Package 'brokenstick'

November 2, 2020

Type Package

Title Broken Stick Model for Irregular Longitudinal Data

Version 1.1.0

Description The broken stick model describes a set of individual curves by a linear mixed model using a second-order linear B-spline. The main use of the model is to align irregularly observed data to a user-specified grid of break ages. All fitting can done in the Z-score scale, so non-linearity and irregular data can be treated as separate problems. This package contains functions for fitting a broken stick model to data, for predicting broken stick curves in new data, and for plotting the broken stick estimates. For additional documentation on background, methodology and applications of the broken stick model see <https://stefvanbuuren.name/publications/2020_Brokenstick_JSS_manuscript.pdf>.

Depends R (>= 3.5.0)

Imports dplyr, lme4, matrixsampling, methods, rlang, splines, stats, tidyr

Suggests AGD, ggplot2, grDevices, gridExtra, knitr, lattice, MASS, Matrix, mice, mvtnorm, plyr, svglite, testthat, rmarkdown

URL https://github.com/growthcharts/brokenstick, https://growthcharts.org/brokenstick/

BugReports https://github.com/growthcharts/brokenstick/issues

Encoding UTF-8

License MIT + file LICENSE

LazyData TRUE

VignetteBuilder knitr

RoxygenNote 7.1.1

NeedsCompilation no

Author Stef van Buuren [aut, cre]

Maintainer Stef van Buuren <stef.vanbuuren@tno.nl>

Repository CRAN

Date/Publication 2020-11-02 09:30:07 UTC

R topics documented:

`brokenstick` *Fit a* `brokenstick` *model to irregular data*

Description

The `brokenstick()` function fits an irregularly observed series of measurements onto a user-specified grid of points. The model codes de grid by a series of linear B-splines. Differences between observations are expressed by one random effect per grid point. When multiple set of series are modelled, each modeled trajectory consists of straight lines that join at the chosen grid points, and hence look like a broken stick.

Usage

```
brokenstick(x, ...)

## Default S3 method:
brokenstick(x, ...)

## S3 method for class 'formula'
brokenstick(
  formula,
  data,
  ...,
  knots = NULL,
```

```
  boundary = NULL,
  k = NULL,
  degree = 1L,
  method = c("lmer", "kr"),
  control = control_brokenstick(),
  seed = NA
)

## S3 method for class 'data.frame'
brokenstick(
  x,
  y,
  group,
  ...,
  knots = NULL,
  boundary = NULL,
  k = NULL,
  degree = 1L,
  method = c("lmer", "kr"),
  control = control_brokenstick(),
  seed = NA
)

## S3 method for class 'matrix'
brokenstick(
  x,
  y,
  group,
  ...,
  knots = NULL,
  boundary = NULL,
  k = NULL,
  degree = 1L,
  method = c("lmer", "kr"),
  control = control_brokenstick(),
  seed = NA
)

## S3 method for class 'numeric'
brokenstick(
  x,
  y,
  group,
  ...,
  knots = NULL,
  boundary = NULL,
  k = NULL,
  degree = 1L,
```

```
  method = c("lmer", "kr"),
  control = control_brokenstick(),
  seed = NA
)
```

Arguments

x	Predictor variables. Depending on the context: • A **data frame** of predictors. • A **matrix** of predictors. If x has one column, then also specify y and group. If x has multiple columns, then specify the model by a formula argument.
...	Not currently used, but required for extensibility.
formula	A formula specifying the outcome terms on the left-hand side, the predictor term on the right-hand side and the group variable after the | sign, e.g formula = hgt ~ age | id. One may specify additional variables, but the brokenstick model will ignored them. Note: This formula specification is specific to the brokenstick() function.
data	When a **formula** is used, data is specified as: • A **data frame** containing predictor, group and outcome variables.
knots	Optional, but recommended. Numerical vector with the locations of the breaks to be placed on the values of the predictor. Values outside the range of the data will extend the boundary knots (see below) beyond the data range.
boundary	Optional. Numerical vector of length 2 with the left and right boundary knots. The boundary setting is passed to splines::bs() as the Boundary.knots argument. If not specified, the range of predictor variable is taken. Automatic model specification is data-dependent. However, if both knots and boundary are supplied, the B-spline transformation parameter do not depend on the data. If specified, the boundary range is internally expanded to include at least range(knots). The warning some 'x' values beyond boundary knots may cause ill-conditioned bases implies that model fitting ignores any data beyond the (expanded) boundary range. It is possible to prevent this warning by pre-filtering rows in data to the boundary range.
k	Optional, a convenience parameter for the number of internal knots. If specified, then k internal knots are placed at equidense quantiles of the predictor. For example, specifying k = 1 puts a knot at the 50th quantile (median), setting k = 3 puts knots at the 25th, 50th and 75th quantiles, and so on. Note: Knots specified via k are data-dependent and do not transfer to other data sets. We recommend using knots and boundary over k. If both k and knots are specified, then k takes precendence.
degree	the degree of the spline. The broken stick model requires linear splines, so the default is degree = 1. Setting degree = 0 yields (crisp) dummy coding, and one column less than for degree = 1. The brokenstick package supports only degree = 0 and degree = 1.

method	Estimation method. Either "kr" (for the Kasim-Raudenbush sampler) or "lmer" (for lme4::lmer()) (default).
control	A list with parameters. Use control_brokenstick() to generate.
seed	Seed number for base::set.seed(). Use NA to bypass seed setting.
y	Outcome variable. When x is a **data frame** or **matrix**, y is specified as: • A **data frame** with 1 numeric column. • A **matrix** with 1 numeric column. • A numeric **vector**.
group	Grouping variable. When x is a **data frame** or **matrix**, group is specified as: • A **data frame** with 1 column. • A **matrix** with 1 column. • A numeric **vector**.

Details

The variance-covariance matrix of the random effects absorbs the relations over time. By default, this matrix is estimated as unstructured by lme4::lmer(). This estimate may be unstable if the number of children is small relative to the number of specified knots. The function can be time consuming for data sets with thousands of children.

An alternative - often faster for models with many random effects - is to use the Bayesian Kasim-Raudenbush sampler (method kr). That method also allow for enforcing a simple structure on the variance-covariance matrix of the random effects. Currently, there are three such models: argyle, cole and none. See kr() and control_brokenstick() for more details.

Value

A brokenstick object.

Examples

```
train <- smocc_200[1:1198, ]

# fit with implicit boundary c(0, 3)
fit <- brokenstick(hgt.z ~ age | id, data = train, knots = 0:3)

# using KR sampler
fit <- brokenstick(hgt.z ~ age | id, data = train, knots = 0:3,
                   method = "kr", seed = 1)


knots <- round(c(0, 1, 2, 3, 6, 9, 12, 15, 18, 24) / 12, 4)
boundary <- c(0, 3)
fit_lmer <- brokenstick(hgt.z ~ age | id, data = train,
                        knots = knots, boundary = boundary)
fit_kr <- brokenstick(hgt.z ~ age | id, data = train, knots = knots,
                      boundary = boundary, method = "kr")
```

```
# Four ways to specify the same model
# Formula interface
mod1 <- brokenstick(hgt.z ~ age | id, train)

# XY interface - numeric vector
mod2 <- with(train, brokenstick(age, hgt.z, id))
identical(mod1, mod2)

# XY interface - data.frame
mod3 <- with(train, brokenstick(data.frame(age), hgt.z, id))
identical(mod1, mod3)

# XY interface - matrix
tt <- as.matrix(train[, c(1, 2, 7)])
mod4 <- brokenstick(tt[, "age", drop = FALSE],
                    tt[, "hgt.z", drop = FALSE],
                    tt[, "id", drop = FALSE])
identical(mod1, mod4)
```

`brokenstick-class` *Class* `brokenstick`

Description

The main fitting function `brokenstick()` returns an object of class `brokenstick`. This object collects the fitted broken stick model.

Details

The package exports S3 methods for the `brokenstick` class for the following generic functions: `fitted()`, `model.frame()`, `model.matrix()`, `plot()`, `predict()`, `print()`, `residuals()`.

The package documents methods `fitted.brokenstick()`, `plot.brokenstick()`, `predict.brokenstick()` and `residuals.brokenstick()`. The package exports two helper functions for `brokenstick` objects: `get_knots()` and `get_r2()`.

A `brokenstick` object is a list with the following named elements:

Elements

`names` A list with elements named `x`, `y` and `g` providing the variables names for the time, outcome and subject columns, respectively.

`knots` Numeric vector of with the internal knots. Use `get_knots()` to extract knots.

`boundary` Numeric vector of length 2 with the boundary knots. Use `get_knots()` to extract knots.

`degree` The degree of the B-spline. See `splines::bs()`. Either 0 (constant model) or 1 (broken stick model).

`model` Not used.

`method` Either `lmer` or `kr`, identifying the fitting model.

`control` List of control option returned by `control_brokenstick()`.

`beta` Numeric vector with fixed effect estimates.

`omega` Numeric matrix with variance-covariance estimates of the random effect.

`sigma2j` Numeric vector with estimates of the residual variance per group. Only used by method `kr`.

`sigma2` Numeric scalar with the mean residual variance.

`draws` Numeric matrix with multiple imputations. The number of rows is equal to the number of missing values in `y`. The number of columns depends on `imp_skip`. Only used by `kr` if `imp_skip` is set.

Author(s)

Stef van Buuren, 2020

References

van Buuren S (2020). Broken Stick Model for Irregular Longitudinal Data. *In preparation*.

`brokenstick-pkg` **brokenstick***: A package for irregular longitudinal data.*

Description

The broken stick model describes a set of individual curves by a linear mixed model using second-order linear B-splines. The main use of the model is to align irregularly observed data to a user-specified grid of break ages.

Details

The **brokenstick** package contains functions for fitting a broken stick model to data, for predicting broken stick curves for new data, and for plotting the results.

brokenstick functions

The main functions are:

`brokenstick()`	Fit a broken stick model to irregular data
`predict()`	Obtain predictions on new data
`plot()`	Plot observed and fitted trajectories by group

The following functions are user-oriented helpers:

`fitted()`	Calculate fitted values
`get_knots()`	Obtain the knots from a broken stick model
`get_r2()`	Obtain proportion of explained variance

`residuals()`	Extract residuals from broken stick model

The following functions perform the calculations:

`control_brokenstick()`	Set controls to steer calculations
`EB()`	Empirical Bayes predictor for random effects
`kr()`	Kasim-Raudenbush sampler for two-level normal model
`make_basis()`	Create linear splines basis

The package follows the `tidymodels` conventions https://tidymodels.github.io/model-implementation-principle For example, training data are not stored in the modelling object and calculated variables are named after the convention. The package architecture borrows important ideas from the `hardhat` package.(Vaughan, 2020)

Note

Development of this package was kindly supported under the Healthy Birth, Growth and Development knowledge integration (HBGDki) program of the Bill & Melinda Gates Foundation.

References

van Buuren, S. (2018). *Flexible Imputation of Missing Data. Second Edition*. Chapman & Hall/CRC. Chapter 11. https://stefvanbuuren.name/fimd/sec-rastering.html#sec:brokenstick

Vaughan, D. and Kuhn, M. (2020). *hardhat: Construct Modeling Packages*. R package version 0.1.4. https://CRAN.R-project.org/package=hardhat

See Also

`brokenstick`, `EB`, `predict.brokenstick`

`control_brokenstick` *Set controls to steer calculations*

Description

Set controls to steer calculations

Usage

```
control_brokenstick(
  lmer = lmerControl(check.nobs.vs.nRE = "warning"),
  kr = control_kr(),
  na.action = na.exclude
)
```

Arguments

lmer	A list generated by lme4::lmerControl. The default is set to lmerControl(check.nobs.vs.nRE = "warning"), which turn fatal errors with respect the number of parameters into warnings. Use lmerControl(check.nobs.vs.nRE = "ignore") to silence lmer().
kr	A list generated by control_kr.
na.action	The function to call for the na.action argument. The default is na.exclude.

---

control_kr *Set controls for Kasim-Raudenbush sampler*

Description

Set controls for Kasim-Raudenbush sampler

Usage

```
control_kr(
  model = c("argyle", "cole", "none"),
  runin = 100L,
  ndraw = 200L,
  par_skip = 1L,
  imp_skip = Inf
)
```

Arguments

model	Correlation model: argyle (default), cole or none
runin	Number of run-in iterations
ndraw	Number of parameter draws
par_skip	Number of iterations to next parameter draw
imp_skip	Number of iterations to next outcome draw

Value

A list with five components

EB *Empirical Bayes predictor for random effects*

Description

This function can estimate random effect for a given set of model estimates and new user data. The unit may be new to the model. The methods implements the EB estimate (also known as BLUP) as described in Skrondral and Rabe-Hasketh, 2009, p. 683. This function can also provide the broken stick estimate for a given level, the sum of the global (fixed) and individual (random) effects. The current implementation does not provide prediction errors.

Usage

```
EB(model, y, X, Z = X, BS = TRUE)
```

Arguments

model	An object of class brokenstick.
y	A vector of new measurements for unit j, scaled in the same metric as the fitted model.
X	A nj * p matrix with fixed effects for unit j, typically produced by bs().
Z	A nj * q matrix with random effects for unit j. The default sets Z equal to X.
BS	A logical indicating whether broken stick estimates should be returned (BS = TRUE) or the random effects (BS = FALSE). The default is TRUE.

Value

A vector of length q containing the random effect or broken stick estimates for unit j.

Author(s)

Stef van Buuren, 2015/2020

References

Skrondal, A., Rabe-Hesketh, S. (2009). Prediction in multilevel generalized linear models. J. R. Statist. Soc. A, 172, 3, 659-687.

Examples

```
#
# EB estimate random effect for child id 10001
data <- smocc_200[smocc_200$id == 10001, ]
y <- data$hgt.z
X <- make_basis(data$age,
  knots = fit_200$knots,
  boundary = fit_200$boundary)
EB(fit_200, y, X)
```

fitted.brokenstick *Calculate fitted values*

Description

Calculate fitted values

Usage

```
## S3 method for class 'brokenstick'
fitted(object, new_data = NULL, ...)
```

Arguments

object	A brokenstick object.
new_data	A data frame or matrix of new predictors.
...	Additional arguments. Ignored.

Value

See predict.brokenstick().

See Also

Other brokenstick: residuals.brokenstick()

fit_200 *Broken stick model with nine lines for 200 children*

Description

The object fit_200 is an object of class brokenstick that contains the fitted broken stick model.

Format

An object of class brokenstick, fitted by the brokenstick().

Details

This dataset was constructed as

```
knots <- round(c(0, 1, 2, 3, 6, 9, 12, 15, 18, 24)/12, 4)
fit_200 <- brokenstick(hgt.z ~ age | id, data = smocc_200,
                       knots = knots, boundary = c(0, 3))
```

get_knots *Obtain the knots from a broken stick model*

Description

Obtain the knots from a broken stick model

Usage

```
get_knots(object, what = c("all", "knots", "boundary", "droplast"), ...)
```

Arguments

object	An object of class brokenstick
what	A character vector of length 1. Valid values are
...	Not used "all", "knots", "boundary" or "droplast". The default is what = "all".

Value

A vector with knot locations, either both, internal only or boundary only. The result is NULL if object does not have proper class. The function can return numeric(0) if there are no internal knots.

Examples

```
get_knots(fit_200, "knots")
```

get_r2 *Obtain proportion of explained variance from a broken stick model*

Description

Obtain proportion of explained variance from a broken stick model

Usage

```
get_r2(object, new_data)
```

Arguments

object	An object of class brokenstick
new_data	Data on which r.squared must be calculated

Value

Proportion of explained variance

Examples

```
get_r2(fit_200, smocc_200)
```

---

kr *Kasim-Raudenbush sampler for two-level normal model*

Description

Simulates posterior distributions of parameters from a two-level normal model with heterogeneous within-cluster variances (Kasim and Raudenbush, 1998). Imputations can be drawn as an extra step to the algorithm.

Usage

```
kr(y, x, g, control, seed, na.action)
```

Arguments

y	Vector with outcome value
x	Matrix with predictor value
g	Vector with group values
control	A list with elements: • model: Correlation model: "argyle", "cole" or "none" • runin: Number of run-in iterations • ndraws: Number of parameter draws • par_skip: Number of iterations to next parameter draw • imp_skip: Number of iterations to next outcome draw
seed	Seed number for base::set.seed(). Use NA to bypass seed setting.
na.action	Not really used here

Details

The calculation time of lme4::lmer() rapidly increases with the number of random effects. More than 10 random effects (knots) takes significant time, and beyond 15 knots generally impossible to fit.

In contrast, the speed of the Kasim-Raudenbush sampler is almost independent of the number of random effect, and foremost depends on the *total number of iterations*: runin + ndraws * par_skip.

The defaults ndraws = 200 and par_skip = 1 provides a *good* approximation to the variance-covariance matrix of the random effects. Increase par_skip to 10 (*better*) or 20 (*best*) to obtain closer approximations at the expense of a linear increase in calculation time. Setting ndraws = 50 (or lower) will reduce computation time, but the result should be treated as *indicative*.

It is possible to subsample the parameter draws at every `imp_skip` iteration, and draw one or more synthetic outcome values from the posterior distribution of the outcome. The number of multiple imputations is equal to `floor(ndraws / imp_skip)`. Thus, setting `imp_skip = 20L` returns 200 / 20 = 10 multiple imputations for each missing value in the outcome. The default does not produce imputations.

Value

A list with components:

```
* `beta`  Fixed effects
* `omega` Variance-covariance of random effects
* `sigma2_j` Residual variance per group
* `sigma2` Average residual variance
* `draws` A matrix with `ndraw` columns with draws for missing data
```

Author(s)

Stef van Buuren, based on `mice::mice.impute.2l.norm()`

References

Kasim RM, Raudenbush SW. (1998). Application of Gibbs sampling to nested variance components models with heterogeneous within-group variance. Journal of Educational and Behavioral Statistics, 23(2), 93–116.

`make_basis` *Create linear splines basis*

Description

This function creates the basis function of a second-order (linear) splines at a user-specific set of break points.

Usage

```
make_basis(
  x,
  knots = NULL,
  boundary = range(x),
  degree = 1L,
  warn = TRUE,
  knotnames = TRUE
)
```

Arguments

x	a data.frame with one column
knots	a vector of internal knots, excluding boundary knots
boundary	vector of external knots
degree	the degree of the spline. The broken stick model requires linear splines, so the default is degree = 1. Setting degree = 0 yields (crisp) dummy coding, and one column less than for degree = 1.
warn	a logical indicating whether warnings from splines::bs() should be given.
knotnames	Should the column names be the knots?

Value

A matrix with length(x) rows and length(breaks) columns, with some extra attributes described by bs().

Note

Before version 0.54, it was standard practice that the knots array always included boundary[1L].

Author(s)

Stef van Buuren, 2020

Examples

```
knots <- c(58, 64, 68, 72)
d1 <- make_basis(data.frame(hgt = women$height), knots = knots)
d0 <- make_basis(data.frame(hgt = women$height), knots = knots, degree = 0)
```

---

parse_formula *Parse formula for brokenstick model*

Description

A bare bones formula parser to extract variables names from formulas of y ~ x | g. It return the name of the first variable mentioned in each formula component.

Usage

```
parse_formula(f)
```

Arguments

f	formula object

Value

A `list` with elements `x`, `y` and `g`. Each element has length 1.

Author(s)

Stef van Buuren, 2020

Examples

```
# examples that yield identical result
parse_formula(y ~ x | z)
parse_formula(y + a ~ x + x1 | z + b)
parse_formula(y + a + log(b) ~ x + x1 * c | z + d)
```

---

plot.brokenstick *Plot observed and fitted trajectories by group*

Description

The `plot` method for a `brokenstick` object plots the observed and fitted trajectories of one or more groups.

Usage

```
## S3 method for class 'brokenstick'
plot(
  x,
  new_data,
  ...,
  what = "droplast",
  .x = NULL,
  group = NULL,
  xlim = NULL,
  ylim = NULL,
  show = c(TRUE, TRUE, FALSE),
  n_plot = 3L
)
```

Arguments

`x`	An object of class `brokenstick`.
`new_data`	A data frame or matrix of new predictors.
`...`	Extra arguments passed down to `predict.brokenstick()` and `plot_trajectory()`.
`what`	Which knots to plot? See `get_knots()`. The default, `what = "droplast"`, does not plot the right boundary knot.
`.x`	The `x` argument of the `predict.brokenstick()` function.

group	A vector with group identifications
xlim	Vector of length 2 with range of x-axis
ylim	Vector of length 2 with range of y-axis
show	A logical vector of length 3. Element 1 specifies whether the observed data are plotted, element 2 specifies whether the broken stick are plotted, element 3 specifies whether imputations are plotted. The default is `c(TRUE,TRUE,FALSE)`.
n_plot	A integer indicating the number of individual plots. The default is 3, which plots the trajectories of the first three groups. The `n_plot` is a safety measure to prevent unintended plots of the entire data set.

Value

An object of class ggplot2::ggplot.

Author(s)

Stef van Buuren 2020

See Also

predict.brokenstick, plot_trajectory.

Examples

```
# fit model on raw hgt with knots at 0, 1, 2 and 3 years
fit1 <- brokenstick(hgt ~ age | id, smocc_200, knots = 0:3)
gp <- c(10001, 10005, 10022)
plot(fit1, smocc_200,
  group = gp, xlim = c(0, 2.1),
  xlab = "Age (years)", ylab = "Length (cm)"
)

# fit model on standard deviation score
fit2 <- brokenstick(hgt.z ~ age | id, smocc_200, knots = 0:3)
plot(fit2, smocc_200,
  group = gp, xlim = c(0, 2.1),
  xlab = "Age (years)", ylab = "Length (SDS)"
)

# model with 11 knots
plot(fit_200, smocc_200,
  group = gp, xlim = c(0, 2.1),
  xlab = "Age (years)", ylab = "Length (SDS)"
)
```

plot_trajectory *Plot observed and fitted trajectories from fitted brokenstick model*

Description

This function is called by `plot.brokenstick`. Normally we wouldn't call `plot_trajectory()` directly.

Usage

```
plot_trajectory(
  x,
  data,
 color_y = c(grDevices::hcl(240, 100, 40, 0.7), grDevices::hcl(240, 100, 40, 0.8)),
  size_y = 2,
 color_yhat = c(grDevices::hcl(0, 100, 40, 0.7), grDevices::hcl(0, 100, 40, 0.8)),
  size_yhat = 2,
  color_imp = c("grey80", "grey80"),
  size_imp = 2,
  ncol = 3L,
  xlab = NULL,
  ylab = NULL,
  xlim = NULL,
  ylim = NULL,
  scales = "fixed",
  theme = ggplot2::theme_light()
)
```

Arguments

x	An object of class `brokenstick`.
data	A `data.frame` with columns names `x$names`, `".source"` and `".pred"`, and possibly `".imp"`
color_y	A character vector with two elements specifying the symbol and line color of the measured data points
size_y	Dot size of measured data points
color_yhat	A character vector with two elements specifying the symbol and line color of the predicted data points
size_yhat	Dot size of predicted data points
color_imp	A character vector with two elements specifying the symbol and line color of the imputed data
size_imp	Dot size of imputed data
ncol	Number of columns in plot
xlab	The label of the x-axis

ylab	The label of the y-axis
xlim	Vector of length 2 with range of x-axis
ylim	Vector of length 2 with range of y-axis
scales	Axis scaling, e.g. "fixed", "free", and so on
theme	Plotting theme

Value

An object of class ggplot

See Also

plot.brokenstick

---

predict.brokenstick *Predict from a* brokenstick *model*

Description

The predictions from a broken stick model coincide with the group-conditional means of the random effects. This function takes an object of class brokenstick and returns predictions in one of several formats. The user can calculate predictions for new persons, i.e., for persons who are not part of the fitted model, through the x and y arguments.

Usage

```
## S3 method for class 'brokenstick'
predict(
  object,
  new_data = NULL,
  type = "numeric",
  ...,
  x = NULL,
  y = NULL,
  group = NULL,
  strip_data = TRUE,
  shape = c("long", "wide", "vector")
)
```

Arguments

object	A brokenstick object.
new_data	A data frame or matrix of new predictors.
type	A single character. The type of predictions to generate. Valid options are: • "numeric" for numeric predictions.

...	Not used, but required for extensibility.
x	Optional. A numeric vector with values of the predictor. It could also be the special keyword x = "knots" replaces x by the positions of the knots.
y	Optional. A numeric vector with measurements.
group	A vector with group identifications
strip_data	A logical indicating whether the row with the observed data from new_data should be stripped from the return. The default is TRUE. Set to FALSE to infer which data points are extracted from new_data.
shape	A string: "long" (default), "wide" or "vector" specifying the shape of the return value. Note that use of "wide" with many unique values in x creates an unwieldy, large and sparse matrix.

Details

By default, predict() calculates predictions for every row in new_data. It is possible to tailor the behavior through the x, y and group arguments. What exactly happens depends on which of these arguments is specified:

1. If the user specifies x, but no y and group, the function returns - for every group in new_data - predictions at x values. This method will use the data from new_data.
2. If the user specifies x and y but no group, the function forms a hypothetical new group with the x and y values. This method uses no information from new_data.
3. If the user specifies group, but no x or y, the function searches for the relevant data in new_data and limits its predictions to the specified groups. This is useful if prediction for only one or a few groups is needed.
4. If the user specifies x and group, but no y, the function will create new values for x in each group, search for the relevant data in new_data and limit prediction to locations x in those groups.
5. If the user specifies x, y and group, the functions assumes that these vectors form a data frame. The lengths of x, y and group must be the same. This procedure uses only information from new_data for groups with group values that match those on newdata.
6. As case 5, but now without a new_data argument. All data are specified through x, y and group. No matching to new_data attempted.

Value

A tibble of predictions. If x, y and group are not specified, the number of rows in the tibble is guaranteed to be the same as the number of rows in new_data.

Examples

```
train <- smocc_200[1:1198, ]
test <- smocc_200[1199:1940, ]

# Fit
fit <- brokenstick(hgt.z ~ age | id, data = train, knots = 0:3)
```

```
# Predict, with preprocessing
tail(predict(fit, test), 3)

# case 1: x as knots
z <- predict(fit, test, x = "knots")

# case 2: x and y, one new group
predict(fit, test, x = "knots", y = c(1, 1, 0.5, 0))

# case 2: x and y, one new group, we need not specify new_data
predict(fit, x = "knots", y = c(1, 1, 0.5, 0))

# case 3: only group
predict(fit, test, group = c(11045, 11120, 999))

# case 4: predict at x in selected groups
predict(fit, test, x = c(0.5, 1, 1.25), group = c(11045, 11120, 999))

# case 5: vectorized
predict(fit, test, x = c(0.5, 1, 1.25), y = c(0, 0.5, 1), group = c(11045, 11120, 999))

# case 6: vectorized, without new_data, results are different for 11045 and 11120
predict(fit, x = c(0.5, 1, 1.25), y = c(0, 0.5, 1), group = c(11045, 11120, 999))
```

`residuals.brokenstick` *Extract residuals from brokenstick model*

Description

Extract residuals from brokenstick model

Usage

```
## S3 method for class 'brokenstick'
residuals(object, new_data = NULL, ...)
```

Arguments

`object`	A `brokenstick` object.
`new_data`	A data frame or matrix of new predictors.
`...`	Additional arguments. Ignored.

Value

See `predict.brokenstick()`.

See Also

Other brokenstick: `fitted.brokenstick()`

smocc_200 *Infant growth of 0-2 years, SMOCC data extract*

Description

Longitudinal height and weight measurements during ages 0-2 years for a representative sample of 1933 Dutch children born in 1988-1989. The dataset `smocc_200` is a subset of the full data covering 200 children.

Format

A tibble with 1940 rows and 7 columns:

id ID, unique `id` of each child (numeric)

age Decimal age, 0-2.12 years (numeric)

sex Sex, `"male"` or `"female"` (character)

ga Gestational age, completed weeks (numeric)

bw Birth weight in grammes (numeric)

hgt Height measurement in cm (34-102) (numeric)

hgt.z Height in SDS relative Fourth Dutch Growth Study 1997 (numeric)

Source

Herngreen WP, van Buuren S, van Wieringen JC, Reerink JD, Verloove-Vanhorick SP & Ruys JH (1994). Growth in length and weight from birth to 2 years of a representative sample of Netherlands children (born in 1988-89) related to socio-economic status and other background characteristics. *Annals of Human Biology*, **21**, 449-463.

weightloss *Weight loss self-measurement data*

Description

Longitudinal weight measurements from 12 individuals with 63 daily measurement under three conditions.

Format

A `data.frame` with 695 rows and 6 columns:

subject ID, consecutive person number 1-12 (integer)

day Measurement day, 0-62 (integer)

sex Sex, 1 = male, 0 = female (integer)

week Week number, 1-9 (integer)

condition Condition (control, diet, activity) (factor)

body_weight Body weight in kg (numeric)

Note

Constructed from file `pone.0232680.s001.csv`. We renumbered `subject` to consecutive integers 1-2 (as in the paper), corrected an error in the `condition` variable for subjects 4 and 12 to match the paper's Figure 4, and filtered the records to the ones woth an observed `body_weight` variable.

Source

Krone T, Boessen R, Bijlsma S, van Stokkum R, Clabbers NDS, Pasman WJ (2020). The possibilities of the use of N-of-1 and do-it-yourself trials in nutritional research. *PloS ONE*, **15**, 5, e0232680.

Index