Package: bpDir (via r-universe)

November 9, 2026

Type Package

Title Boxplots for Directional Data

Version 0.2.6

Description Functions for drawing boxplots for data on (the boundary of) a unit circle (i.e., circular and axial data), from Buttarazzi D., Pandolfo G., Porzio G.C. (2018) <doi:10.1111/biom.12889>. The package allows producing also circular violin plots, and includes methods to calculate robust measures of dispersion for circular data.

License GPL-2

Depends R (>= 3.6.0)

Imports circular, plotrix, graphics, stats, ggplot2, dplyr

Encoding UTF-8

Date 2026-11-08

RoxygenNote 7.3.3

NeedsCompilation no

Author Davide Buttarazzi [aut, cre]

Maintainer Davide Buttarazzi <davidebuttarazzi90@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-11-09 17:40:24 UTC

RemoteUrl https://github.com/cran/bpDir

RemoteRef HEAD

RemoteSha 9dccf758b5ecc0c7334d9e8299d5c41c3c7cbd8f

Contents

AxialBoxplot 2
CircularBoxplot 4
CircularLMS 5
CircularLTS 6

CircularMAD 7
CircularTukeyDepth 8
CircularViolinPlot 9
e_healthy 13
e_parkinson 14
EstimateKappa 15
EstimateKappaKo 16
EstimateSigma 17
generate_results_grid_sigma 18
generate_vonmises_grid 19
Spring 20
Summer 21

Index **23**

`AxialBoxplot` *AxialBoxplot*

Description

Draw a boxplot for axial data

Usage

```
AxialBoxplot(A, template = "degrees", place = "none", marg =
                    "large", stack = FALSE, H = FALSE, shrink = 1.5, units
             = "degrees", constant = "optimal", concentration = NULL, mirror = TRUE)
```

Arguments

`A`	numeric; a vector of class circular in `units=c("degrees", "radians")`
`template`	character; graphical parameter to set the template to be used in the plot. Options are `"degrees"`(default), `"radians"` and `"geographics"`. If `NULL`, a generic template can be defined, with angles displayed according to the graphical parameter `place`.
`place`	character; graphical parameter. Active only when `template = NULL` to add an additional generic template either `"outside"` or `"inside"` the circle. If `place = "none"`(default), no additional template is plotted.
`marg`	character; graphical parameter to set the plot region to either `"large"`(default) or `"small"`, expressed in number of lines of text.
`stack`	If `TRUE` and `H=FALSE`, then far out values are stacked on the circumference.
`H`	logical; graphical parameter to specify if observations between the hinges and the whiskers are plotted. Default is `FALSE`.
`shrink`	numeric; graphical parameter to control the size of the plotted circle. Default is 1.5. Larger values shrink the circle, while smaller values enlarge the circle.

units	character; graphical parameter to select the unit of measure to be displayed on the axes when template = NULL and place=c("inside","outside"). Options are "degrees"(default) or "radians".
constant	numeric; the boxplot multiplying factor (or "constant") that determines how far the fences extend out from the box. When "optimal"(default), it ensures the probability that an observation falls outside the fences (i.e., it will be a far out value) is 0,7% under a von Mises distribution (see Web Appendix A in Buttarazzi et al., 2018).
concentration	numeric; if constant = "optimal", this argument can be used to specify the concentration parameter to be used to estimate the boxplot multiplying factor.
mirror	If TRUE, then a second axial boxplot will be plotted on the opposite semi-circle.

Details

The arguments stack, shrink are parameters passed to plot.circular.

Value

farout	a vector of class circular in radians, containing the values of the points lying outside the whiskers.
constant	the value of the constant used to draw the boxplot.

Author(s)

Davide Buttarazzi

References

Buttarazzi D., Pandolfo G., Porzio G.C. (2018). A boxplot for circular data, Biometrics.

Examples

```
set.seed(1)
#install.packages("circular")
require(circular)
#install.packages("plotrix")
require(plotrix)
theta <- circular::rvonmises(100, circular(pi/2), 3, control.circular=list(units="radians"))
axialTheta <- circular(theta, modulo = "pi")
AxialBoxplot(axialTheta, template = "radians", mirror = FALSE)
```

CircularBoxplot *Circular Box-and-Wisker Plot*

Description

The CircularBoxplot function produces a box-and-whisker-plot for circular data.

Usage

```
CircularBoxplot ( A, template="degrees", place="none", units="degrees", marg=
                  "large", shrink = 1.5,  H=FALSE, stack=FALSE, constant=
                  "optimal", concentration= NULL)
```

Arguments

A	numeric; a vector of class circular in units=c("degrees", "radians")
template	character; graphical parameter to set the template to be used in the plot. Options are "degrees"(default), "radians" and "geographics". If NULL, a generic template can be defined, with angles displayed according to the graphical parameter place.
place	character; graphical parameter. Active only when template = NULL to add an additional generic template either "outside" or "inside" the circle. If place = "none"(default), no additional template is plotted.
units	character; graphical parameter to select the unit of measure to be displayed on the axes when template = NULL and place=c("inside","outside"). Options are "degrees"(default) or "radians".
marg	character; graphical parameter to set the plot region to either "large"(default) or "small", expressed in number of lines of text.
shrink	numeric; graphical parameter to control the size of the plotted circle. Default is 1.5. Larger values shrink the circle, while smaller values enlarge the circle.
H	logical; graphical parameter to specify if observations between the hinges and the whiskers are plotted. Default is FALSE.
stack	If TRUE and H=FALSE, then far out values are stacked on the circumference.
constant	numeric; the boxplot multiplying factor (or "constant") that determines how far the fences extend out from the box. When "optimal"(default), it ensures the probability that an observation falls outside the fences (i.e., it will be a far out value) is 0,7% under a von Mises distribution (see Web Appendix A in Buttarazzi et al., 2018).
concentration	numeric; if constant = "optimal", this argument can be used to specify the concentration parameter to be used to estimate the boxplot multiplying factor.

Details

The arguments stack, shrink are parameters passed to plot.circular.

Value

farout	a vector of class circular in radians, containing the values of the points lying outside the whiskers.
statistics	a dataframe of summary statistics in degrees: circular median, hinges and whiskers.
constant	the value of the constant used to draw the boxplot.

Author(s)

Davide Buttarazzi

References

Buttarazzi D., Pandolfo G., Porzio G.C. (2018). A boxplot for circular data, Biometrics.

Examples

```
# Circular Boxplot on Vanishing directions of homing pigeons
#install.packages("circular")
#install.packages("plotrix")
require(circular)
require(plotrix)
#help(fisherB12c)
data(fisherB12c)
CircularBoxplot(fisherB12c, template="geographics")
```

CircularLMS *Compute the robust measure of circular spread: the Circular Least Median Spread (CLMS)*

Description

Compute the Circular Least Median Spread (Demni et al, 2026)

Usage

```
CircularLMS(X)
```

Arguments

X	numeric; a vector of class circular in units=c("radians")

Value

Circular Least Median Spread (CLMS)

Author(s)

Davide Buttarazzi, Houyem Demni

References

Demni et al. (2026), Robust measures of dispersion for circular data with an anomaly detection rule. Under review.

See Also

CircularViolinPlot

Examples

```
#install.packages("circular")
require(circular)
data1 <- rvonmises(50, circular(pi/2), 5)
CircularLMS(data1)
```

`CircularLTS` *Compute the robust measure of circular spread: the Circular Least Trimmed Standard deviation (CLTS)*

Description

Compute the Circular Least Trimmed Standard deviation (Demni et al, 2026)

Usage

```
CircularLTS(X)
```

Arguments

`X`	numeric; a vector of class `circular` in `units=c("radians")`

Value

Circular Least Trimmed Standard deviation (CLTS)

Author(s)

Davide Buttarazzi, Houyem Demni

References

Demni et al. (2026), Robust measures of dispersion for circular data with an anomaly detection rule. Under review.

See Also

CircularViolinPlot

Examples

```
#install.packages("circular")
require(circular)
data1 <- rvonmises(50, circular(pi/2), 5)
CircularLTS(data1)
```

`CircularMAD` *Compute the robust measure of circular spread: the Circular Median Absolute Deviation (CMAD)*

Description

Compute the Circular Median Absolute Deviation (Demni et al, 2026)

Usage

```
CircularMAD(X, h = FALSE)
```

Arguments

`X`	numeric; a vector of class `circular` in `units=c("radians")`
`h`	logical; optionally produce histogram of the distances between the median and X, set to FALSE by default.

Value

Circular Median Absolute Deviation (CMAD)

Author(s)

Davide Buttarazzi, Houyem Demni

References

Demni et al. (2026), Robust measures of dispersion for circular data with an anomaly detection rule. Under review.

See Also

CircularViolinPlot

Examples

```
#install.packages("circular")
require(circular)
data1 <- rvonmises(50, circular(pi/2), 5)
CircularMAD(data1)
```

`CircularTukeyDepth` *Circular Tukey Depth*

Description

Empirical depth ranking based on Tukey's ranking procedure (Tukey, 1977) adapted to the circular setting (Buttarazzi et al, 2018).

Usage

```
CircularTukeyDepth(x)
```

Arguments

`x`	numeric; a vector of class `circular` in `units=c("degrees", "radians")`

Value

`depth`	a two column matrix of circular observations with corresponding depth value
`median`	the depth-based circular median as an object of class `circular`
`iqr`	the depth-based circular inter-quartile range as an object of class `circular`

Author(s)

Davide Buttarazzi

References

Buttarazzi D., Pandolfo G., Porzio G.C. (2018). A boxplot for circular data, Biometrics.

Tukey, J. W. (1977). Exploratory Data Analysis. MA: AddisonWesley, Reading (UK).

See Also

CircularBoxplot

Examples

```
#install.packages("circular")
require(circular)
data1 <- rvonmises(50, circular(pi/2), 5)
CircularTukeyDepth(data1)
```

CircularViolinPlot *Circular Violin Plot*

Description

The CircularViolinPlot function produces a violin plot for circular data, with support for full, outer, inner, and two-halves representations. It provides options for displaying confidence intervals, notches, outliers, and additional graphical elements.

Usage

```
CircularViolinPlot(
    data,
    data2 = NULL,
    type = c("full", "outer", "inner", "two_halves"),
    notch = FALSE,
    beans = c("none", "all", "inliers", "outliers", "outlier_points"),
    outlier_point_color = "red",
    outlier_point_size = 2,
    outlier_point_alpha = 0.8,
    show_ci_interval = TRUE,
    show_notch_cutout = TRUE,
    show_notch_guides = FALSE,
    r0 = 1,
    density_scale = 0.3,
    vonMises_grid = NULL,
    bw = NULL,
    bw2 = NULL,
    kernel = "vonmises",
    fill = "skyblue",
    fill2 = "darkorange",
    alpha = 0.75,
    alpha2 = 0.75,
    fill_outlier = NULL,
    fill2_outlier = NULL,
    alpha_outlier = 0.2,
    alpha2_outlier = 0.2,
    arc_color = "black",
    arc_color2 = "black",
    ci_color = "white",
    ci_color2 = "white",
    arc_thickness = 0.06,
    ci_thickness = 0.012,
    box_width_factor = 1,
    density_threshold = 0,
    cutoff = NULL,
    cutoff2 = NULL,
```

```
    n_boot = 1000,
    ci_level = 0.95,
    notch_factor = 0.9,
    notch_guide_linetype = "solid",
    notch_guide_linewidth = 0.4,
    bean_length = 0.035,
    bean_min_size = 0.15,
    bean_max_size = 0.6,
    circle_color = "grey70",
    circle_linewidth = 0.7,
    center_point = TRUE,
    center_point_color = "black",
    center_point_size = 1.8,
    title = "",
    xlim = c(-1.5, 1.5),
    ylim = c(-1.5, 1.5)
)
```

Arguments

`data`	Numeric vector of class `circular`, containing the primary circular observations. The data should have units specified as degrees or radians.
`data2`	Optional numeric vector of class `circular`, containing a second set of circular observations. Required when `type = "two_halves"`.
`type`	Character string specifying the plot representation. Options are `"full"` (default), `"outer"`, `"inner"`, and `"two_halves"`.
`notch`	Logical; whether to display notches in the plot. Default is `FALSE`.
`beans`	Character string specifying which observations or groups should be represented by beans. Options are `"none"`, `"all"`, `"inliers"`, `"outliers"`, and `"outlier_points"`.
`outlier_point_color`	Character string specifying the color of individual outlier points. Default is `"red"`.
`outlier_point_size`	Numeric; size of individual outlier points. Default is 2.
`outlier_point_alpha`	Numeric; transparency of individual outlier points, between 0 and 1. Default is 0.8.
`show_ci_interval`	Logical; whether to display the confidence interval. Default is `TRUE`.
`show_notch_cutout`	Logical; whether to display the notch cutout. Default is `TRUE`.
`show_notch_guides`	Logical; whether to display guide lines associated with the notches. Default is `FALSE`.
`r0`	Numeric; radius of the reference circle. Default is 1.
`density_scale`	Numeric; scaling factor applied to the normalized density to determine the width of the violin. Default is 0.3.

vonMises_grid An optional data frame containing a precomputed calibration grid generated by generate_vonmises_grid. If supplied, this grid is used for estimating the von Mises concentration parameter without regenerating the calibration grid. If NULL (default), the grid is generated automatically when required.

bw Numeric; smoothing bandwidth used to estimate the density of data. By default, when kernel is "vonmises", the bandwidth is determined automatically.

bw2 Numeric; smoothing bandwidth used to estimate the density of data2. Default is NULL, allowing automatic selection.

kernel Character string specifying the smoothing kernel used for density estimation. Supported options are "vonmises" and "wrappednormal", corresponding to kernels of type "K" in density.circular from the **circular** package.

fill Character string specifying the fill color of the primary violin. Default is "skyblue".

fill2 Character string specifying the fill color of the second violin or half. Default is "darkorange".

alpha Numeric; transparency of the primary violin fill, between 0 and 1. Default is 0.75.

alpha2 Numeric; transparency of the second violin fill, between 0 and 1. Default is 0.75.

fill_outlier Character string specifying the fill color used for outliers in the primary data. Default is NULL.

fill2_outlier Character string specifying the fill color used for outliers in the second data set. Default is NULL.

alpha_outlier Numeric; transparency of the primary outlier fill, between 0 and 1. Default is 0.2.

alpha2_outlier Numeric; transparency of the second outlier fill, between 0 and 1. Default is 0.2.

arc_color Character string specifying the color of the arcs of the primary violin. Default is "black".

arc_color2 Character string specifying the color of the arcs of the second violin or half. Default is "black".

ci_color Character string specifying the color used to display the confidence interval for the primary data. Default is "white".

ci_color2 Character string specifying the color used to display the confidence interval for the second data set. Default is "white".

arc_thickness Numeric; thickness of the violin arcs. Default is 0.06.

ci_thickness Numeric; thickness of the confidence interval elements. Default is 0.012.

box_width_factor
Numeric; scaling factor controlling the width of the box or summary-statistic component. Default is 1.

density_threshold
Numeric; density threshold below which density values are set to zero, resulting in a narrower violin. Default is zero, meaning no additional shrinking.

cutoff Numeric; cutoff threshold for the outlier detection rule applied to the primary data, used to clip the density. Default is NULL.

`cutoff2` Numeric; cutoff threshold for the outlier detection rule applied to the second data set. Default is `NULL`.

`n_boot` Integer; number of bootstrap iterations used to estimate confidence intervals. Default is 1000.

`ci_level` Numeric; confidence level used for confidence interval estimation. Default is 0.95.

`notch_factor` Numeric; scaling factor controlling the size of the notches. Default is 0.9.

`notch_guide_linetype` Character string specifying the line type of the notch guide lines. Default is `"solid"`.

`notch_guide_linewidth` Numeric; line width of the notch guide lines. Default is 0.4.

`bean_length` Numeric; parameter controlling the length of the bean elements. Default is 0.035.

`bean_min_size` Numeric; minimum size of the bean elements. Default is 0.15.

`bean_max_size` Numeric; maximum size of the bean elements. Default is 0.6.

`circle_color` Character string specifying the color of the reference circle. Default is `"grey70"`.

`circle_linewidth` Numeric; line width of the reference circle. Default is 0.7.

`center_point` Logical; whether to display the center point. Default is `TRUE`.

`center_point_color` Character string specifying the color of the center point. Default is `"black"`.

`center_point_size` Numeric; size of the center point. Default is 1.8.

`title` Character string specifying the plot title. Default is an empty string.

`xlim` Numeric vector of length two specifying the horizontal axis limits. Default is `c(-1.5, 1.5)`.

`ylim` Numeric vector of length two specifying the vertical axis limits. Default is `c(-1.5, 1.5)`.

Value

outliers A vector of class `circular` in radians, containing observations classified as outliers as lying in the outlying region.

cutoff The cutoff threshold used by the outlier detection rule to clip the density. The violin plot is truncated at the furthest non-outliers.

statistics A data frame containing circular summary statistics including the circular median, first and third quartiles.

Author(s)

Davide Buttarazzi, Houyem Demni Demni et al. (2026), Robust measures of dispersion for circular data with an anomaly detection rule. Under review. Demni et al. (2026), Notched violin plots for circular data. In: Martella, F., Arima, S., Marino, M.F., Mollica, C. (eds) Statistical Science: From Theory to Applied Research I. SIS-FENStatS 2026. Italian Statistical Society Series on Advances in Statistics. Springer, Cham. https://doi.org/10.1007/978-3-032-30873-3 10.

See Also

density.circular

Examples

```
require(circular)
x <- rvonmises(50, circular(pi/2), 5)
v = generate_vonmises_grid(rho_seq = seq(0.01, 0.99, by = 0.05))
CircularViolinPlot(x, vonMises_grid = v)

# Specify a plot representation explicitly
CircularViolinPlot(x, vonMises_grid = v, type = "full")

# Example with two data sets
y <- rvonmises(50, circular(pi), 5)
CircularViolinPlot(x, data2 = y, vonMises_grid = v, type = "two_halves")
```

---

e_healthy *Elbow angles for a healthy patient*

Description

Angular gait measurements for a healthy patient, recorded as left-elbow angles.

Usage

```
data("GaitData")
```

Format

A matrix of class `circular`, with angles in radians. The supplied object includes a column label `left_elbow_angle_healthy`.

Details

This dataset was supplied in the file '`GaitData.RData`'. The observations use the `circular` class from the **circular** package.

Source

Pace et al. 2026, Markerless gait analysis and circular statistics: a case study in Parkinson's disease classification. In: Sintorn, I.M., Pal, U., Lu, S. (eds), Pattern Recognition. ICPR 2026 International Workshops. Lecture Notes in Computer Science, vol. 17110. Springer, Cham. DOI: 10.1007/978-3-032-39379-1 4.

Kour, N., Arora, S.: Vision-based gait dataset for gait analysis. In: 2022 2nd Inter-national Conference on Advance Computing andInnovativeTechnologiesinEngi-neering,pp.2163-2168.IEEE(2022)

See Also

circular

Examples

```
data("GaitData")
str(e_healthy)
summary(e_healthy)
```

---

e_parkinson *Elbow angles for a Parkinson patient*

Description

Angular gait measurements for a Parkinson parkinson, recorded as left-elbow angles.

Usage

```
data("GaitData")
```

Format

A matrix of class `circular`, with angles in radians. The supplied object includes a column label `left_elbow_angle_parkinson`.

Details

This dataset was supplied in the file '`GaitData.RData`'. The observations use the `circular` class from the **circular** package.

Source

Pace et al. 2026, Markerless gait analysis and circular statistics: a case study in Parkinson's disease classification. In: Sintorn, I.M., Pal, U., Lu, S. (eds), Pattern Recognition. ICPR 2026 International Workshops. Lecture Notes in Computer Science, vol. 17110. Springer, Cham. DOI: 10.1007/978-3-032-39379-1 4.

Kour, N., Arora, S.: Vision-based gait dataset for gait analysis. In: 2022 2nd Inter-national Conference on Advance Computing andInnovativeTechnologiesinEngi-neering,pp.2163-2168.IEEE(2022)

See Also

circular

Examples

```
data("GaitData")
str(e_parkinson)
summary(e_parkinson)
```

`EstimateKappa` *Robust estimation of the concentration parameter kappa of the von Mises distribution from robust measures of dispersion*

Description

Robust estimation of the concentration parameter of the von Mises distribution from robust measures of dispersion (Demni et al, 2026)

Usage

```
EstimateKappa(X, stat = c("CLMS","CMAD", "CLTS"), results_grid = NULL)
```

Arguments

`X`	numeric; a vector of class `circular`
`stat`	character; one of the available estimators `c("CLMS","CMAD", "CLTS")`, respectively Circular Median Absolute Deviation, Circular Least Median Spread, Circular Least Trimmed Standard deviation.
`results_grid`	a dataframe of a custom grid of kappa used for interpolation. If NULL (default), the grid is handled automatically.

Value

`CSD`	the estimated circular standard deviation based on the robust measures of dispersion
`kappa`	the estimated kappa based on the estimated robust circular standard deviation

Author(s)

Davide Buttarazzi, Houyem Demni

References

Demni et al. (2026), Robust measures of dispersion for circular data with an anomaly detection rule. Under review.

See Also

CircularViolinPlot

Examples

```
#install.packages("circular")
require(circular)
data1 <- rvonmises(50, circular(pi/2), 5)
EstimateKappa(data1)
```

`EstimateKappaKo` *Robust estimation of the concentration parameter of the von Mises-Fisher distribution*

Description

Robust estimation of the concentration parameter of the von Mises-Fisher distribution (Ko, 1992).

Usage

```
EstimateKappaKo(X)
```

Arguments

`X`	numeric; a vector of class `circular` in `units=c("radians")`

Value

A numeric value

Author(s)

Davide Buttarazzi

References

Ko, D. (1992), Robust estimation of the concentration parameter of the von Mises-Fisher distribution. Ann. Stat. 20(2) 917-928

See Also

CircularBoxplot

Examples

```
#install.packages("circular")
require(circular)
data1 <- rvonmises(50, circular(pi/2), 5)
EstimateKappaKo(data1)
```

`EstimateSigma` *Robust estimation of the scale parameter sigma of the Wrapped-Normal distribution from the robust measures of dispersion*

Description

Robust estimation of the scale parameter sigma of the Wrapped-Normal distribution from the robust measures of dispersion (Demni et al, 2026)

Usage

```
EstimateSigma(X, stat = c("CLMS", "CMAD", "CLTS"), results_grid = NULL)
```

Arguments

`X`	numeric; a vector of class `circular`
`stat`	character; one of the available estimators `c("CLMS", "CMAD", "CLTS")`, respectively Circular Median Absolute deviation, Circular Least Median Spread, Circular Least Trimmed Standard deviation.
`results_grid`	a dataframe of a custom grid of kappa used for interpolation. If NULL (default), the grid is handled automatically.

Value

`CSD`	the estimated circular standard deviation based on the robust measures of dispersion
`sigma`	the estimated sigma based on the estimated robust circular standard deviation

Author(s)

Davide Buttarazzi, Houyem Demni

References

Demni et al. (2026), Robust measures of dispersion for circular data with an anomaly detection rule. Under review.

See Also

CircularViolinPlot

Examples

```
#install.packages("circular")
require(circular)
data1 <- rvonmises(25, circular(pi/2), 5)
g=generate_results_grid_sigma(sigma_seq = seq(0.1, 1.5, by = 0.1))
EstimateSigma(data1, results_grid = g)
```

`generate_results_grid_sigma` *Generate a Calibration Grid for Estimating the Wrapped Normal Scale Parameter*

Description

Generates a calibration grid relating robust measures of circular dispersion to the circular standard deviation (CSD) and the scale parameter (σ) of a wrapped normal distribution.

The grid is used by the estimation procedure to convert an observed robust circular dispersion statistic into an estimate of CSD through monotone spline interpolation, and subsequently into an estimate of the wrapped normal scale parameter.

Usage

```
generate_results_grid_sigma(
  sigma_seq = seq(0.1, 1.5, by = 0.0005)
)
```

Arguments

`sigma_seq`	A numeric vector of positive values of the wrapped normal scale parameter σ for which the calibration grid is generated. The default sequence ranges from 0.1 to 1.5 in increments of 0.0005.

Details

The calibration grid is constructed by evaluating the dispersion statistics over a sequence of wrapped normal distributions with different values of σ.

For a given observed dataset, the estimation procedure first computes one of the following robust measures of circular dispersion: `CircularMAD`, `CircularLMS`, or `CircularLTS`, depending on the selected statistic.

A precomputed calibration grid can be supplied to the estimation function through its `results_grid` argument, avoiding the need to regenerate the grid for every estimation.

Value

A data frame with the following columns:

sigma Values of the wrapped normal scale parameter.

rho The mean resultant length of the wrapped normal distribution, calculated as $\exp(-\sigma^2/2)$.

CSD The circular standard deviation, calculated as $\sqrt{2(1-\rho)}$.

CMAD The calibration values obtained using the 0.75 quantile of the centred wrapped normal distribution, for the CMAD-based estimation procedure.

CLMS The calibration values obtained using the 0.75 quantile of the centred wrapped normal distribution, for the CLMS-based estimation procedure.

CLTS The calibration values obtained using the CLTS-based estimation procedure.

See Also

CircularMAD CircularLMS CircularLTS

Examples

```
# Generate a smaller calibration grid
grid <- generate_results_grid_sigma(
  sigma_seq = seq(0.1, 1.5, length.out = 50)
)

head(grid)
```

`generate_vonmises_grid` *Generate a Calibration Grid for the von Mises Concentration Parameter*

Description

Generates a calibration grid relating robust measures of circular dispersion to the circular standard deviation (CSD) and the concentration parameter (κ) of a von Mises distribution.

The grid is used by the estimation procedure to convert an observed robust circular dispersion statistic into an estimate of CSD through monotone spline interpolation, and subsequently into an estimate of the von Mises concentration parameter.

Usage

```
generate_vonmises_grid(
  rho_seq = seq(0.01, 0.99, by = 0.002)
)
```

Arguments

`rho_seq`	A numeric vector of values of the mean resultant length ρ for which the calibration grid is generated. Values must lie strictly between 0 and 1. The default sequence ranges from 0.01 to 0.99 in increments of 0.002.

Details

The calibration grid is constructed by evaluating the dispersion statistics over a sequence of von Mises distributions with different values of the concentration parameter κ.

A precomputed calibration grid can be supplied to the estimation function through its `results_grid` argument, avoiding the need to regenerate the grid for every estimation.

Value

A data frame with the following columns:

kappa Values of the von Mises concentration parameter, calculated by applying `circular::A1inv` to the corresponding values of ρ.

rho Values of the mean resultant length.

CSD The circular standard deviation, calculated as $2(1-\rho)$.

CMAD Calibration values obtained using the 0.75 quantile of the centred von Mises distribution, for the CMAD-based estimation procedure.

CLMS Calibration values obtained using the 0.75 quantile of the centred von Mises distribution, for the CLMS-based estimation procedure.

CLTS Calibration values computed using the internal procedure for the von Mises distribution, for the CLTS-based estimation procedure.

See Also

`CircularMAD` `CircularLMS` `CircularLTS`

Examples

```
# Generate a custom calibration grid
custom_grid <- generate_vonmises_grid(
  rho_seq = seq(0.05, 0.95, length.out = 50)
)

head(custom_grid)
```

Spring *Spring wind directions*

Description

Wind-direction measurements for spring, stored as circular observations.

Usage

```
data("Wind_directions")
```

Format

A matrix of class `circular`, with angles in radians. The supplied object includes a column label `Spring2010`.

Details

This dataset was supplied in the file '`Wind_directions.RData`'. The observations use the `circular` class from the **circular** package.

Source

Demni et al. (2026), Notched violin plots for circular data. In: Martella, F., Arima, S., Marino, M.F., Mollica, C. (eds) Statistical Science: From Theory to Applied Research I. SIS-FENStatS 2026. Italian Statistical Society Series on Advances in Statistics. Springer, Cham. https://doi.org/10.1007/978-3-032-30873-3 10.

See Also

circular

Examples

```
data("Wind_directions")
str(Spring)
summary(Spring)
```

Summer *Summer wind directions*

Description

Wind-direction measurements for summer, stored as circular observations.

Usage

```
data("Wind_directions")
```

Format

A matrix of class `circular`, with angles in radians. The supplied object includes a column label `Summer2010`.

Details

This dataset was supplied in the file '`Wind_directions.RData`'. The observations use the `circular` class from the **circular** package.

Source

Demni et al. (2026), Notched violin plots for circular data. In: Martella, F., Arima, S., Marino, M.F., Mollica, C. (eds) Statistical Science: From Theory to Applied Research I. SIS-FENStatS 2026. Italian Statistical Society Series on Advances in Statistics. Springer, Cham. https://doi.org/10.1007/978-3-032-30873-3 10.

See Also

circular

Examples

```
data("Wind_directions")
str(Summer)
summary(Summer)
```

Index

∗ **circular**
  CircularLMS, 5
  CircularLTS, 6
  CircularMAD, 7
  CircularViolinPlot, 9
  EstimateKappa, 15
  EstimateKappaKo, 16
  EstimateSigma, 17
∗ **datasets**
  e_healthy, 13
  e_parkinson, 14
  Spring, 20
  Summer, 21
∗ **depth**
  CircularTukeyDepth, 8
∗ **device**
  CircularBoxplot, 4

AxialBoxplot, 2

circular, *14*, *21*
CircularBoxplot, 4
CircularLMS, 5, *19*, *20*
CircularLTS, 6, *19*, *20*
CircularMAD, 7, *19*, *20*
CircularTukeyDepth, 8
CircularViolinPlot, 9

density.circular, *13*

e_healthy, 13
e_parkinson, 14
EstimateKappa, 15
EstimateKappaKo, 16
EstimateSigma, 17

generate_results_grid_sigma, 18
generate_vonmises_grid, *11*, 19

Spring, 20
Summer, 21