRUE, the function returns a numeric value numeric from the interval $[0, 2\pi)$.

Examples

```
bearing(c(3,5), c(1,4))
bearing(c(3,5), c(1,4), fullcirc = FALSE)
```

---

ccylcop *Conditional Distributions of Circular-Linear Copulas*

Description

Calculates the conditional distributions and their inverses of circular-linear copulas and 2-dimensional linear-linear copulas.

Usage

```
ccylcop(u, copula, cond_on = 2, inverse = FALSE, ...)

## S4 method for signature 'Copula'
ccylcop(u, copula, cond_on, inverse)

## S4 method for signature 'cyl_cubsec'
ccylcop(u, copula, cond_on = 2, inverse = FALSE)

## S4 method for signature 'cyl_quadsec'
ccylcop(u, copula, cond_on = 2, inverse = FALSE)

## S4 method for signature 'cyl_rect_combine'
ccylcop(u, copula, cond_on = 2, inverse = FALSE)

## S4 method for signature 'cyl_rot_combine'
ccylcop(u, copula, cond_on = 2, inverse = FALSE)

## S4 method for signature 'cyl_vonmises'
ccylcop(u, copula, cond_on = 2, inverse = FALSE)
```

Arguments

u	matrix (or vector) of numeric values in $[0,1]^2$, containing as first column the circular (periodic) and as second the linear dimension.
copula	R object of class 'cyl_copula'. or 'Copula' (package '**copula**', only 2-dimensional).
cond_on	column number of u on which the copula is conditioned. E.g if cond_on = 2, the function calculates for each element in the first column of u the copula conditional on the corresponding element in the second column.
inverse	logical indicating whether the inverse of the conditional copula is calculated.
...	additional arguments.

Details

This is a generic that calls the function copula::cCopula() for 2-dimensional 'Copula' objects from the '**copula**' package for which copula::cCopula() is available. If copula::cCopula() is not available, the conditional copula is calculated numerically. For 'cyl_copula' objects, the conditional copula is calculated analytically or numerically (depending on the copula and the values of u). Note that the input arguments and the output of cylcop::ccylcop() differ from those of copula::cCopula().

Value

A vector containing the values of the distribution of the copula at [u,-cond_on] conditional on the values of [u,cond_on].

References

Nelsen RB (2006). *An Introduction to Copulas*, volume 139 of *Lecture Notes in Statistics*. Springer New York, New York, NY. ISBN 978-0-387-98623-4, doi: 10.1007/9781475730760, https://doi.org/10.1007/978-1-4757-3076-0.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

See Also

copula::cCopula()

Examples

```
cop <- cyl_quadsec(0.1)
#calculate C_u(v) with u = 0.1 and v = 0.5
cylcop::ccylcop(u = c(0.1, 0.5), copula = cop, cond_on = 1, inverse = FALSE)
#calculate C^-1_v(u) with u = 0.1 and v = 0.5 and with u = 0.4 and v = 0.2
cylcop::ccylcop(u = rbind(c(0.1, 0.5), c(0.4, 0.2)), copula = cop, cond_on = 2, inverse = TRUE)
```

circ_plot *Circular Scatterplot of Turn Angles and Step Lengths*

Description

This function produces a circular scatterplot with the step lengths plotted as distance from the center of a circle and the turn angles as angles (polar coordinates).

Usage

```
circ_plot(traj)
```

Arguments

traj	data.frame containing the trajectory produced by e.g. make_traj(). It must contain the columns traj$angle and traj$steplength.

Value

A 'ggplot' object.

References

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

See Also

cop_scat_plot(), traj_plot(), cop_plot(), scat_plot().

Examples

```
set.seed(123)

traj <- make_traj(100,
  copula = cyl_quadsec(0.1),
  marginal_circ = "vonmises",
  parameter_circ = list(0, 1),
  marginal_lin = "weibull", list(shape=3)
)
plot1 <- circ_plot(traj)
```

`cop_plot` *Surface Plot or Heat Map of the Distribution or the Density of a Copula*

Description

This function plots the distribution or the density of a copula. It can produce a surface plot using either functions from the '**rgl**' or from the '**plotly**' package, or it can produce a heat map using functions from '**ggplot2**'.

Usage

```
cop_plot(
  copula,
  type = c("pdf", "cdf"),
  plot_type = "rgl",
  resolution = 50,
  n_gridlines = 11
)
```

Arguments

`copula`	'`cyl_copula`' or a '`Copula`' object from the package '**copula**'.
`type`	character string describing what is plotted, either `"pdf"` or `"cdf"`.
`plot_type`	character string describing what type of plot is produced. Available plot types are: `"rgl"`: surface plot, `"plotly"`: interactive surface plot, or `"ggplot"`: heatmap
`resolution`	numeric value. The density or distribution will be calculated at `resolution^2` points.
`n_gridlines`	numeric value giving the number of grid lines drawn in u and v direction.

Value

Depending on `plot_type`, a '`ggplot`' object is returned, or a '**plotly**' visualization or '**rgl**' plot is produced.

References

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

See Also

`cop_scat_plot()`, `traj_plot()`, `circ_plot()`, `scat_plot()`.

Examples

```
cop_plot(copula::frankCopula(2), type="pdf", plot_type="ggplot")
cop_plot(copula::frankCopula(2), type="cdf", plot_type="ggplot")
cop_plot(copula::frankCopula(2), type="pdf", plot_type="ggplot", resolution = 5)

#opens a new window
cop_plot(cyl_quadsec(0.1), type="pdf", plot_type="rgl")
cop_plot(cyl_quadsec(0.1), type="pdf", plot_type="rgl", n_gridlines = 60)

cop_plot(cyl_quadsec(0.1), type="pdf", plot_type="plotly", n_gridlines = 20)
```

---

cop_scat_plot *Scatterplot of Copula Values*

Description

This function produces a scatterplot ('ggplot' object) of a sample from a copula. Either a sample is provided as input, or a sample of 10000 points is drawn from a copula to quickly visualize it.

Usage

```
cop_scat_plot(input)
```

Arguments

input	Either a data.frame containing the trajectory produced by e.g. make_traj(), which must contain columns traj$cop_u and traj$cop_v, or a 'cyl_copula' object or a 'Copula' object of the package '**copula**'.

Value

A 'ggplot' object, the scatterplot.

References

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

See Also

traj_plot(), circ_plot(), cop_plot(), scat_plot().

Examples

```
set.seed(123)

traj <- make_traj(100,
  copula = cyl_quadsec(0.1),
  marginal_circ = "vonmises",
  parameter_circ = list(0, 1),
  marginal_lin = "weibull",
  parameter_lin = list(shape=3)
)
cop_scat_plot(traj)
cop_scat_plot(cyl_quadsec(0.1))
```

---

cor_cyl *Estimate a Rank-Based Circular-Linear Correlation Coefficient*

Description

The code is based on Mardia (1976), Solow et al. (1988) and Tu (2015). The function returns a numeric value between 0 and 1, not -1 and 1, positive and negative correlation cannot be discerned. Note also that the correlation coefficient is independent of the marginal distributions.

Usage

```
cor_cyl(theta, x)
```

Arguments

theta	numeric vector of angles (measurements of a circular variable).
x	numeric vector of step lengths (measurements of a linear variable).

Value

A numeric value between 0 and 1, the circular-linear correlation coefficient.

References

Mardia KV (1976). "Linear-Circular Correlation Coefficients and Rhythmometry." *Biometrika*, **63**(2), 403–405. ISSN 00063444, doi: 10.2307/2335637, https://doi.org/10.2307/2335637.

Solow AR, Bullister JL, Nevison C (1988). "An application of circular-linear correlation analysis to the relationship between Freon concentration and wind direction in Woods Hole, Massachusetts." *Environmental Monitoring and Assessment*, **10**(3), 219–228. ISSN 1573-2959, doi: 10.1007/BF00395081, https://doi.org/10.1007/BF00395081.

Tu R (2015). "A Study of the Parametric and Nonparametric Linear-Circular Correlation Coefficient." *California Polytechnic State University, San Luis Obispo*, 1–24. https://digitalcommons.calpoly.edu/statsp/51/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

See Also

mi_cyl(), optCor().

Examples

```
set.seed(123)

cop <- cyl_quadsec(0.1)

#draw samples and calculate the correlation coefficient
sample <- rcylcop(100, cop)
cor_cyl(theta = sample[,1], x = sample[,2])

#the correlation coefficient is independent of the marginal distribution.
sample <- make_traj(100,
  cop,
  marginal_circ = "vonmises",
  parameter_circ = list(0, 1),
  marginal_lin = "weibull",
  parameter_lin = list(shape = 2)
)
cor_cyl(theta = sample$angle, x = sample$steplength)
cor_cyl(theta = sample$cop_u, x = sample$cop_v)

# Estimate correlation of samples drawn from circular-linear copulas with
# perfect correlation
cop <- cyl_rect_combine(copula::normalCopula(1))
sample <- rcylcop(100, cop)
cor_cyl(theta = sample[,1], x = sample[,2])
```

Cylcop *Distribution, Density, and Random Number Generation for Circular-Linear Copulas'*

Description

Calculate the distribution (pcylcop()), the density (dcylcop()), and generate random samples (rcylcop()) of a 'cyl_copula' object or a 'Copula' object (package '**copula**', only 2-dimensional). For 'Copula' objects pcylcop() and rcylcop() just call the functions of the '**copula**' package pCopula() and rCopula(), respectively. The density is, however, calculated differently in dcylcop() and dCopula(). The difference is that copula::dCopula() will return a density of 0 for points on the boundary of the unit square, whereas dcylcop() will return the correct density on the boundaries for both 'cyl_copula' and 'Copula' objects.

Usage

```
pcylcop(u, copula, ...)

rcylcop(n, copula, ...)

dcylcop(u, copula, log = FALSE, ...)

## S4 method for signature 'matrix,Copula'
dcylcop(u, copula)

## S4 method for signature 'numeric,Copula'
rcylcop(n, copula)

## S4 method for signature 'matrix,Copula'
pcylcop(u, copula)

## S4 method for signature 'numeric,cyl_cubsec'
rcylcop(n, copula)

## S4 method for signature 'matrix,cyl_cubsec'
dcylcop(u, copula)

## S4 method for signature 'matrix,cyl_cubsec'
pcylcop(u, copula)

## S4 method for signature 'numeric,cyl_quadsec'
rcylcop(n, copula)

## S4 method for signature 'matrix,cyl_quadsec'
dcylcop(u, copula)

## S4 method for signature 'matrix,cyl_quadsec'
pcylcop(u, copula)

## S4 method for signature 'numeric,cyl_rect_combine'
rcylcop(n, copula)

## S4 method for signature 'matrix,cyl_rect_combine'
dcylcop(u, copula)

## S4 method for signature 'matrix,cyl_rect_combine'
pcylcop(u, copula)

## S4 method for signature 'numeric,cyl_rot_combine'
rcylcop(n, copula)

## S4 method for signature 'matrix,cyl_rot_combine'
dcylcop(u, copula)
```

```
## S4 method for signature 'matrix,cyl_rot_combine'
pcylcop(u, copula)

## S4 method for signature 'numeric,cyl_vonmises'
rcylcop(n, copula)

## S4 method for signature 'matrix,cyl_vonmises'
dcylcop(u, copula)

## S4 method for signature 'matrix,cyl_vonmises'
pcylcop(u, copula)
```

Arguments

u	matrix (or vector) of numeric values in $[0, 1]^2$, containing as first column the circular (periodic) and as second the linear dimension
copula	R object of class 'cyl_copula'. or 'Copula' (package '**copula**', only 2-dimensional).
...	additional arguments
n	number of random samples to be generated with rcylcop().
log	logical indicating if the logarithm of the density should be returned.

Value

The functions pcylcop() and dcylcop() give a vector of length nrow(u) containing the distribution and the density, respectively, at the corresponding values of u. Te function rcylcop() generates a matrix with 2 columns and n rows containing the random samples.

References

Nelsen RB (2006). *An Introduction to Copulas*, volume 139 of *Lecture Notes in Statistics*. Springer New York, New York, NY. ISBN 978-0-387-98623-4, doi: 10.1007/9781475730760, https://doi.org/10.1007/978-1-4757-3076-0.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

See Also

copula::dCopula(), copula::pCopula(), copula::rCopula().

Examples

```
set.seed(123)

cop <- cyl_quadsec(0.1)
rcylcop(5, cop)
pcylcop(c(0.3, 0.1), cop)
pcylcop(rbind(c(0.3, 0.1), c(0.2, 1)), cop)
```

```
cop <- cyl_rot_combine(copula::frankCopula(2), shift = TRUE)
dcylcop(u = rbind(c(0.1, 0.4), c(1.0, 0.2)), copula = cop)
dcylcop(c(0.1, 0.3), cyl_quadsec(0.1), log = TRUE)

cop <- copula::normalCopula(0.3)
copula::dCopula(c(.Machine$double.eps,0.2),cop)
copula::dCopula(c(0,0.2),cop)
dcylcop(c(.Machine$double.eps,0.2),cop)
dcylcop(c(0,0.2),cop)
```

`cylcop_get_option` *Get Package Options*

Description

Currently the only option (`"silent"`) is to toggle verbosity on or off.

Usage

```
cylcop_get_option(option = NULL)
```

Arguments

`option`	character string, the name of the option.

Value

The numeric value of option. If no argument is provided, a list of all options is printed.

See Also

`cylcop_set_option()`

Examples

```
cylcop_get_option("silent")
cylcop_get_option()
```

cylcop_set_option *Set Package Options*

Description

Currently the only option is to toggle verbosity on or off.

Usage

```
cylcop_set_option(silent = FALSE)
```

Arguments

silent	logical, suppress all sounds and messages.

Value

No output, only side effects.

See Also

cylcop_get_option()

Examples

```
cylcop_set_option(silent = FALSE)
```

cyl_copula-class *An S4 Class of Bivariate Copulas on the Cylinder*

Description

The class 'cyl_copula' follows somewhat the structure of the class 'Copula' of the package '**copula**'. It contains circular-linear copulas.

Slots

name character string holding the name of the copula.

parameters numeric vector holding the parameter values.

param.names character vector holding the parameter names.

param.lowbnd numeric vector holding the lower bounds of the parameters.

param.upbnd numeric vector holding the upper bounds of the parameters.

Extended by

'cyl_copula' is extended by the following classes:

- 'cyl_vonmises': von Mises copulas.
- 'cyl_quadsec': Copulas with quadratic sections.
- 'cyl_cubsec': Copulas with cubic sections.
- 'cyl_rot_combine': Linear combinations of copulas and their 180 degree rotations.
- 'cyl_rect_combine': Rectangular patchwork copulas.

Objects from the Class

Objects are created by the functions cyl_vonmises(), cyl_quadsec(), cyl_cubsec(), cyl_rot_combine(), and cyl_rect_combine().

References

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

Examples

```
cop <- cyl_quadsec(0.1)
is(cop)
```

---

cyl_cubsec *Construction of 'cyl_cubsec' Objects*

Description

Constructs a circular-linear copula with cubic sections of class 'cyl_cubsec'.

Usage

```
cyl_cubsec(a = 1/(2 * pi), b = 1/(2 * pi))
```

Arguments

a	numeric value of the first parameter of the copula. It must be in $[-1/(2\pi)), 1/(2\pi))]$.
b	numeric value of the second parameter of the copula. It must be in $[-1/(2\pi)), 1/(2\pi))]$.

Value

An R object of class 'cyl_cubsec'.

References

Nelsen RB, Quesada-Molina JJ, RodrÃguez-Lallena JA (1997). "Bivariate copulas with cubic sections." *Journal of Nonparametric Statistics*, **7**(3), 205–220. ISSN 10485252, doi: 10.1080/10485259708832700, https://doi.org/10.1080/10485259708832700.

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

Examples

```
cop <- cyl_cubsec(a = 0.1, b = -0.1)
cop_plot(copula = cop, type = "pdf", plot_type = "ggplot")
```

`cyl_cubsec-class` *An S4 Class of Bivariate Copulas with Cubic Sections*

Description

This class contains bivariate circular-linear copulas with cubic sections in the linear dimension. They are periodic in the circular dimension, u, and symmetric with respect to u=0.5. I.e. the can capture correlation in data where there is symmetry between positive and negative angles. These copulas are described by two parameters, `a` and `b`.

Slots

`name` character string holding the name of the copula.

`parameters` numeric vector holding the parameter values.

`param.names` character vector holding the parameter names.

`param.lowbnd` numeric vector holding the lower bounds of the parameters.

`param.upbnd` numeric vector holding the upper bounds of the parameters.

Objects from the Class

Objects are created by `cyl_cubsec()`.

Extends

Class '`cyl_cubsec`' extends class '`cyl_copula`'.

References

Nelsen RB, Quesada-Molina JJ, RodrÃguez-Lallena JA (1997). "Bivariate copulas with cubic sections." *Journal of Nonparametric Statistics*, **7**(3), 205–220. ISSN 10485252, doi: 10.1080/10485259708832700, https://doi.org/10.1080/10485259708832700.

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

`cyl_quadsec` *Construction of* '`cyl_quadsec`' *Objects*

Description

Constructs a circular-linear copula with cubic sections of class '`cyl_quadsec`'.

Usage

```
cyl_quadsec(a = 1/(2 * pi))
```

Arguments

a	numeric value of the parameter of the copula. It must be in $[-1/(2\pi)), 1/(2\pi))]$.

Value

An R object of class '`cyl_quadsec`'.

References

Quesada-Molina JJ, RodrÃguez-Lallena JA (1995). "Bivariate copulas with quadratic sections." *Journal of Nonparametric Statistics*, **5**(4), 323–337. ISSN 10290311, doi: 10.1080/10485259508832652, https://doi.org/10.1080/10485259508832652.

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

Examples

```
cop <- cyl_quadsec(a = 0.1)
cop_plot(copula = cop, type = "pdf", plot_type = "ggplot")
```

cyl_quadsec-class *An S4 Class of Bivariate Copulas with Quadratic Sections*

Description

This class contains bivariate circular-linear copulas with quadratic sections in the linear dimension. They are periodic in the circular dimension, u, and symmetric with respect to u=0.5. I.e the can capture correlation in data where there is symmetry between positive and negative angles. These copulas are described by one parameter, a.

Slots

`name` character string holding the name of the copula.

`parameters` numeric vector holding the parameter value.

`param.names` character vector holding the parameter name.

`param.lowbnd` numeric vector holding the lower bound of the parameter.

`param.upbnd` numeric vector holding the upper bound of the parameter.

Objects from the Class

Objects are created by `cyl_quadsec()`.

Extends

Class '`cyl_quadsec`' extends class '`cyl_copula`'.

References

Quesada-Molina JJ, RodrÃguez-Lallena JA (1995). "Bivariate copulas with quadratic sections." *Journal of Nonparametric Statistics*, **5**(4), 323–337. ISSN 10290311, doi: 10.1080/10485259508832652, https://doi.org/10.1080/10485259508832652.

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

`cyl_rect_combine` *Construction of '*`cyl_rect_combine`*' Objects*

Description

Constructs a circular-linear copula of class '`cyl_rect_combine`' from a rectangular patchwork of copulas.

Usage

```
cyl_rect_combine(
  copula,
  background = indepCopula(),
  low_rect = c(0, 0.5),
  up_rect = "symmetric",
  flip_up = TRUE
)
```

Arguments

`copula`	'`Copula`' object of the package '**copula**' or '`cyl_vonmises`' object, the copula in the rectangles.
`background`	'`cyl_vonmises`' or '`Copula`' object of the package '**copula**' (does not lead to an overall symmetric circular-linear copula), the copula where no rectangles overlay the unit square.
`low_rect`	numeric vector containing the lower and upper edge (u-value) of the lower rectangle.
`up_rect`	numeric vector containing the lower and upper edge (u-value) of the upper rectangle, or the character string "symmetric" if it should be the mirror image (with respect to u=0.5) of the lower rectangle.
`flip_up`	logical value indicating whether the copula (`sym.cop`) is rotated 90 degrees in the upper (`flip_up = TRUE`) or lower rectangle.

Value

An R object of class '`cyl_rect_combine`'.

References

Durante F, Saminger-Platz S, Sarkoci P (2009). "Rectangular patchwork for bivariate copulas and tail dependence." *Communications in Statistics - Theory and Methods*, **38**(15), 2515–2527. ISSN 03610926, doi: 10.1080/03610920802571203, https://doi.org/10.1080/03610920802571203.

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

Examples

```
#symmetric rectangles spanning entire unit square
cop <- cyl_rect_combine(copula::frankCopula(2))
cop_plot(copula = cop, type = "pdf", plot_type = "ggplot", resolution = 20)

#symmetric rectangles, independence copula as background
cop <- cyl_rect_combine(copula::frankCopula(2),
  low_rect = c(0, 0.3),
  up_rect = "symmetric",
  flip_up = FALSE
)
cop_plot(copula = cop, type = "pdf", plot_type = "ggplot", resolution = 20)

#symmetric rectangles, cy_quadsec-copula as background
cop <- cyl_rect_combine(copula::normalCopula(0.3),
  low_rect = c(0.1, 0.4),
  up_rect = "symmetric",
  background = cyl_quadsec(-0.1)
)
cop_plot(copula = cop, type = "pdf", plot_type = "ggplot", resolution = 20)

#asymmetric rectangles, von Mises copula as background.
#!!Not a symmetric circular linear copula!!
cop <- cyl_rect_combine(copula::normalCopula(0.3), low_rect = c(0.1, 0.4),
up_rect = c(0.5, 0.7), background = cyl_vonmises(mu = pi, kappa = 0.3))
cop_plot(copula = cop, type = "pdf", plot_type = "ggplot", resolution = 20)
```

---

cyl_rect_combine-class

An S4 Class of Circular-Linear Copulas Generated from a Rectangular Patchwork

---

Description

This class contains bivariate circular-linear copulas generated from linear-linear bivariate 'Copula' objects of the package '**copula**' or circular-linear copulas of class 'cyl_copula'. 2 non-overlapping rectangles are laid over the unit square, both have width 1 in v-direction. In the area covered by the first rectangle, the copula is derived from a linear-linear bivariate 'Copula' object. Rectangle 2 contains the same copula as rectangle 1, but 90 degrees rotated. In the area not covered by the rectangles, the "background", the copula is derived from a circular-linear 'cyl_copula' object. The copula regions are combined in a way that the overall result on the entire unit square is also a copula.

Details

With appropriate choices of the rectangles this results in copulas that are periodic in u-direction (and not in v-direction) and therefore are circular-linear. When the 2 rectangles are mirror images

with respect to $u = 0.5$, the resulting overall copula is symmetric with respect to $u = 0.5$, i.e. there is symmetry between positive and negative angles.

Note that as "background copula", we can also chose a linear-linear copula, the overall result will then, however, not be a symmetric circular linear copula.

Slots

`name` character string holding the name of the copula.

`parameters` numeric vector holding the parameter values.

`param.names` character vector the parameter names.

`param.lowbnd` numeric vector holding the lower bounds of the parameters.

`param.upbnd` numeric vector holding the upper bounds of the parameters.

`sym.cop` 'Copula' object of the package '**copula**' or 'cyl_vonmises' object. The copula in the rectangles.

`background.cop` 'cyl_vonmises' or 'Copula' object of the package '**copula**' (does not lead to an overall symmetric circular-linear copula). The copula where no rectangles overlay the unit square.

`flip_up` logical value indicating whether the copula (`sym.cop`) is rotated 90 degrees in the upper or lower rectangle.

`sym_rect` logical value indicating whether the upper rectangle was forced to be a mirror image of the lower one with respect to u=0.5 at the construction of the object.

Objects from the Class

Objects are created by `cyl_rect_combine()`.

Extends

Class '`cyl_rect_combine`' extends class 'Copula'.

References

Durante F, Saminger-Platz S, Sarkoci P (2009). "Rectangular patchwork for bivariate copulas and tail dependence." *Communications in Statistics - Theory and Methods*, **38**(15), 2515–2527. ISSN 03610926, doi: 10.1080/03610920802571203, https://doi.org/10.1080/03610920802571203.

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

`cyl_rot_combine` *Construction of '*`cyl_rot_combine`*' Objects*

Description

Constructs a circular-linear copula of class '`cyl_rot_combine`' from linear combinations of copulas.

Usage

```
cyl_rot_combine(copula, shift = FALSE)
```

Arguments

`copula`	linear-linear 2-dimensional '`Copula`' object of the package '**copula**'.
`shift`	logical value indicating whether the (u-periodic) copula should be shifted by 0.5 in u direction.

Value

An R object of class '`cyl_rot_combine`'.

References

Nelsen RB (2006). *An Introduction to Copulas*, volume 139 of *Lecture Notes in Statistics*. Springer New York, New York, NY. ISBN 978-0-387-98623-4, doi: 10.1007/9781475730760, https://doi.org/10.1007/978-1-4757-3076-0.

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

Examples

```
cop <- cyl_rot_combine(copula = copula::frankCopula(param = 3), shift = TRUE)
cop_plot(copula = cop, type = "pdf", plot_type = "ggplot", resolution = 20)

cop <- cyl_rot_combine(copula = copula::claytonCopula(param = 10), shift = FALSE)
cop_plot(copula = cop, type = "pdf", plot_type = "ggplot", resolution = 20)
```

`cyl_rot_combine-class` *An S4 Class of Circular-Linear Copulas generated from Linear Combinations of Copulas*

Description

This class contains bivariate circular-linear copulas, generated from linear-linear bivariate '`Copula`' objects of the package '**copula**', by taking the arithmetic mean of the original copula and the 90 deg rotated copula. This results in copulas that are periodic in the circular dimension, u, and symmetric with respect to $u = 0.5$, i.e. positive and negative angles.

Slots

`name` character string holding the name of the copula.

`parameters` numeric vector holding the parameter values.

`param.names` character vector the parameter names.

`param.lowbnd` numeric vector holding the lower bounds of the parameters.

`param.upbnd` numeric vector holding the upper bounds of the parameters.

`orig.cop` linear-linear 2-dimensional '`Copula`' object of the package '**copula**'.

`shift` logical value indicating whether the (u-periodic) copula should be shifted by 0.5 in u direction.

Objects from the Class

Objects are created by `cyl_rot_combine()`.

Extends

Class '`cyl_rot_combine`' extends class '`Copula`'.

References

Nelsen RB (2006). *An Introduction to Copulas*, volume 139 of *Lecture Notes in Statistics*. Springer New York, New York, NY. ISBN 978-0-387-98623-4, doi: 10.1007/9781475730760, https://doi.org/10.1007/978-1-4757-3076-0.

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

cyl_vonmises *Construction of* 'cyl_vonmises' *Objects*

Description

Constructs a circular-linear von Mises copula according to Johnson and Wehrly (1978) of class 'cyl_vonmises'.

Usage

```
cyl_vonmises(mu = 0, kappa = 1, flip = FALSE)
```

Arguments

mu	numeric value giving the mean of the vonMises function used to construct the copula.
kappa	numeric value giving the concentration of the vonMises function used to construct the copula.
flip	logical value indicating whether the copula should be rotated 90 degrees to capture negative correlation.

Value

An R object of class 'cyl_vonmises'.

References

Johnson RA, Wehrly TE (1978). "Some Angular-Linear Distributions and Related Regression Models." *Journal of the American Statistical Association ISSN:*, **73**(363), 602–606. ISSN 00401706, doi: 10.2307/1270921, https://doi.org/10.2307/1270921.

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

Examples

```
cop <- cyl_vonmises(mu=pi, kappa=10, flip = TRUE)
cop_plot(copula = cop, type = "pdf", plot_type = "ggplot", resolution = 20)

cop <- cyl_vonmises(mu=0, kappa=8, flip = FALSE)
cop_plot(copula = cop, type = "pdf", plot_type = "ggplot", resolution = 20)
```

`cyl_vonmises-class` *An S4 Class of Bivariate vonMises Copulas*

Description

This class contains circular-linear copulas that are based on the approach by Johnson and Wehrly (1978) with a von Mises periodic function. They are periodic in the circular dimension, u, but not symmetric with respect to $u = 0.5$ i.e. there is no symmetry between positive and negative angles.

Slots

`name` character string holding the name of the copula.

`parameters` numeric vector holding the parameter values.

`param.names` character vector holding the parameter names.

`param.lowbnd` numeric vector holding the lower bounds of the parameters.

`param.upbnd` numeric vector holding the upper bounds of the parameters.

`flip` logical value indicating whether the copula should be rotated 90 degrees to capture negative correlation.

Objects from the Class

Objects are created by `cyl_vonmises()`.

Extends

Class '`cyl_vonmises`' extends class '`cyl_copula`'.

References

Johnson RA, Wehrly TE (1978). "Some Angular-Linear Distributions and Related Regression Models." *Journal of the American Statistical Association ISSN:*, **73**(363), 602–606. ISSN 00401706, doi: 10.2307/1270921, https://doi.org/10.2307/1270921.

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

dens *Density, Distribution, Random Number Generation and Quantiles of Kernel Density Estimates*

Description

Calculate the density (ddens()), the distribution (pdens()), the quantiles (qdens()) and generate random samples (rdens()) of a kernel density estimate as returned by fit_angle() or fit_steplength().

Usage

```
rdens(n, density)

ddens(x, density)

pdens(x, density)

qdens(p, density)
```

Arguments

n	integer value, the number of random samples to be generated with rdens().
density	list containing information about the kernel density estimate. The structure of the list must be as returned by fit_angle() or fit_steplength().
x	numeric vector giving the points where the density or distribution function is evaluated.
p	numeric vector giving the probabilities where the quantile function is evaluated.

Value

ddens() and pdens() give a vector of length length(x) containing the density or distribution function at the corresponding values of x. qdens() gives a vector of length length(p) containing the quantiles at the corresponding values of p. The function rdens() generates a vector of length n containing the random samples.

See Also

fit_angle(), fit_steplength(), fit_steplength().

Examples

```
set.seed(123)

steps <- rweibull(10, shape=3)
dens <- fit_steplength(x = steps, parametric = FALSE)
ddens(c(0.1,0.3), dens)
```

```
pdens(c(0.1,0.3), dens)
qdens(c(0.1,0.3), dens)
rdens(4, dens)
```

`fit_angle` *Fit a Circular Univariate Distribution*

Description

This function finds parameter estimates of the marginal circular distribution (with potentially fixed mean), or gives a kernel density estimate using a von Mises smoothing kernel.

Usage

```
fit_angle(
  theta,
  parametric = c("vonmises", "wrappedcauchy", "mixedvonmises", FALSE),
  bandwidth = NULL,
  mu = NULL
)
```

Arguments

`theta`	numeric vector of angles in $[-\pi, \pi)$.
`parametric`	either a character string describing what distribution should be fitted (`"vonmises"`, `"wrappedcauchy"`, or `"mixedvonmises"`), or the logical `FALSE` if a non-parametric estimation (kernel density) should be made.
`bandwidth`	If `parametric = FALSE`, the numeric value of the kernel density bandwidth. Default is `cylcop::opt_circ_bw(theta,"adhoc")`.
`mu`	(optional) numeric vector, fixed mean direction(s) of the parametric distribution.

Value

If a parametric estimate is made, a list containing the estimated parameters, their standard errors and the log-likelihood is returned. If a non-parametric estimate is made, the output is a list and comes directly from the function `circular::density.circular()` of the '**circular**' package.

See Also

`circular::density.circular()`, `fit_angle()`, `opt_circ_bw()`.

Examples

```
require(circular)
require(graphics)
set.seed(123)

silent_curr <- cylcop_get_option("silent")
cylcop_set_option(silent = TRUE)

n <- 100  #n (number of samples) is set small for performance. Increase n to
# a value larger than 1000 to see the effects of multimodality

angles <- circular::rmixedvonmises(n,
  mu1 = circular::circular(0),
  mu2 = circular::circular(pi),
  kappa1 = 2,
  kappa2 = 1,
  prop = 0.5
)
angles <- as.double(angles)

bw <- opt_circ_bw(theta = angles,
  loss="adhoc",
  kappa.est = "trigmoments"
)
dens_non_param <- fit_angle(theta = angles,
  parametric = FALSE,
  bandwidth = bw
)

param_estimate <- fit_angle(theta = angles,
  parametric = "mixedvonmises"
)
param_estimate_fixed_mean <- fit_angle(theta = angles,
  parametric = "mixedvonmises",
  mu = c(0, pi)
)

true_dens <- circular::dmixedvonmises(circular(seq(-pi,pi,0.001)),
  mu1 = circular(0),
  mu2 = circular(pi),
  kappa1 = 2,
  kappa2 = 1,
  prop = 0.5
)
dens_estimate <- circular::dmixedvonmises(circular(seq(-pi,pi,0.001)),
  mu1 = circular::circular(param_estimate$coef$mu1),
  mu2 = circular::circular(param_estimate$coef$mu2),
  kappa1 = param_estimate$coef$kappa1,
  kappa2 = param_estimate$coef$kappa2,
  prop = param_estimate$coef$prop
)
dens_estimate_fixed_mean <- circular::dmixedvonmises(circular(seq(-pi,pi,0.001)),
```

```
  mu1 = circular(param_estimate_fixed_mean$coef$mu1),
  mu2 = circular(param_estimate_fixed_mean$coef$mu2),
  kappa1 = param_estimate_fixed_mean$coef$kappa1,
  kappa2 = param_estimate_fixed_mean$coef$kappa2,
  prop = param_estimate_fixed_mean$coef$prop
)

plot(seq(-pi, pi, 0.001), true_dens, type = "l")
lines(as.double(dens_non_param$x), as.double(dens_non_param$y), col = "red")
lines(seq(-pi, pi, 0.001), dens_estimate, , col = "green")
lines(seq(-pi, pi, 0.001), dens_estimate_fixed_mean, , col = "blue")

cylcop_set_option(silent = silent_curr)
```

`fit_steplength` *Fit a Linear Univariate Distribution*

Description

This function finds parameter estimates of the marginal linear distribution, or gives a kernel density estimate using a Gaussian smoothing kernel.

Usage

```
fit_steplength(
  x,
 parametric = c("beta", "cauchy", "chi-squared", "exponential", "gamma", "lognormal",
    "logistic", "normal", "t", "weibull", FALSE),
  start = NULL,
  bandwidth = NULL
)
```

Arguments

`x`	numeric vector of measurements of a linear random variable in $[0, \infty)$.
`parametric`	either a character string describing what distribution should be fitted (`"beta"`, `"cauchy"`, `"chi-squared"`, `"exponential"`, `"gamma"`, `"lognormal"`, `"logistic"`, `"normal"`, `"t"`, or `"weibull"`), or the logical `FALSE` if a non-parametric estimation (kernel density) should be made.
`start`	(optional, except when `parametric = "chi-squared"`) named list containing the parameters to be optimized with inital values.
`bandwidth`	numeric value for the kernel density bandwidth. Default is `cylcop::opt_lin_bw(x,"adhoc")`.

Value

If a parametric estimate is made, a list containing the estimated parameters, their standard errors and the log-likelihood is returned. If a non-parametric estimate is made, , the output is a list, and comes directly from the function `GoFKernel::density.reflected()` of the '**GoFKernel**' package.

See Also

GoFKernel::density.reflected(), fit_angle(), opt_lin_bw().

Examples

```
require(graphics)
set.seed(123)

silent_curr <- cylcop_get_option("silent")
cylcop_set_option(silent = TRUE)

n <- 10000

x <- rweibull(n, shape = 10)

dens_non_param <- fit_steplength(x = x, parametric = FALSE)
weibull <- fit_steplength(x = x, parametric = "weibull")
gamma <- fit_steplength(x = x, parametric = "gamma")
chisq <- fit_steplength(x = x, parametric = "chi-squared", start = list(df = 1))

true_dens <- dweibull(seq(0, max(x), length.out = 200),
  shape = 10
)
dens_weibull <- dweibull(seq(0, max(x),length.out = 200),
  shape = weibull$coef$shape,
  scale = weibull$coef$scale
)
dens_gamma <- dgamma(seq(0, max(x),length.out = 200),
  shape = gamma$coef$shape,
  rate = gamma$coef$rate
)
dens_chisq <- dchisq(seq(0, max(x),length.out = 200),
  df = chisq$coef$df
)

plot(seq(0,max(x),length.out = 200), true_dens, type = "l")
lines(dens_non_param$x, dens_non_param$y, col = "red")
lines(seq(0,max(x),length.out = 200), dens_weibull, col = "green")
lines(seq(0,max(x),length.out = 200), dens_gamma, col = "blue")
lines(seq(0,max(x),length.out = 200), dens_chisq, col = "cyan")

cylcop_set_option(silent = silent_curr)
```

---

full2half_circ *Convert Angle from Full Circle to Half Circle*

Description

Converts an angle from the full circle (i.e. in the interval $[0, 2\pi)$) to an angle on the half circle (i.e. in the interval $[-\pi, \pi)$).

Usage

```
full2half_circ(angle)
```

Arguments

angle	numeric value of an angle or a circular-objekt in $[0, 2\pi)$.

Value

The numeric value of the angle in $[-\pi, \pi)$.

Examples

```
full2half_circ(0 * pi) / pi
full2half_circ(0.5 * pi) / pi
full2half_circ(1 * pi) / pi
full2half_circ(1.5 * pi) / pi
full2half_circ(2 * pi) / pi
```

---

half2full_circ *Convert Angle from Half Circle to Full Circle*

Description

Converts an angle from the half circle (i.e. in the interval $[-\pi, \pi)$) to an angle on the full circle (i.e. in the interval $[0, 2\pi)$).

Usage

```
half2full_circ(angle)
```

Arguments

angle	numeric value of an angle or a circular-objekt in $[-\pi, \pi)$.

Value

The numeric value of the angle in $[0, 2\pi)$.

Examples

```
half2full_circ(-1 * pi) / pi
half2full_circ(-0.5 * pi) / pi
half2full_circ(-0 * pi) / pi
half2full_circ(0.5 * pi) / pi
```

make_traj *Generate a Trajectory with Correlated Step Lengths and Turn Angles*

Description

The function draws values from a circular-linear bivariate distribution of turn angles and step lengths specified by the marginal distributions and a circular-linear copula. Samples are drawn from the copula and then transformed using the quantile functions of the marginal distributions. From the start point (0,0) and the second (user specified) point, a trajectory is then built with these turn angles and step lengths.

Usage

```
make_traj(
  n,
  copula,
  marginal_circ = c("vonmises", "wrappedcauchy", "mixedvonmises", "dens"),
  parameter_circ,
  marginal_lin,
  parameter_lin,
  pos_2 = c(1, 0)
)
```

Arguments

n	integer, number of trajectory steps to generate.
copula	'cyl_copula' object.
marginal_circ	character string denoting the name of the circular distribution. It can be "vonmises", "mixedvonmises", "wrappedcauchy", or "dens" (for kernel density estimate).
parameter_circ	(named) list of parameters of the circular marginal distribution as taken by the functions qvonmises(), qmixedvonmises(), or qwrappedcauchy(). If marginal_circ = "dens", parameter_circ must be a named list, containing information on the kernel density estimate, which can be obtained using fit_angle(...,parametric = FALSE).
marginal_lin	character string denoting the name of the linear distribution, i.e. the name of its distribution function without the "p", e.g. "norm" for normal distribution, or "dens" (for kernel density estimate).
parameter_lin	(named) list of parameters of the linear marginal distribution. For marginal_lin = "dens", parameter_lin must be a named list, containing information on the kernel density estimate, which can be obtained using fit_steplength(...,parametric = FALSE).
pos_2	numeric vector containing the coordinates of the second point in the trajectory. The first point is always at (0,0).

Value

A data.frame containing the trajectory. It has 6 columns containing the x and y coordintates, the step lengths, the turn angles, and the values sampled from the copula.

See Also

fit_steplength(), fit_angle(), traj_plot(), cop_scat_plot(), scat_plot(), circ_plot().

Examples

```
require(circular)
set.seed(123)

traj <- make_traj(5,
  copula = cyl_quadsec(0.1),
  marginal_circ = "vonmises",
  parameter_circ = list(0, 1),
  marginal_lin = "weibull",
  parameter_lin = list(shape=3)
)
traj

angles <- circular::rmixedvonmises(100,
  mu1 = circular::circular(0),
  mu2 = circular::circular(pi),
  kappa1 = 2,
  kappa2 = 3,
  prop = 0.4
)
angles <- full2half_circ(angles)
bw <- opt_circ_bw(theta = angles,loss = "adhoc", kappa.est = "trigmoments")
dens <- fit_angle(theta = angles, parametric = FALSE, bandwidth = bw)
make_traj(5,
  copula = cyl_quadsec(0.1),
  marginal_circ = "dens",
  parameter_circ = dens,
  marginal_lin = "weibull",
  parameter_lin = list(shape=3),
  pos_2 = c(5,5)
)
```

---

mi_cyl *Estimate the Mutual Information Between a Circular and a Linear Random Variable*

Description

The mutual information can be normalized to lie between 0 ans 1 by dividing by the product of the entropies of `x` and `theta`. Even if `x` and `theta` are perfectly correlated, the normalized mutual information will not be 1 if the underlying copula is periodic and symmetric. Therefore, we can set `symmetrize = TRUE` to set all u-values of the empirical copula that are larger than 0.5 to $1-0.5$. The mutual information is then calculated from those values and the is exactly 1 in the case of perfect correlation as captured by e.g. `cyl_rect_combine(normalCopula(1))`. The estimate (output of `mi_cyl()`) will be less than one for numerical reasons. Note also that the mutual information is independent of the marginal distributions.

Usage

```
mi_cyl(theta, x, normalize = TRUE, symmetrize = FALSE)
```

Arguments

`theta`	numeric vector of angles (measurements of a circular variable).
`x`	numeric vector of step lengths (measurements of a linear variable).
`normalize`	logical value whether the mutual information should be normalized to lie within $[0, 1]$.
`symmetrize`	logical value whether it should be assumed that positive and negative angles are equivalent.

Value

A numeric value, the mutual information between `theta` and `x`.

References

Ma J, Sun Z (2011). "Mutual Information Is Copula Entropy." *Tsinghua Science and Technology*, **16**(1), 51–54. ISSN 1007-0214, doi: 10.1016/S10070214(11)700086, https://www.sciencedirect.com/science/article/pii/S1007021411700086/.

Calsaverini RS, Vicente R, Systems C, Artes ED (2009). "An information-theoretic approach to statistical dependence: Copula information." *Europhysics Letters*, **88**(6), 1–6. doi: 10.1209/0295-5075/88/68003, https://iopscience.iop.org/article/10.1209/0295-5075/88/68003/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

See Also

`cor_cyl()`, `optCor()`.

Examples

```
set.seed(123)

cop <- cyl_quadsec(0.1)

#draw samples and calculate the mutual information.
sample <- rcylcop(100, cop)
mi_cyl(theta = sample[,1],
  x = sample[,2],
  normalize = TRUE,
  symmetrize = FALSE
)

#the correlation coefficient is independent of the marginal distribution.
sample <- make_traj(100,
  cop,
  marginal_circ = "vonmises",
  parameter_circ = list(0, 1),
  marginal_lin = "weibull",
  parameter_lin = list(shape = 2)
)
mi_cyl(theta = sample$angle,
  x = sample$steplength,
  normalize = TRUE,
  symmetrize = FALSE)
mi_cyl(theta = sample$cop_u,
  x = sample$cop_v,
  normalize = TRUE,
  symmetrize = FALSE)

# Estimate correlation of samples drawn from circular-linear copulas
# with perfect correlation.
cop <- cyl_rect_combine(copula::normalCopula(1))
sample <- rcylcop(100, cop)
# without normalization
mi_cyl(theta = sample[,1],
  x = sample[,2],
  normalize = FALSE,
  symmetrize = FALSE
)
#with normalization
mi_cyl(theta = sample[,1],
  x = sample[,2],
  normalize = TRUE,
  symmetrize = FALSE
)
#only with normaliztion and symmetrization do we get a value close to 1
mi_cyl(theta = sample[,1],
  x = sample[,2],
  normalize = TRUE,
  symmetrize = TRUE
)
```

`mle.mixedvonmises` *Mixed von Mises Maximum Likelihood Estimates*

Description

Computes the maximum likelihood estimates for the parameters of a mixed von Mises distribution: the mean directions, the concentration parameters, and the proportion of the 2 distributions. The code is a simplified version of `movMF::movMF()` with the added feature of optionally fixed mean directions (Hornik and GrÃ¼n 2014).

Usage

```
mle.mixedvonmises(theta, mu = NULL)
```

Arguments

`theta`	numeric vector of angles.
`mu`	(optional) numeric vector of length 2 holding the 2 mean directions (angles). If not specified the mean directions are estimated.

Details

The function complements the '**circular**' package, which provides functions to make maximum likelihood estimates of e.g. von Mises (`circular::mle.vonmises()`), or wrapped Cauchy distributions (`circular::mle.wrappedcauchy()`)

Value

A list containing the optimized parameters `mu1`, `mu2`, `kappa1`, `kappa2` and `prop`.

References

Hornik K, GrÃ¼n B (2014). "movMF : An R Package for Fitting Mixtures of von Mises-Fisher Distributions." *Journal of Statistical Software*, **58**. doi: 10.18637/jss.v058.i10, https://doi.org/10.18637/jss.v058.i10..

See Also

`movMF::movMF()`, `circular::mle.vonmises()`, `circular::dmixedvonmises()`, `qmixedvonmises()`.

Examples

```
set.seed(123)

n <- 1000
angles <- circular::rmixedvonmises(n,
  mu1 = circular::circular(0),
  mu2 = circular::circular(pi),
  kappa1 = 2,
  kappa2 = 1,
  prop = 0.4
)
angles <- as.double(angles)
mle.mixedvonmises(theta = angles)
mle.mixedvonmises(theta = angles, mu = c(0, pi))
```

`numerical_conditional_cop` *Numerically Calculate the Conditional Copula*

Description

Numerically Calculate the Conditional Copula

Usage

```
numerical_conditional_cop(u, copula, cond_on)
```

Arguments

u	matrix or vector of numeric values in I^2, containing as first column the circular (periodic) and as second the linear dimension.
copula	R object of class '`cyl_copula`' or '`Copula`' (package '**copula**', only 2-dimensional).
cond_on	column number of u on which the copula is conditioned. E.g. if cond_on = 2, the function calculates for each element in the first column of u the copula conditional on the element in the second column.

Value

A vector containing the values of the distribution of the copula at `[u,-cond_on]` conditional on the values of `[u,cond_on]`.

References

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

See Also

ccylcop(), numerical_inv_conditional_cop().

Examples

```
cop <- cyl_quadsec(0.1)
u <- cbind(c(0.3, 0.1), c(0.7, 0.3))
numerical_conditional_cop(u = u, cop = cop, cond_on = 1)
```

---

numerical_inv_conditional_cop

Numerically calculate the inverse of the conditional copula

Description

Numerically calculate the inverse of the conditional copula

Usage

```
numerical_inv_conditional_cop(u, copula, cond_on)
```

Arguments

u	matrix or vector of numeric values in I^2, containing as first column the circular (periodic) and as second the linear dimension.
copula	R object of class 'cyl_copula' or 'Copula' (package '**copula**', only 2-dimensional).
cond_on	column number of u on which the copula is conditioned. E.g if cond_on = 2, the function calculates for each element in the first column of u the inverse of the Copula conditional on the element in the second column.

Value

A vector containing the values of the inverse distribution of the copula at [u,-cond_on] conditional on the values of [u,cond_on].

References

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

See Also

ccylcop(), numerical_conditional_cop().

Examples

```
cop <- cyl_quadsec(0.1)
u <- cbind(c(0.3, 0.1), c(0.7, 0.3))
numerical_inv_conditional_cop(u = u, cop = cop, cond_on = 1)
```

---

optCor *Estimate Copula Parameters from Correlation Measures*

Description

See individual methods for more detailed explanations. Kendall's tau is only available for 'cyl_rect_combine' copulas, for which it is the recommended method to use.

Usage

```
optCor(
  copula,
  theta,
  x,
  acc = NULL,
  n = 10000,
  method = c("cor_cyl", "mi_cyl", "tau"),
  ...
)

## S4 method for signature 'cyl_vonmises'
optCor(copula, theta, x, acc, n, method = "cor_cyl")

## S4 method for signature 'cyl_quadsec'
optCor(copula, theta, x, acc, n, method = "cor_cyl")

## S4 method for signature 'cyl_cubsec'
optCor(copula, theta, x, acc, n, method = "cor_cyl", parameter = "both")

## S4 method for signature 'cyl_rot_combine'
optCor(copula, theta, x, acc, n, method = "mi_cyl")

## S4 method for signature 'cyl_rect_combine'
optCor(copula, theta, x, acc, n, method = "tau", background = FALSE)
```

Arguments

copula	R object of class 'cyl_copula'.
theta	numeric vector of angles (measurements of a circular variable).
x	numeric vector of step lengths (measurements of a linear variable).

acc	numeric value, the interval of the copula parameter at which to evaluate the correlation.
n	numeric value, the number of sample points at each optimization step.
method	character string describing what correlation metric to use. Either a rank-based circular-linear coefficient ("cor_cyl"), mutual information ("mi_cyl"), or Kendall's tau ("tau").
...	Additional parameters (see individual methods).
parameter	A character string specifying which parameter of a 'cyl_cubsec' copula to optimize, "a", "b", or "both"
background	logical value describing whether to optimize the parameter of the background copula, (background = TRUE) or the one of the copula in the rectangles(background = FALSE).

Value

numeric vector containing the estimated parameter value(s).

Methods (by class)

- cyl_vonmises: only parameter "kappa" can be optimized, since parameter "mu" does not influence the correlation.
- cyl_quadsec: the absolute value of the parameter is optimized, positive and negative values give the same correlation.
- cyl_cubsec: optimization of parameters, "a" and "b", can be done separately or simultaneously.
- cyl_rot_combine: the circular-linear correlation coefficient will give a value close to 0 for any parameter value. It therefore only makes sense to use method = "mi_cyl" for the optimization.
- cyl_rect_combine: it is recommended to use method = "tau", since this calculates the copula parameter analytically.

References

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

See Also

mi_cyl(), cor_cyl(), optML(), opt_auto(), copula::fitCopula().

Examples

```
set.seed(123)

sample <- rcylcop(1000, cyl_rect_combine(copula::frankCopula(2)))
optCor(cyl_rect_combine(copula::frankCopula()),
  theta = sample[,1],
  x = sample[,2],
  method = "tau"
)

optCor(cyl_rect_combine(copula::frankCopula()),
  theta = sample[,1],
  x = sample[,2],
  method = "mi_cyl",
  n = 1000
)

optCor(cyl_rect_combine(copula::claytonCopula()),
  theta = sample[,1],
  x = sample[,2],
  method = "tau"
)

optCor(cyl_quadsec(), theta = sample[,1], x = sample[,2], method = "mi_cyl")
optCor(cyl_quadsec(), theta = sample[,1], x = sample[,2], method = "cor_cyl")
optCor(cyl_quadsec(),
  theta = sample[,1],
  x = sample[,2],
  method = "cor_cyl",
  n = 1000,
  acc = 0.001
)
```

`optML` *Estimate Parameters of a Circular-Linear Copula According to Maximum Likelihood*

Description

The code of this function is based on `copula::fitCopula()`. A circular-linear copula is fit to a set of bivariate observations.

Usage

```
optML(
  copula,
  theta,
  x,
```

```
  parameters,
  start,
  lower = NULL,
  upper = NULL,
  optim.method = "L-BFGS-B",
  optim.control = list(maxit = 100),
  estimate.variance = FALSE,
  traceOpt = FALSE
)
```

Arguments

copula	R object of class 'cyl_copula'.
theta	numeric vector of angles (measurements of a circular variable).
x	numeric vector of step lengths (measurements of a linear variable).
parameters	vector of character strings holding the names of the parameters to be optimized. These can be any parameters in copula@parameters.
start	vector of staring values of the parameters.
lower	OPTIONAL: vector of lower bounds of the parameters.
upper	OPTIONAL: vector of upper bounds of the parameters.
optim.method	character string, optimizer used in optim(), can be "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN", or "Brent".
optim.control	list of additional controls passed to optim().
estimate.variance	logical value, denoting whether to include an estimate of the variance (NOT YET IMPLEMENTED).
traceOpt	logical value, whether to print information regarding convergence, current values, etc. during the optimization process.

Value

A list of length 3 containing the same type of 'cyl_copula' object as copula, but with optimized parameters, the log-likelihood and the AIC.

References

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

See Also

copula::fitCopula(), optCor(), optML(), opt_auto().

Examples

```
set.seed(123)

#optimization of copula is independent of the marginals
sample <- rcylcop(100,cyl_quadsec(0.1))
optML(copula = cyl_quadsec(),
  theta = sample[,1],
  x = sample[,2],
  parameters = "a",
  start = 0
)
optML(copula = cyl_rect_combine(copula::frankCopula()),
  theta = sample[,1],
  x = sample[,2],
  parameters = "alpha",
  start = 1
)
```

`opt_auto` *Automatically Find the Best Fitting Copula*

Description

The parameters of 15 different circular-linear copulas are fitted to data and sorted according to AIC. For each copula, first, a starting value for the maximum likelihood estimation (MLE) is found using `optCor()`. Then MLE with a "reasonable" setup is carried out using `optML()`. If MLE fails, parameters obtained with `optCor()` are reported.

Usage

```
opt_auto(theta, x)
```

Arguments

`theta`	numeric vector of angles (measurements of a circular variable).
`x`	numeric vector of step lengths (measurements of a linear variable).

Value

A list containing 3 lists: Descriptions of the copulae, the '`cyl_copula`' objects with fitted parameters, and the AIC. The lists are sorted by ascending AIC. If `optML()` has failed, the reported parameters are the ones with `optCor()` and the AIC is set to `NA`.

References

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

See Also

optCor(), optML()

Examples

```
set.seed(123)

#Optimal copula is independent of marginals.
data <- rcylcop(100,cyl_quadsec(0.1))

#This takes a few seconds to run.
copula_lst <- opt_auto(theta = data[,1], x = data[,2])
```

---

opt_circ_bw *Find the Optimal Bandwidth for a Circular Kernel Density Estimate*

Description

This function is basically wraps `circular::bw.cv.ml.circular()` and `circular::bw.nrd.circular()` of the '**circular**' package, simplifying their inputs. For more control, these '**circular**' functions could be used directly. The ad-hoc method of finding the bandwidth parameter might give very bad results, especially for multimodal population distributions. In these cases it can help to set `kappa.est = trigmoments`.

Usage

```
opt_circ_bw(theta, loss = c("KullbackLeibler", "adhoc"), kappa.est = "ML")
```

Arguments

`theta`	numeric vector of angles in $[-\pi, \pi)$.
`loss`	character string describing the loss function, either `"KullbackLeibler"` (leading to a maximum likelihood estimate), or `"adhoc"` leading to a rule-of-thumb estimate.
`kappa.est`	character string describing how the spread is estimated. Either maximum likelihood `"ML"`, or trigonometric moment `"trigmoments"`.

Value

A numeric value, the optimized bandwidth.

See Also

circular::bw.cv.ml.circular(), circular::bw.nrd.circular(), opt_circ_bw().

Examples

```
require(circular)
require(graphics)
set.seed(123)
n <- 100  #n (number of samples) is set small for performance. Increase n to
# a value larger than 1000 to see the effects of multimodality

angles <- circular::rmixedvonmises(n,
  mu1 = circular::circular(0),
  mu2 = circular::circular(pi),
  kappa1 = 2,
  kappa2 = 1,
  prop = 0.5
)
angles <- as.double(angles)
bw1 <- opt_circ_bw(theta = angles, loss="adhoc")
bw2 <- opt_circ_bw(theta = angles, loss="adhoc", kappa.est = "trigmoments")
bw3 <- opt_circ_bw(theta = angles, loss="KullbackLeibler")

dens1 <- fit_angle(theta = angles, parametric = FALSE, bandwidth = bw1)
dens2 <- fit_angle(theta = angles, parametric = FALSE, bandwidth = bw2)
dens3 <- fit_angle(theta = angles, parametric = FALSE, bandwidth = bw3)
true_dens <- circular::dmixedvonmises(
  circular(seq(-pi,pi,0.001)),
  mu1 = circular::circular(0),
  mu2 = circular::circular(pi),
  kappa1 = 2,
  kappa2 = 1,
  prop = 0.5
)

plot(seq(-pi, pi, 0.001), true_dens, type = "l")
lines(as.double(dens1$x), as.double(dens1$y), col = "red")
lines(as.double(dens2$x), as.double(dens2$y), col = "green")
lines(as.double(dens3$x), as.double(dens3$y), col = "blue")
```

---

opt_lin_bw *Find the Optimal Bandwidth for a Linear Kernel Density Estimate*

Description

This function is basically wraps `stats::bw.ucv()` and `stats::bw.nrd()` of the '**stats**' package, simplifying their inputs. For more control, these '**stats**' functions could be used directly.

Usage

```
opt_lin_bw(x, loss = c("KullbackLeibler", "adhoc"))
```

Arguments

`x`	numeric vector of linear measurements.
`loss`	character string describing the loss function, either `"KullbackLeibler"` (leading to an maximum likelihood estimate), or `"adhoc"` leading to a rule-of-thumb estimate.

Value

A numeric value, the optimized bandwidth.

See Also

`stats::bw.ucv()`, `stats::bw.nrd()` `opt_lin_bw()`.

Examples

```
require(graphics)
set.seed(123)
n <- 1000

x <- rweibull(n, shape = 10)
bw1 <- opt_lin_bw(x = x, loss="adhoc")
bw2 <- opt_lin_bw(x = x, loss="KullbackLeibler")

dens1 <- fit_steplength(x = x, parametric = FALSE, bandwidth = bw1)
dens2 <- fit_steplength(x = x, parametric = FALSE, bandwidth = bw2)
true_dens <- dweibull(seq(0,max(x),length.out = 200), shape = 10)

plot(seq(0,max(x),length.out = 200), true_dens, type = "l")
lines(dens1$x, dens1$y, col = "red")
lines(dens2$x, dens2$y, col = "green")
```

plot,cyl_copula,missing-method
Plot 'cyl_copula' Objects

Description

Methods for `plot()` to draw a scatter plot of a random sample from bivariate distributions from package **cylcop**.

Usage

```
## S4 method for signature 'cyl_copula,missing'
plot(x, n = 1000, ...)
```

Arguments

x	R object of class 'cyl_copula'.
n	sample size of the random sample drawn from x.
...	additional arguments passed to `plot()`.

Value

An invisible `NULL`. As side effect, a plot is produced.

Examples

```
set.seed(123)

plot(cyl_quadsec(0.1))
plot(cyl_vonmises(0,2), n=100)
```

prob,cyl_copula-method
Calculate the C-Volume of a 'cyl_copula' Copula

Description

This is a method corresponding to the generic `prob()` in the '**copula**' package.

Usage

```
## S4 method for signature 'cyl_copula'
prob(x, l, u)
```

Arguments

x	R object of class 'cyl_copula'.
l	numeric vector of length 2 holding the coordinates of the lower left corner in $[0, 1]^2$.
u	numeric vector of length 2 holding the coordinates of the upper right corner in $[0, 1]^2$.

Value

A numeric in $[0, 1]$, the probability that a draw from the 2-dimensional copula x falls in the rectangle defined by l and u.

See Also

copula::prob

Examples

```
cop <- cyl_quadsec(0.1)
prob(cop, l = c(0.1, 0.3), u = c(0.3, 0.9))
```

---

qmixedvonmises *Quantiles of the Mixed von Mises Distribution*

Description

The quantiles are numerically obtained from the distribution function using monotone cubic splines.

Usage

```
qmixedvonmises(p, mu1, mu2, kappa1, kappa2, prop)
```

Arguments

p	numeric vector giving the probabilities where the quantile function is evaluated.
mu1	numeric value, mean direction of the first component.
mu2	numeric value, mean direction of the second component.
kappa1	numeric value, concentration parameter of the first component.
kappa2	numeric value, concentration parameter of the second component.
prop	numeric value, mixing proportion.

Value

a vector of length length(p), the quantiles of the mixed von Mises distribution.

See Also

circular::dmixedvonmises(), circular::pmixedvonmises(), circular::rmixedvonmises().

Examples

```
qmixedvonmises(p = c(0.1, 0.8),
  mu1 = 0,
  mu2 = pi,
  kappa1 = 1,
  kappa2 = 3,
  prop = 0.4
)
```

---

scat_plot *Scatterplot of Turn Angles ans Step Lengths*

Description

This function produces a scatterplot ('ggplot' object) of the turn angles and step lengths.

Usage

```
scat_plot(traj, periodic = FALSE)
```

Arguments

traj	data.frame containing the trajectory produced by e.g. make_traj(). It must contain the columns traj$angle and traj$steplength.
periodic	logical value denoting whether the plot should be periodically extended past -pi and pi.

Value

A 'ggplot' object, the scatterplot.

References

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

See Also

cop_scat_plot(), traj_plot(), circ_plot(), cop_plot().

Examples

```
set.seed(123)

traj <- make_traj(100,
  copula = cyl_quadsec(0.1),
  marginal_circ = "vonmises",
  parameter_circ = list(0, 1),
  marginal_lin = "weibull",
  parameter_lin = list(shape=3)
)
plot1 <- scat_plot(traj)
plot2 <- scat_plot(traj, periodic = TRUE)
```

---

setCopParam *Change Attributes of 'cyl_copula' Objects*

Description

These methods can be used, e.g. in other functions, to give users limited access to the parameters of a copula.

Usage

```
setCopParam(copula, param_val, param_name = NULL, ...)

## S4 method for signature 'cyl_cubsec'
setCopParam(copula, param_val, param_name)

## S4 method for signature 'cyl_quadsec'
setCopParam(copula, param_val, param_name)

## S4 method for signature 'cyl_rect_combine'
setCopParam(copula, param_val, param_name)

## S4 method for signature 'cyl_rot_combine'
setCopParam(copula, param_val, param_name)

## S4 method for signature 'cyl_vonmises'
setCopParam(copula, param_val, param_name)
```

Arguments

copula	R object of class 'cyl_copula'.
param_val	numeric vector holding the values to which the parameters given in copula@parameters should be changed.
param_name	vector of character strings holding the names of the parameters to be changed.
...	additional arguments.

Details

Note that for a rectangular patchwork copula ('cyl_rect_combine') the attribute rectangles_symmetric cannot be changed by setCopParam(), since rectangular patchwork copulas with symmetric rectangles are treated as distinct from rectangular patchwork copulas with potentially asymmetric rectangles. Therefore, when changing one of the bounds of the lower rectangle of such a copula, the corresponding bound of the upper rectangle is automatically changed as well (see examples).

Value

A 'cyl_copula' object with the changed parameters.

Examples

```
cop <- cyl_rect_combine(copula::normalCopula(0.2),low_rect = c(0.1,0.4), up_rect="symmetric")
cop
cop <- setCopParam(cop, param_val = c(0.1, 0.3), param_name = c("rho.1", "low_rect2"))
cop <- cyl_rect_combine(copula::normalCopula(0.2),low_rect = c(0.1,0.4), up_rect=c(0.6,0.9))
cop
cop <- setCopParam(cop, param_val = 0.3, param_name = "low_rect2")
cop
```

---

show,cyl_copula-method

Print Information of 'cyl_copula' Objects

Description

Methods for function show() in package **cylcop**

Usage

```
## S4 method for signature 'cyl_copula'
show(object)

## S4 method for signature 'cyl_rect_combine'
show(object)

## S4 method for signature 'cyl_rot_combine'
show(object)
```

Arguments

object	R object of class 'cyl_copula'.

Value

An invisible NULL. As side effect, information on object is printed.

traj_plot *Plot a Trajectory in Euclidean Space*

Description

This function plots the locations of a trajectory. The first and last point are marked in red.

Usage

```
traj_plot(traj)
```

Arguments

traj	data.frame containing the trajectory produced by e.g. make_traj(). It must contain the columns traj$pos_x and traj$pos_y.

Value

A 'ggplot' object.

References

Hodel FH, Fieberg JR (2021). "Circular-Linear Copulae for Animal Movement Data." *bioRxiv*. doi: 10.1101/2021.07.14.452404, https://www.biorxiv.org/content/10.1101/2021.07.14.452404v1/.

Hodel FH, Fieberg JR (2021). "Cylcop: An R Package for Circular-Linear Copulae with Angular Symmetry." *bioRxiv*. doi: 10.1101/2021.07.14.452253, https://www.biorxiv.org/content/10.1101/2021.07.14.452253v1/.

See Also

cop_scat_plot(), circ_plot(), cop_plot(), scat_plot().

Examples

```
set.seed(123)

traj <- make_traj(50,
  copula = cyl_quadsec(0.1),
  marginal_circ = "vonmises",
  parameter_circ = list(0, 1),
  marginal_lin = "weibull",
  parameter_lin = list(shape=3)
)
plot1 <- traj_plot(traj)
```

wrappedcauchy *Density, Distribution, Quantiles and Random Number Generation for the Wrapped Cauchy Distribution*

Description

The distribution function `pwrappedcauchy()` and quantiles `qwrappedcauchy()`of the wrapped Cauchy distribution can not be obtained analytically. They are therefore missing in the '**circular**' package and are obtained here numerically. Random number generation `rwrappedcauchy()` and density `dwrappedcauchy()` don't need a numerical approximation and are provided here just for consistency in parametrization with the other wrapped Cauchy functions. One could also convert `scale` to `rho` (`rho = exp(-scale)`) and use `circular::rwrappedcauchy(rho)`. In fact, for the density, one usually SHOULD convert `scale` to `rho` (`rho = exp(-scale)`) and use `circular::dwrappedcauchy(rho)` which does not make a numerical approximation and is therefore faster than `cylcop::dwrappedcauchy()`.

Usage

```
rwrappedcauchy(n, location = 0, scale = 1)

dwrappedcauchy(theta, location = 0, scale = 1, K = 100, check_prec = FALSE)

pwrappedcauchy(theta, location = 0, scale = 1, K = 100, check_prec = FALSE)

qwrappedcauchy(p, location = 0, scale = 1, K = 100, check_prec = FALSE)
```

Arguments

`n`	integer value, the number of random samples to be generated with `rwrappedcauchy()`.
`location`	numeric value, the mean of the distribution.
`scale`	numeric value, the parameter tuning the spread of the density.
`theta`	numeric vector giving the angles where the density or distribution function is evaluated.
`K`	integer value, the number of "wraps" used in each direction to approximate the distribution.
`check_prec`	logical, whether to check if the precision of the numerical approximation with the current parameters is higher than 99%.
`p`	numeric vector giving the probabilities where the quantile function is evaluated.

Details

The density is calculated by wrapping the Cauchy distribution K times around the circle in each direction and summing the density at each point of the circle. E.g. the density of the wrapped Cauchy distribution at angle θ is calculated as the sum of the Cauchy density at $\theta + 2\pi k$, where the integer k goes from $-K$ to K. The distribution function is obtained similarly and the quantiles are calculated by numerical inversion.

Value

`dwrappedcauchy()` and `pwrappedcauchy()` give a vector of length `length(theta)` containing the density or distribution function at the corresponding values of `theta`. `qwrappedcauchy()` gives a vector of length `length(p)` containing the quantiles at the corresponding values of `p`. `rwrappedcauchy()` generates a vector of length `n` containing the random samples, i.e. angles in $[-\pi, \pi)$.

See Also

`circular::dwrappedcauchy()`, `circular::rwrappedcauchy()`.

Examples

```
set.seed(123)

rwrappedcauchy(10, location = 0, scale =3)

dwrappedcauchy(c(0.1, pi), location = pi, scale =2)
circular::dwrappedcauchy(circular::circular(c(0.1,pi)), mu = circular::circular(pi), rho =exp(-2))

prob <- pwrappedcauchy(c(0.1, pi), location = pi, scale =2)
prob
qwrappedcauchy(prob, location = pi, scale =2)
```

Index