

Package: CircularRegression (via r-universe)

June 11, 2026

Type Package

Title Circular Regression Models

Version 0.5.0

Description Implements regression models for circular response data, including homogeneous angular regression, consensus angular regression, a two-step workflow, selected special-case model wrappers, and a random-intercept extension for clustered circular outcomes.

License GPL-3

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.3

Imports circular, graphics, stats, utils

Depends R (>= 3.5.0)

Suggests ggplot2, gridExtra, knitr, pkgdown, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

URL <https://github.com/AurelienNicosiaULaval/CircularRegression>

BugReports <https://github.com/AurelienNicosiaULaval/CircularRegression/issues>

Config/testthat/edition 3

Repository <https://aureliennicosiaulaval.r-universe.dev>

Date/Publication 2026-05-31 18:19:42 UTC

RemoteUrl <https://github.com/aureliennicosiaulaval/circularregression>

RemoteRef HEAD

RemoteSha 864325dbba13d418b596697415e8986aefc5e901

Contents

angular	2
angular_lrtest	3
angular_re	4
angular_two_step	5
autoregressivedata	6
bison	7
circular_regression	8
consensus	9
decentredPredictorModel	10
jammalamadakaModel	11
meanDirectionModel	12
multiplebison	13
noshiro	14
pick_reference_angle	15
plot.angular_re	16
plot_ranef.angular_re	17
predict.angular_re	17
presnellModel	18
print.angular	19
print.angular_re	20
print.angular_two_step	21
print.circular_regression	22
print.consensus	23
print.pick_reference_angle	25
ranef.angular_re	25
Sandhopper	26
select_reference_angle	27
Index	28

 angular

Angular Regression Model with Homogeneous von Mises Errors

Description

Fits the homogeneous angular regression model of Rivest et al. (2016), where the mean direction is the orientation of a weighted resultant vector and one reference angle has coefficient fixed to 1.

Usage

```
angular(
  formula,
  data,
  reference = c("auto", "first", "name"),
  initbeta = NULL,
  control = list(),
```

```

    na.action = stats::na.omit
  )

```

Arguments

formula	A formula with terms of the form x or $x:z$.
data	A data frame, list or environment containing model variables.
reference	Reference-angle rule. Use "auto" (default, article criterion), "first", or <code>c("name", "x_ref")</code> . A plain angle term <code>x_ref</code> must exist in the formula.
initbeta	Optional numeric vector of initial values for non-reference coefficients.
control	Control list with optional components <code>pginit</code> , <code>maxiter</code> , and <code>mindiff</code> .
na.action	Function used by <code>model.frame</code> to handle missing values. The default is <code>na.omit</code> .

Value

An object of class "angular".

References

Rivest, L.-P., Duchesne, T., Nicosia, A., and Fortin, D. (2016). A general angular regression model for the analysis of data on animal movement in ecology. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(3), 445-463.

angular_lrtest	<i>Likelihood-ratio test (T_2) between two homogeneous angular models</i>
----------------	--

Description

Computes the statistic from Rivest et al. (2016), Section 3.6:

$$T_2 = 2n\hat{\kappa}_1(\bar{C}_1 - \bar{C}_0),$$

where model 1 is the full model and model 0 the nested reduced model.

Usage

```
angular_lrtest(full, reduced, check = TRUE)
```

Arguments

full	Object of class "angular" (full model).
reduced	Object of class "angular" (nested reduced model).
check	Logical; perform consistency checks.

Value

An object of class "hctest".

angular_re

*Angular mixed-effects regression with von Mises random intercepts***Description**

Fits the homogeneous angular regression of Rivest et al. (2016) augmented with a cluster-specific random intercept $a_i \sim \text{VM}(0, \kappa_a)$ and unit-errors $e_{ij} \sim \text{VM}(0, \kappa_e)$, following Rivest & Kato (2019). The fixed-effects component shares the simplified syntax of `angular()`, where the first term provides the reference direction and subsequent terms are written `x:z`.

Usage

```
angular_re(
  formula,
  data,
  cluster,
  init = list(),
  control = list(),
  na.action = stats::na.omit
)
```

Arguments

<code>formula</code>	Model formula with a circular response and terms of the form <code>x</code> or <code>x:z</code> . The first angular covariate supplies the reference direction whose coefficient is fixed to one.
<code>data</code>	A data frame containing the variables used in <code>formula</code> .
<code>cluster</code>	Factor or vector of cluster identifiers. It must have length equal to <code>nrow(data)</code> or to the model-frame size after missing-value handling.
<code>init</code>	Optional list with components <code>beta</code> , <code>kappa_e</code> , and <code>kappa_a</code> providing starting values. Missing entries are filled with moment-based estimates.
<code>control</code>	Optional list controlling the optimisation with components <code>maxit</code> (default 2000), <code>reltol</code> (1e-8), <code>trace</code> (0/1), <code>hessian</code> (TRUE), and <code>start_from_angular</code> (TRUE) indicating whether <code>angular()</code> should be used to initialise the fixed effects.
<code>na.action</code>	Function used by <code>model.frame</code> to handle missing values. The default is <code>na.omit</code> .

Value

An object of class "angular_re" containing fixed-effect estimates, model-based and cluster-robust covariance matrices, concentration estimates, fitted marginal mean directions, fixed and conditional residuals, and estimated random intercepts.

References

Rivest, L.-P., Duchesne, T., Nicosia, A., and Fortin, D. (2016). A general angular regression model for the analysis of data on animal movement in ecology. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(3), 445-463.

Rivest, L.-P., and Kato, S. (2019). A random-effects model for clustered circular data. *Canadian Journal of Statistics*, 47(4), 712-728.

Examples

```
data(Sandhopper)
sh <- Sandhopper[seq_len(24), ]
sh$y <- sh$LN1 * pi / 180
sh$ref <- sh$Azimuth * pi / 180
sh$wind <- sh$DirW * pi / 180
sh$wind_speed <- sh$SpeedW / max(sh$SpeedW, na.rm = TRUE)

fit <- angular_re(
  y ~ ref + wind:wind_speed,
  data = sh,
  cluster = sh$Anim,
  control = list(maxit = 25, reltol = 1e-6)
)
coef(fit)
head(predict(fit))
```

angular_two_step

Two-step angular regression workflow (consensus then homogeneous)

Description

Implements the strategy recommended by Rivest et al. (2016): fit the consensus model first, choose the reference angle, then fit the homogeneous model with the reference fixed and initialized from consensus ratios.

Usage

```
angular_two_step(
  formula,
  data,
  weights = NULL,
  reference = c("auto", "first", "name"),
  control_consensus = list(),
  control_angular = list(),
  na.action = stats::na.omit
)
```

Arguments

formula	Model formula with terms of the form x or $x:z$.
data	Data frame containing model variables.
weights	Optional non-negative weights passed to <code>consensus()</code> .
reference	Reference-angle rule for the homogeneous step: "auto", "first", or <code>c("name", "x_ref")</code> .
control_consensus	Control list for <code>consensus()</code> .
control_angular	Control list for <code>angular()</code> .
na.action	Function used by <code>model.frame</code> to handle missing values. The default is <code>na.omit</code> .

Value

An object of class "angular_two_step".

References

Rivest, L.-P., Duchesne, T., Nicosia, A., and Fortin, D. (2016). A general angular regression model for the analysis of data on animal movement in ecology. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(3), 445-463.

autoregressivedata *Create autoregressive dataset*

Description

This function augments a data frame with lagged versions of an angular response variable so that autoregressive circular regression models can be fitted.

Usage

```
autoregressivedata(data, response, p, ID = NULL)
```

Arguments

data	A data frame containing at least the response variable.
response	A character string specifying the name of the angular response variable.
p	Integer giving the order of the autoregressive structure.
ID	Optional. A character string specifying the name of a grouping variable. If provided, lags are computed within each group.

Value

The input data with p additional columns named `response_tm k` for $k = 1, \dots, p$ containing the lagged values of response.

bison

*Bison movement data***Description**

A subset of the bison movement dataset used for illustrating angular regression models of the form $y = x_1 + x_2 : z_2 + \dots$, where ‘x.*’ denote angular covariates (target directions) and ‘z.*’ the corresponding positive covariates (weights) in the product $x : z$. The data originate from GPS-tracked bison trajectories and environmental variables describing landscape features such as meadows and canopy gaps.

Usage

bison

Format

A data frame with 5,696 rows and 7 columns:

y.dir Direction of movement at time t .

y.prec Direction of movement at time $t - 1$.

y.prec2 Direction of movement at time $t - 2$.

x.meadow Angle toward the nearest meadow.

z.meadow Distance toward the nearest meadow associated with `x.meadow`.

x.gap Angle toward the nearest canopy gap.

z.gap Distance toward the nearest canopy gap associated with `x.gap`.

Details

All angular variables in this processed dataset are stored in radians.

This reduced dataset is designed to work directly with the functions of the *CircularRegression* package implementing the general angular regression model of Rivest et al. (2016). Each $x : z$ pair represents a weighted target direction contributing to the mean resultant vector $\sum_j \beta_j z_j (\cos x_j, \sin x_j)$, where $z.*$ modulates the attraction toward the angular covariate $x.*$. Variables `y.prec` and `y.prec2` can be included to model directional persistence between successive movement steps.

Ecologically, these variables describe the orientation of bison movement relative to attractive features in the landscape: open meadows and canopy gaps. The dataset serves as a clean example for model testing, illustration, and teaching applications involving directional movement analysis.

The repository does not include the raw GPS files or a full reconstruction script for this processed dataset. The package maintainer has confirmed that this processed package dataset may be redistributed with the package.

Source

Reduced package dataset derived from the bison movement application described by Rivest et al. (2016). The repository does not include raw GPS files or a reconstruction script for this reduced teaching dataset.

References

Rivest, L.-P., Duchesne, T., Nicosia, A., and Fortin, D. (2016). A general angular regression model for the analysis of data on animal movement in ecology. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(3), 445-463.

See Also

[angular](#), [consensus](#), [pick_reference_angle](#)

Examples

```
# View structure
str(bison)
```

circular_regression *Fit Circular Regression Models with a Common Interface*

Description

`circular_regression()` is the recommended high-level interface for fixed-effect circular regression models in this package. It keeps the existing model-specific functions available while providing a common, `glm()`-like entry point for the homogeneous angular model, the consensus model, and the two-step workflow.

Usage

```
circular_regression(
  formula,
  data,
  method = c("two_step", "homogeneous", "consensus"),
  weights = NULL,
  reference = c("auto", "first", "name"),
  control = list(),
  control_consensus = list(),
  control_angular = list(),
  na.action = stats::na.omit,
  ...
)
```

Arguments

<code>formula</code>	Model formula with a circular response on the left-hand side and explanatory direction terms on the right-hand side. Terms must be of the form <code>x</code> or <code>x:z</code> , where <code>x</code> is an angular covariate and <code>z</code> is a finite non-negative modifier.
<code>data</code>	Data frame, list, or environment containing model variables.

method	Model fitting strategy. "two_step" first fits <code>consensus()</code> and then <code>angular()</code> with the selected reference angle. "homogeneous" calls <code>angular()</code> . "consensus" calls <code>consensus()</code> .
weights	Optional non-negative observation weights. Used by "two_step" and "consensus".
reference	Reference-angle rule for homogeneous fits. Use "auto", "first", or <code>c("name", "x_ref")</code> .
control	Optional control list passed to the selected model. For method = "two_step", use <code>control_consensus</code> and <code>control_angular</code> for separate control.
control_consensus	Control list for the consensus step.
control_angular	Control list for the homogeneous step.
na.action	Function used by <code>model.frame</code> to handle missing values. The default is <code>na.omit</code> .
...	Additional arguments passed to the underlying model-specific function.

Value

An object of class "circular_regression" containing the fitted model, the method used, the original call, and the formula.

References

Rivest, L.-P., Duchesne, T., Nicosia, A., and Fortin, D. (2016). A general angular regression model for the analysis of data on animal movement in ecology. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(3), 445-463.

Examples

```
data(bison)
d <- bison[seq_len(100), ]
fit <- circular_regression(
  y.dir ~ y.prec + x.meadow:z.meadow,
  data = d
)
coef(fit)
head(predict(fit))
```

consensus

Consensus Angular Regression Model

Description

Fits the consensus model of Rivest et al. (2016), where each formula term of the form `x` or `x:z` receives a κ_j coefficient in the log-likelihood parameterization.

Usage

```
consensus(
  formula,
  data,
  weights = NULL,
  initkappa = NULL,
  initbeta = NULL,
  control = list(),
  na.action = stats::na.omit
)
```

Arguments

formula	Model formula containing terms of the form x or $x:z$.
data	Data frame, list or environment containing model variables.
weights	Optional non-negative weights.
initkappa	Optional numeric vector of initial κ_j values.
initbeta	Deprecated alias for <code>initkappa</code> , retained for backward compatibility.
control	Optional list with <code>pginit</code> , <code>maxiter</code> , <code>mindiff</code> .
na.action	Function used by <code>model.frame</code> to handle missing values. The default is <code>na.omit</code> .

Value

An object of class "consensus".

References

Rivest, L.-P., Duchesne, T., Nicosia, A., and Fortin, D. (2016). A general angular regression model for the analysis of data on animal movement in ecology. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(3), 445-463.

decentredPredictorModel

Decentered Predictor Model Wrapper (two-step workflow)

Description

Fits the decentered-predictor special case by augmenting the data with synthetic orthogonal directions and applying `angular_two_step()`. Natural parameters are returned with model-based and robust delta-method standard errors.

Usage

```
decentredPredictorModel(data, response, w, ...)
```

Arguments

data	A data frame containing response and w.
response	Name of the angular response variable.
w	Name of the angular explanatory variable.
...	Additional arguments passed to angular_two_step().

Value

A list with consensus_fit, homogeneous_fit, reference, data_aug, formula, and natural-parameter outputs natural_parameters, natural_vcov_model, natural_vcov_robust, natural_jacobian.

References

Rivest, L.-P., Duchesne, T., Nicosia, A., and Fortin, D. (2016). A general angular regression model for the analysis of data on animal movement in ecology. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(3), 445-463.

Examples

```
data(bison)
d <- na.omit(bison[seq_len(80), c("y.dir", "y.prec")])
fit <- decentredPredictorModel(d, response = "y.dir", w = "y.prec")
fit$natural_parameters
```

jammalamadakaModel *Jammalamadaka-Sengupta Model Wrapper (two-step workflow)*

Description

Fits the Jammalamadaka-Sengupta special case by creating the required sine/cosine-equivalent angular terms and returning the corresponding natural parameters through a delta-method transformation.

Usage

```
jammalamadakaModel(data, response, w, ...)
```

Arguments

data	A data frame containing response and w.
response	Name of the angular response variable.
w	Name of the angular explanatory variable.
...	Additional arguments passed to angular_two_step().

Value

A list with `consensus_fit`, `homogeneous_fit`, `reference`, `data_aug`, `formula`, and natural-parameter outputs `natural_parameters`, `natural_vcov_model`, `natural_vcov_robust`, `natural_jacobian`.

References

Rivest, L.-P., Duchesne, T., Nicosia, A., and Fortin, D. (2016). A general angular regression model for the analysis of data on animal movement in ecology. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(3), 445-463.

Examples

```
data(bison)
d <- na.omit(bison[seq_len(80), c("y.dir", "y.prec")])
fit <- jammalamadakaModel(d, response = "y.dir", w = "y.prec")
fit$natural_parameters
```

meanDirectionModel *Mean Direction Model Wrapper (two-step workflow)*

Description

Fits the mean-direction special case through the same two-step workflow used by `angular_two_step()`. The wrapper creates synthetic reference directions internally and returns natural-parameter summaries obtained by a delta-method transformation of the homogeneous fit.

Usage

```
meanDirectionModel(data, response, mu0 = NULL, ...)
```

Arguments

<code>data</code>	A data frame containing at least the response variable.
<code>response</code>	Name of the angular response variable.
<code>mu0</code>	Optional finite numeric scalar or vector of length <code>nrow(data)</code> used as a baseline direction.
<code>...</code>	Additional arguments passed to <code>angular_two_step()</code> .

Value

A list with `consensus_fit`, `homogeneous_fit`, `reference`, `data_aug`, `formula`, and natural-parameter outputs `natural_parameters`, `natural_vcov_model`, `natural_vcov_robust`, `natural_jacobian`.

References

Rivest, L.-P., Duchesne, T., Nicosia, A., and Fortin, D. (2016). A general angular regression model for the analysis of data on animal movement in ecology. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(3), 445-463.

Examples

```
data(bison)
d <- na.omit(bison[seq_len(80), "y.dir", drop = FALSE])
fit <- meanDirectionModel(d, response = "y.dir")
fit$natural_parameters
```

multiplebison

*Two plains bison tracked by GPS (Prince Albert National Park, 2013)***Description**

Hourly GPS-derived movement metrics for two plains bison (IDs ‘1044-a’ and ‘1045-a’) monitored in Prince Albert National Park (Saskatchewan, Canada) during July-October 2013. From the hourly locations, the dataset includes per-animal direction of movement, step length (distance between consecutive fixes), latitude/longitude, the inter-animal distance at each hour, and turning-angles (difference between consecutive movement directions).

Usage

```
multiplebison
```

Format

A data frame with $n = 1475$ rows (hours kept after filtering) and the following columns (direction and turning-angle variables are in degrees):

datetime_round_next POSIXct. Timestamp of the retained hourly record.

heure integer. Hour of day (0-23).

bearing_ang numeric. Bearing from bison 1044-a to 1045-a.

diff numeric. Source variable retained from the merged GPS-derived data.

direction_1044-a numeric. Movement direction of bison 1044-a, in degrees.

direction_1045-a numeric. Movement direction of bison 1045-a, in degrees.

distance_1044-a numeric. Step length (m) of bison 1044-a between consecutive fixes.

distance_1045-a numeric. Step length (m) of bison 1045-a between consecutive fixes.

distance numeric. Inter-animal distance (m) at the timestamp.

lat_1044-a numeric. Latitude of bison 1044-a (decimal degrees).

lat_1045-a numeric. Latitude of bison 1045-a (decimal degrees).

long_1044-a numeric. Longitude of bison 1044-a (decimal degrees).

long_1045-a numeric. Longitude of bison 1045-a (decimal degrees).

turn_ang_1044 numeric. Turning angle of bison 1044-a between consecutive directions, in degrees.

turn_ang_1045 numeric. Turning angle of bison 1045-a between consecutive directions, in degrees.

Details

The file results from merging individual hourly GPS tracks (one per animal), retaining rows where the time gap between successive fixes is exactly one hour, and reshaping to a "wide" format with one row per timestamp holding both animals' variables. Turning-angles are computed from consecutive directions.

- Angles are measured around the circle and stored in degrees in this package dataset. Convert to radians before modeling, for example with $x * \pi / 180$. - Step lengths and distances were computed from consecutive GPS fixes; rows are kept only when the time difference equals one hour. - The turning-angle columns are circular differences between consecutive movement directions, stored in degrees on a 0-360 scale. - Scientific provenance is partially documented in the repository. The maintainer identifies these GPS tracks as associated with doctoral work and/or an article on bison movement. A closely related scientific article is Nicosia et al. (2017), which analyzes GPS-collared bison movement in Prince Albert National Park using an individual winter trajectory from November 2013 to April 2014. The repository does not verify whether the current two-bison July-October 2013 data file is a direct subset or preprocessing product of that analysis. - The repository does not include raw GPS files or a full reconstruction script for this processed dataset. The package maintainer has confirmed that this processed package dataset may be redistributed with the package.

Source

Derived package dataset prepared from GPS tracks of two plains bison monitored in Prince Albert National Park in 2013. The repository does not include raw GPS files or a reconstruction script. The package maintainer has confirmed that this processed package dataset may be redistributed with the package. Nicosia et al. (2017) is a related bison movement article from the same study context, but the exact mapping from that article to this processed data file is not verified in the repository.

References

Nicosia, A., Duchesne, T., Rivest, L.-P., and Fortin, D. (2017). A multi-state conditional logistic regression model for the analysis of animal movement. *The Annals of Applied Statistics*, 11(3), 1537-1560. DOI: [doi:10.1214/17AOAS1045](https://doi.org/10.1214/17AOAS1045).

Examples

```
## Basic glimpse
head(multiplebison)

## Example: inter-animal proximity rate (< 500 m)
# mean(multiplebison$distance < 500, na.rm = TRUE)
```

Description

This data frame contains two circular variables for 678 locations associated with the Noshiro earthquake. The variables are DIRDSC, the direction of steepest descent, and DIRMV, the direction of movement.

Usage

```
data(noshiro)
```

Format

A data frame with 678 rows and 2 columns:

DIRDSC Direction of steepest descent.

DIRMV Direction of movement.

Details

The data were used by Rivest (1997) for circular-circular regression. The processed version included in this package was provided by Louis-Paul Rivest to the package maintainer after duplicate observations had been removed, yielding $n = 678$. In the original analysis, DIRMV is the response variable to be predicted from DIRDSC.

The angular units are not documented in the repository or in the provenance notes provided to the maintainer. Raw data and a reconstruction script are not present in the repository. The package maintainer has confirmed that this processed package dataset may be redistributed with the package.

Source

Processed Noshiro earthquake data used in Rivest (1997), provided by Louis-Paul Rivest to the package maintainer. Duplicate observations were removed before inclusion in this package, yielding $n = 678$. The package maintainer has confirmed that this processed package dataset may be redistributed with the package.

References

Rivest, L.-P. (1997). A decentered predictor for circular-circular regression. *Biometrika*, 84, 717–726.

pick_reference_angle *Backward-compatible wrapper for reference-angle selection*

Description

Backward-compatible wrapper for reference-angle selection

Usage

```
pick_reference_angle(formula, data, tie_method = c("first", "random"), ...)
```

Arguments

formula	Model formula with terms x or x:z.
data	Data frame containing model variables.
tie_method	Tie-break rule for equal scores.
...	Unused.

Value

A select_reference_angle object, with legacy class pick_reference_angle prepended.

plot.angular_re	<i>Plot method for angular_re objects</i>
-----------------	---

Description

Draws rose/circular plots of residuals: fixed ($y - \mu$) and conditional ($y - \mu - \hat{a}_{\text{cluster}}$) as in Rivest & Kato (2019), Fig. 1-2. If the 'circular' package is available, uses it; otherwise falls back to a simple polar scatter with density ticks.

Usage

```
## S3 method for class 'angular_re'
plot(
  x,
  which = c("both", "fixed", "conditional"),
  breaks = 24,
  points = TRUE,
  main = "Residuals",
  ...
)
```

Arguments

x	an object of class angular_re
which	one of "both", "fixed", "conditional"
breaks	number of bins for rose histogram when using circular
points	logical; add individual points on the circle
main	base title (panel-specific suffixes are added)
...	unused

plot_ranef.angular_re *Plot random-intercept directions for an angular_re fit*

Description

Draws one arrow per cluster pointing in the direction of the estimated random intercept \hat{a}_i . Arrow length is either $A1(\text{kappa}_a)$ or 1. If the 'circular' package is available, uses it for nicer axes; otherwise falls back to base polar plotting.

Usage

```
plot_ranef.angular_re(
  x,
  scale = c("A1_kappa_a", "unit"),
  labels = FALSE,
  cex.labels = 0.7,
  main = "Random intercepts (a_hat[i])",
  ...
)
```

Arguments

x	an object of class angular_re
scale	character, one of "A1_kappa_a" (default) or "unit". Controls the common arrow length.
labels	logical; draw cluster labels at arrow tips.
cex.labels	numeric expansion for labels.
main	plot title.
...	unused

predict.angular_re *Predict method for angular_re objects*

Description

Predict method for angular_re objects

Usage

```
## S3 method for class 'angular_re'
predict(
  object,
  newdata = NULL,
  cluster = NULL,
  type = c("auto", "marginal", "conditional"),
  a_hat = NULL,
  se.fit = FALSE,
  ...
)
```

Arguments

object	an angular_re fit.
newdata	data.frame; if NULL, uses model frame from the original call.
cluster	optional vector/factor of cluster IDs for newdata.
type	One of "auto" (default), "marginal" (mean direction) or "conditional" (mean direction plus the predicted random intercept). With "auto" the conditional mean is returned whenever cluster information is provided; otherwise the marginal mean is used.
a_hat	optional numeric vector of random-intercept predictions to use for cluster (same length as newdata). Ignored unless type="conditional".
se.fit	not implemented (reserved).
...	unused

Value

numeric vector (angles in radians).

presnellModel

Presnell Model Wrapper (two-step workflow)

Description

Fits the Presnell special case by building the linear and orthogonal synthetic terms required by the homogeneous angular formulation. The returned natural parameters are ratios computed from the homogeneous coefficients.

Usage

```
presnellModel(data, response, w, ...)
```

Arguments

data	A data frame containing response and w.
response	Name of the angular response variable.
w	Name of the continuous covariate.
...	Additional arguments passed to angular_two_step().

Value

A list with consensus_fit, homogeneous_fit, reference, data_aug, formula, and natural-parameter outputs natural_parameters, natural_vcov_model, natural_vcov_robust, natural_jacobian.

References

Rivest, L.-P., Duchesne, T., Nicosia, A., and Fortin, D. (2016). A general angular regression model for the analysis of data on animal movement in ecology. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(3), 445-463.

Examples

```
data(bison)
d <- na.omit(bison[500:579, c("y.dir", "z.gap")])
fit <- presnellModel(d, response = "y.dir", w = "z.gap")
fit$natural_parameters
```

print.angular

Methods for Angular Regression Objects

Description

Methods for Angular Regression Objects

Usage

```
## S3 method for class 'angular'
print(x, ...)

## S3 method for class 'angular'
coef(object, ...)

## S3 method for class 'angular'
vcov(object, robust = FALSE, ...)

## S3 method for class 'angular'
fitted(object, ...)

## S3 method for class 'angular'
```

```

residuals(object, ...)

## S3 method for class 'angular'
predict(object, newdata = NULL, type = c("response", "components"), ...)

## S3 method for class 'angular'
summary(object, ...)

## S3 method for class 'summary.angular'
print(x, ...)

## S3 method for class 'angular'
plot(x, ...)

```

Arguments

x	An object of class "angular".
...	Additional arguments.
object	An object of class "angular".
robust	Logical; for vcov(), return the sandwich covariance matrix when TRUE.
newdata	Optional data frame for prediction.
type	Prediction type. Use "response" for fitted angles in radians or "components" for cosine and sine components.

print.angular_re *Methods for Angular Random-Effects Fits*

Description

Methods for Angular Random-Effects Fits

Usage

```

## S3 method for class 'angular_re'
print(x, ...)

## S3 method for class 'angular_re'
summary(object, ...)

## S3 method for class 'summary.angular_re'
print(x, ...)

## S3 method for class 'angular_re'
coef(object, ...)

## S3 method for class 'angular_re'

```

```
vcov(object, robust = FALSE, ...)

## S3 method for class 'angular_re'
residuals(object, type = c("fixed", "conditional"), ...)

## S3 method for class 'angular_re'
fitted(object, ...)
```

Arguments

x, object	An object of class "angular_re".
...	Additional arguments, currently unused.
robust	Logical. For <code>vcov()</code> , return the cluster-robust covariance matrix when TRUE; otherwise return the model-based covariance matrix.
type	Residual type. "fixed" returns residuals from the marginal fixed-effect mean direction. "conditional" subtracts the estimated cluster random intercepts.

Value

`coef()` returns the parameter estimates. `vcov()` returns the requested covariance matrix. `fitted()` returns marginal fitted mean directions. `residuals()` returns wrapped circular residuals. `summary()` returns a list with convergence information, likelihood, sample sizes, concentration estimates, residual summaries and coefficient table. `print()` invisibly returns the input object.

```
print.angular_two_step
```

Methods for two-step angular regression objects

Description

Methods for two-step angular regression objects

Usage

```
## S3 method for class 'angular_two_step'
print(x, ...)

## S3 method for class 'angular_two_step'
summary(object, ...)

## S3 method for class 'summary.angular_two_step'
print(x, ...)

## S3 method for class 'angular_two_step'
coef(object, step = c("homogeneous", "consensus"), ...)
```

```
## S3 method for class 'angular_two_step'
fitted(object, ...)

## S3 method for class 'angular_two_step'
residuals(object, ...)

## S3 method for class 'angular_two_step'
predict(object, newdata = NULL, type = c("response", "components"), ...)

## S3 method for class 'angular_two_step'
plot(x, ...)

## S3 method for class 'angular_two_step'
logLik(object, ...)

## S3 method for class 'angular_two_step'
AIC(object, ..., k = 2)

## S3 method for class 'angular_two_step'
BIC(object, ...)
```

Arguments

x, object	An object of class "angular_two_step".
...	Additional arguments passed to the underlying method.
step	Which fitted step to use for <code>coef()</code> .
newdata	Optional data frame for prediction.
type	Prediction type passed to <code>predict.angular()</code> .
k	Numeric penalty used by <code>AIC()</code> .

```
print.circular_regression
```

Methods for Circular Regression Objects

Description

Methods for Circular Regression Objects

Usage

```
## S3 method for class 'circular_regression'
print(x, ...)

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

```

## S3 method for class 'summary.circular_regression'
print(x, ...)

## S3 method for class 'circular_regression'
coef(object, ...)

## S3 method for class 'circular_regression'
fitted(object, ...)

## S3 method for class 'circular_regression'
residuals(object, ...)

## S3 method for class 'circular_regression'
predict(object, newdata = NULL, type = c("response", "components"), ...)

## S3 method for class 'circular_regression'
plot(x, ...)

## S3 method for class 'circular_regression'
logLik(object, ...)

## S3 method for class 'circular_regression'
AIC(object, ..., k = 2)

## S3 method for class 'circular_regression'
BIC(object, ...)

```

Arguments

x, object	A "circular_regression" object.
...	Additional arguments passed to the underlying method.
newdata	Optional data frame for prediction.
type	Prediction type passed to the underlying model.
k	Numeric penalty used by AIC().

print.consensus

Methods for Consensus Objects

Description

Methods for Consensus Objects

Usage

```
## S3 method for class 'consensus'  
print(x, ...)  
  
## S3 method for class 'consensus'  
coef(  
  object,  
  type = c("kappa", "beta"),  
  reference = c("auto", "first", "name"),  
  ...  
)  
  
## S3 method for class 'consensus'  
vcov(object, robust = FALSE, ...)  
  
## S3 method for class 'consensus'  
fitted(object, ...)  
  
## S3 method for class 'consensus'  
residuals(object, ...)  
  
## S3 method for class 'consensus'  
predict(object, newdata = NULL, type = c("response", "components"), ...)  
  
## S3 method for class 'consensus'  
summary(object, ...)  
  
## S3 method for class 'summary.consensus'  
print(x, ...)  
  
## S3 method for class 'consensus'  
plot(x, ...)  
  
## S3 method for class 'consensus'  
AIC(object, ..., k = 2)  
  
## S3 method for class 'consensus'  
BIC(object, ...)  
  
## S3 method for class 'consensus'  
logLik(object, ...)
```

Arguments

x	An object of class "consensus".
...	Additional arguments.
object	An object of class "consensus".

type	For coef.consensus(), either "kappa" or "beta". The latter returns ratios relative to a selected reference.
reference	Reference selection rule for type = "beta".
robust	Logical; for vcov(), accepted for consistency and currently ignored because the consensus model stores one covariance matrix.
newdata	Optional data frame for prediction.
k	Numeric penalty for AIC.

```
print.pick_reference_angle
```

Print method for pick_reference_angle

Description

Print method for pick_reference_angle

Usage

```
## S3 method for class 'pick_reference_angle'
print(x, ...)
```

Arguments

x	Object returned by pick_reference_angle().
...	Unused.

```
ranef.angular_re
```

Extract estimated random intercepts from an angular mixed model

Description

Returns the cluster-specific random intercepts \hat{a}_i estimated by angular_re().

Usage

```
ranef.angular_re(object, ...)
```

Arguments

object	A fitted angular_re model.
...	Unused; included for compatibility with the generic.

Value

A named numeric vector of random intercept estimates, one per cluster.

 Sandhopper

Sandhopper Escape Orientation Data

Description

This dataset contains repeated measurements on the escape orientation of 59 sandhoppers (*Talitrus saltator*) in a circular arena experiment with a diameter of 120 cm. Each individual was released five times, and its escape angle with respect to the north was recorded. Several explanatory variables were also measured during the experiment.

Usage

```
data(Sandhopper)
```

Format

A data frame with 72 rows and 22 variables:

Anim Individual identifier (integer).

Day Day of the experiment.

Daytime Part of the day (character, e.g., "mor", "aft").

Azimuth Sun azimuth at release (degrees).

LN1 Escape orientation, trial 1 (degrees).

LN2 Escape orientation, trial 2 (degrees).

LN3 Escape orientation, trial 3 (degrees).

LN4 Escape orientation, trial 4 (degrees).

LN5 Escape orientation, trial 5 (degrees).

Umidity Relative humidity (%).

Temp Air temperature (degrees Celsius).

Tsab Sand temperature (degrees Celsius).

DirW Wind direction (degrees).

SpeedW Wind speed (arbitrary units).

Sex Sex of the sandhopper ("f" or "m").

Weight Weight of the sandhopper (grams).

Odmx Morphometric measurement.

Odmn Morphometric measurement.

Osmx Morphometric measurement.

Osmn Morphometric measurement.

Artdx Morphometric measurement.

Artsx Morphometric measurement.

Details

The dataset was originally presented and analyzed by D'Elia (2001) and later used in Rivest & Kato (2019) to illustrate angular regression models with repeated measures.

Angular variables such as Azimuth, LN1-LN5 and DirW are stored in degrees. The repository does not include a full raw-data reconstruction script for this processed dataset. The package maintainer has confirmed that this processed package dataset may be redistributed with the package.

Source

Package data file corresponding to the sandhopper example discussed by D'Elia (2001) and Rivest and Kato (2019). The repository does not include a raw-data reconstruction script.

References

D'Elia, G. (2001). Orientation in sandhoppers. **Behavioural Ecology**. Rivest, L.-P., & Kato, S. (2019). A random-effects model for clustered circular data. **Canadian Journal of Statistics**, 47(4), 712-728.

select_reference_angle

Select Reference Angle for Homogeneous Angular Model

Description

Selects the reference direction using the criterion of Rivest et al. (2016): the angle variable maximizing the empirical mean cosine $\cos(y - x_j)$ among plain angle terms x present in the formula.

Usage

```
select_reference_angle(
  formula,
  data,
  tie_method = c("first", "random"),
  na.action = stats::na.omit
)
```

Arguments

formula	Model formula containing terms of the form x or $x:z$.
data	Data frame used to evaluate the formula.
tie_method	Tie-break rule when multiple candidates share the same score: "first" (default) or "random".
na.action	Function used by <code>model.frame</code> to handle missing values. The default is <code>na.omit</code> .

Value

An object of class "select_reference_angle".

Index

* datasets

- bison, 7
 - multiplebison, 13
 - noshiro, 14
 - Sandhopper, 26
- AIC.angular_two_step
(print.angular_two_step), 21
- AIC.circular_regression
(print.circular_regression), 22
- AIC.consensus (print.consensus), 23
- angular, 2, 8, 9
- angular_lrtest, 3
- angular_re, 4
- angular_two_step, 5, 10, 12
- autoregressivedata, 6
- BIC.angular_two_step
(print.angular_two_step), 21
- BIC.circular_regression
(print.circular_regression), 22
- BIC.consensus (print.consensus), 23
- bison, 7
- circular_regression, 8
- coef.angular (print.angular), 19
- coef.angular_re (print.angular_re), 20
- coef.angular_two_step
(print.angular_two_step), 21
- coef.circular_regression
(print.circular_regression), 22
- coef.consensus (print.consensus), 23
- consensus, 8, 9, 9
- decentredPredictorModel, 10
- fitted.angular (print.angular), 19
- fitted.angular_re (print.angular_re), 20
- fitted.angular_two_step
(print.angular_two_step), 21
- fitted.circular_regression
(print.circular_regression), 22
- fitted.consensus (print.consensus), 23
- jammalamadakaModel, 11
- logLik.angular_two_step
(print.angular_two_step), 21
- logLik.circular_regression
(print.circular_regression), 22
- logLik.consensus (print.consensus), 23
- meanDirectionModel, 12
- model.frame, 3, 4, 6, 9, 10, 27
- multiplebison, 13
- na.omit, 3, 4, 6, 9, 10, 27
- noshiro, 14
- pick_reference_angle, 8, 15
- plot.angular (print.angular), 19
- plot.angular_re, 16
- plot.angular_two_step
(print.angular_two_step), 21
- plot.circular_regression
(print.circular_regression), 22
- plot.consensus (print.consensus), 23
- plot_ranef.angular_re, 17
- predict.angular, 22
- predict.angular (print.angular), 19
- predict.angular_re, 17
- predict.angular_two_step
(print.angular_two_step), 21
- predict.circular_regression
(print.circular_regression), 22
- predict.consensus (print.consensus), 23
- presnellModel, 18
- print.angular, 19
- print.angular_re, 20
- print.angular_two_step, 21
- print.circular_regression, 22

`print.consensus`, 23
`print.pick_reference_angle`, 25
`print.summary.angular` (`print.angular`),
19
`print.summary.angular_re`
(`print.angular_re`), 20
`print.summary.angular_two_step`
(`print.angular_two_step`), 21
`print.summary.circular_regression`
(`print.circular_regression`), 22
`print.summary.consensus`
(`print.consensus`), 23

`ranef.angular_re`, 25
`residuals.angular` (`print.angular`), 19
`residuals.angular_re`
(`print.angular_re`), 20
`residuals.angular_two_step`
(`print.angular_two_step`), 21
`residuals.circular_regression`
(`print.circular_regression`), 22
`residuals.consensus` (`print.consensus`),
23

Sandhopper, 26
`select_reference_angle`, 27
`summary.angular` (`print.angular`), 19
`summary.angular_re` (`print.angular_re`),
20
`summary.angular_two_step`
(`print.angular_two_step`), 21
`summary.circular_regression`
(`print.circular_regression`), 22
`summary.consensus` (`print.consensus`), 23

`vcov.angular` (`print.angular`), 19
`vcov.angular_re` (`print.angular_re`), 20
`vcov.consensus` (`print.consensus`), 23