Package 'npregfast'

July 22, 2025

Type Package

Title Nonparametric Estimation of Regression Models with Factor-by-Curve Interactions

Version 1.5.2

Date 2022-09-01

Maintainer Marta Sestelo <`sestelo@uvigo.es`>

Description A method for obtaining nonparametric estimates of regression models with or without factor-by-curve interactions using local polynomial kernel smoothers or splines. Additionally, a parametric model (allometric model) can be estimated.

BugReports `https://github.com/sestelo/npregfast/issues`

Depends R (>= 3.2.0)

Suggests gridExtra

License MIT + file LICENSE

LazyData true

Imports graphics, stats, utils, shiny, doParallel, foreach, mgcv, sfsmisc, shinyjs, wesanderson, ggplot2

RoxygenNote 7.1.2

Encoding UTF-8

NeedsCompilation yes

Author Marta Sestelo [aut, cre] (ORCID: <`https://orcid.org/0000-0003-4284-6509`>),
Nora M. Villanueva [aut],
Javier Roca-Pardinas [aut]

Repository CRAN

Date/Publication 2022-09-02 12:00:02 UTC

Contents

`allotest` *Bootstrap based test for testing an allometric model*

Description

Bootstrap-based procedure that tests whether the data can be modelled by an allometric model.

Usage

```
allotest(
  formula,
  data,
  na.action = "na.omit",
  nboot = 500,
  seed = NULL,
  cluster = TRUE,
  ncores = NULL,
  test = "res",
  ...
)
```

Arguments

`formula`	An object of class `formula`: a sympbolic description of the model to be fitted.
`data`	An optional data frame, matrix or list required by the formula. If not found in data, the variables are taken from `environment(formula)`, typically the environment from which `allotest` is called.
`na.action`	A function which indicates what should happen when the data contain 'NA's. The default is 'na.omit'.

`nboot`	Number of bootstrap repeats.
`seed`	Seed to be used in the bootstrap procedure.
`cluster`	A logical value. If `TRUE` (default), the bootstrap procedure is parallelized (only for `smooth = "splines"`). Note that there are cases (e.g., a low number of bootstrap repetitions) that R will gain in performance through serial computation. R takes time to distribute tasks across the processors also it will need time for binding them all together later on. Therefore, if the time for distributing and gathering pieces together is greater than the time need for single-thread computing, it does not worth parallelize.
`ncores`	An integer value specifying the number of cores to be used in the parallelized procedure. If `NULL` (default), the number of cores to be used is equal to the number of cores of the machine - 1.
`test`	Statistic test to be used, based on residuals on the null model (`res`) or based on the likelihood ratio test using rss0 and rss1 `lrt`.
`...`	Other options.

Details

In order to facilitate the choice of a model appropriate to the data while at the same time endeavouring to minimise the loss of information, a bootstrap-based procedure, that test whether the data can be modelled by an allometric model, was developed. Therefore, `allotest` tests the null hypothesis of an allometric model taking into account the logarithm of the original variable ($X^* = log(X)$ and $Y^* = log(Y)$).

Based on a general model of the type

$$Y^* = m(X^*) + \varepsilon$$

the aim here is to test the null hypothesis of an allometric model

$$H_0 = m(x^*) = a^* + b^* x^*$$

$vs.$ the general hypothesis H_1, with m being an unknown nonparametric function; or analogously,

$$H_1 : m(x^*) = a^* + b^* x^* + g(x^*)$$

with $g(x^*)$ being an unknown function not equal to zero.

To implement this test we have used the wild bootstrap.

Value

An object is returned with the following elements:

`statistic`	the value of the test statistic.
`value`	the p-value of the test.

Author(s)

Marta Sestelo, Nora M. Villanueva and Javier Roca-Pardinas.

References

Sestelo, M. and Roca-Pardinas, J. (2011). A new approach to estimation of length-weight relationship of *Pollicipes pollicipes* (Gmelin, 1789) on the Atlantic coast of Galicia (Northwest Spain): some aspects of its biology and management. Journal of Shellfish Research, 30 (3), 939–948.

Sestelo, M. (2013). Development and computational implementation of estimation and inference methods in flexible regression models. Applications in Biology, Engineering and Environment. PhD Thesis, Department of Statistics and O.R. University of Vigo.

Examples

```
library(npregfast)
data(barnacle)
allotest(DW ~ RC, data = barnacle, nboot = 50, seed = 130853,
cluster = FALSE)
```

`autoplot.frfast` *Visualization of* `frfast` *objects with ggplot2 graphics*

Description

Useful for drawing the estimated regression function, first and second derivative (for each factor's level) using ggplot2 graphics. Additionally, with the `diffwith` argument it is possible to draw the differences between two factor's levels.

Usage

```
## S3 method for class 'frfast'
autoplot(
  object = model,
  fac = NULL,
  der = 0,
  diffwith = NULL,
  points = TRUE,
  xlab = model$name[2],
  ylab = model$name[1],
  ylim = NULL,
  main = NULL,
  col = "black",
  CIcol = "black",
  CIlinecol = "transparent",
  pcol = "grey80",
  abline = TRUE,
  ablinecol = "red",
  lty = 1,
  CIlty = 2,
  lwd = 1,
```

```
  CIlwd = 1,
  cex = 1.4,
  alpha = 0.2,
  ...
)
```

Arguments

`object`	`frfast` object.
`fac`	Factor's level to be taken into account in the plot. By default is `NULL`.
`der`	Number which determines any inference process. By default `der` is `0`. If this term is `0`, the plot shows the initial estimate. If it is `1` or `2`, it is designed for the first or second derivative, respectively.
`diffwith`	Factor's level used for drawing the differences respect to the level specified in the `fac` argument. By default, `NULL`. The differences are computed for the r-th derivative specified in the `der` argument.
`points`	Draw the original data into the plot. By default it is `TRUE`.
`xlab`	A title for the `x` axis.
`ylab`	A title for the `y` axis.
`ylim`	The `y` limits of the plot.
`main`	An overall title for the plot.
`col`	A specification for the default plotting color.
`CIcol`	A specification for the default confidence intervals plotting color (for the fill).
`CIlinecol`	A specification for the default confidence intervals plotting color (for the edge).
`pcol`	A specification for the points color.
`abline`	Draw an horizontal line into the plot of the second derivative of the model.
`ablinecol`	The color to be used for `abline`.
`lty`	The line type. Line types can either be specified as an integer (0 = blank, 1 = solid (default), 2 = dashed, 3 = dotted, 4 = dotdash, 5 = longdash, 6 = twodash). See details in `par`.
`CIlty`	The line type for confidence intervals. Line types can either be specified as an integer (0 = blank, 1 = solid (default), 2 = dashed, 3 = dotted, 4 = dotdash, 5 = longdash, 6 = twodash).
`lwd`	The line width, a positive number, defaulting to 1. See details in `par`.
`CIlwd`	The line width for confidence intervals, a positive number, defaulting to 1.
`cex`	A numerical value giving the amount by which plotting symbols should be magnified relative to the default. See details in `par`.
`alpha`	Alpha transparency for overlapping elements expressed as a fraction between 0 (complete transparency) and 1 (complete opacity).
`...`	Other options.

Value

A ggplot object, so you can use common features from ggplot2 package to manipulate the plot.

Author(s)

Marta Sestelo, Nora M. Villanueva and Javier Roca-Pardinas.

Examples

```
library(npregfast)
library(ggplot2)


data(barnacle)

# Nonparametric regression without interactions
fit <- frfast(DW ~ RC, data = barnacle, nboot = 50)
autoplot(fit)
autoplot(fit, points = FALSE) + ggtitle("Title")
autoplot(fit, der = 1) + xlim(4, 20)
#autoplot(fit, der = 1, col = "red", CIcol = "blue")

# Nonparametric regression with interactions
fit2 <- frfast(DW ~ RC : F, data = barnacle, nboot = 50)
autoplot(fit2, fac = "barca")
# autoplot(fit2, der = 1, fac = "lens")

# Visualization of the differences between two factor's levels
autoplot(fit2, fac = "barca", diffwith = "lens")
# autoplot(fit2, der = 1, fac = "barca", diffwith = "lens")


#Plotting in the same graphics device
## Not run:

if (requireNamespace("gridExtra", quietly = TRUE)) {

# For plotting two derivatives in the same graphic windows
ders <- lapply(0:1, function(x) autoplot(fit, der = x))
gridExtra::grid.arrange(grobs = ders, ncol = 2, nrow = 1)

# For plotting two levels in the same graphic windows
facs <- lapply(c("barca", "lens"), function(x) autoplot(fit2, der = 0, fac = x))
gridExtra::grid.arrange(grobs = facs, ncol = 2, nrow = 1)

}


## End(Not run)
```

`barnacle` *Barnacle dataset.*

Description

This barnacle data set gives the measurements of the variables dry weight (in g.) and rostro-carinal lenght (in mm) for 2000 barnacles collected along the intertidal zone from two sites of the Atlantic coast of Galicia (Spain).

Usage

```
barnacle
```

Format

`barnacle` is a data frame with 2000 cases (rows) and 3 variables (columns).

DW Dry weight (in g.)

RC Rostro-carinal lenght (in mm).

F Factor indicating the sites of harvest: `barca` and `lens`.

References

Sestelo, M. and Roca-Pardinas, J. (2011). A new approach to estimation of length-weight relationship of $Pollicipes\ pollicipes$ (Gmelin, 1789) on the Atlantic coast of Galicia (Northwest Spain): some aspects of its biology and management. Journal of Shellfish Research, 30(3), 939–948.

Sestelo, M., Villanueva, N.M., Meira-Machado, L., Roca-Pardinas, J. (2017). npregfast: An R Package for Nonparametric Estimation and Inference in Life Sciences. Journal of Statistical Software, 82(12), 1-27.

Examples

```
data(barnacle)
head(barnacle)
```

`children` *Children dataset.*

Description

This children data set contains the age and height measurements of 2500 children aged 5 to 19 years, splitted by sex (1292 females and 1208 males).

Usage

```
children
```

Format

`children` is a data frame with 2500 cases (rows) and 3 variables (columns).

sex Individual's gender (female or male).

height Height measured in centimeters.

age Age in years.

Note

Other data sets of this type can be obtained from `https://www.who.int/toolkits/child-growth-standards`.

Examples

```
data(children)
head(children)
```

`critical` *Critical points of the regression function*

Description

This function draws inference about some critical point in the support of X which is associated with some features of the regression function (e.g., minimum, maximum or inflection points which indicate changes in the sign of curvature). Returns the value of the covariate `x` which maximizes the estimate of the function, the value of the covariate `x` which maximizes the first derivative and the value of the covariate `x` which equals the second derivative to zero, for each level of the factor.

Usage

```
critical(model, der = NULL)
```

Arguments

`model`	Parametric or nonparametric regression out obtained by `frfast` function.
`der`	Number which determines any inference process. By default `der` is `NULL`. If this term is `0`, the calculation is for the point which maximize the estimate. If it is `1` it is designed for the first derivative and if it is `2`, it returns the point which equals the second derivative to zero.

Value

An object is returned with the following elements:

`Estimation`	x value which maximize the regression function with their 95% confidence intervals (for each level).
`First_der`	x value which maximize the first derivative with their 95% confidence intervals (for each level).
`Second_der`	x value which equals the second derivative to zero with their 95% confidence intervals (for each level).

Author(s)

Marta Sestelo, Nora M. Villanueva and Javier Roca-Pardinas.

References

Sestelo, M. (2013). Development and computational implementation of estimation and inference methods in flexible regression models. Applications in Biology, Engineering and Environment. PhD Thesis, Department of Statistics and O.R. University of Vigo.

Sestelo, M., Villanueva, N.M., Meira-Machado, L., Roca-Pardinas, J. (2017). npregfast: An R Package for Nonparametric Estimation and Inference in Life Sciences. Journal of Statistical Software, 82(12), 1-27.

Examples

```
library(npregfast)
data(barnacle)

fit <- frfast(DW ~ RC, data = barnacle) # without interactions
critical(fit)
critical(fit, der = 0)
critical(fit, der = 1)
critical(fit, der = 2)

# fit2 <- frfast(DW ~ RC : F, data = barnacle) # with interactions
# critical(fit2)
# critical(fit2, der = 0)
# critical(fit2, der = 1)
# critical(fit2, der = 2)
```

`criticaldiff` *Differences between the critical points for two factor's levels*

Description

Differences between the estimation of `critical` for two factor's levels.

Usage

```
criticaldiff(model, level1 = NULL, level2 = NULL, der = NULL)
```

Arguments

`model`	Parametric or nonparametric regression model obtained by `frfast` function.
`level1`	First factor's level at which to perform the differences between critical points.
`level2`	Second factor's level at which to perform the differences between critical points.
`der`	Number which determines any inference process. By default `der` is `NULL`. If this term is `0`, the calculate of the differences for the critical point is for the estimate. If it is `1` or `2`, it is designed for the first or second derivative, respectively.

Details

Differences are calculated by subtracting a factor relative to another ($level2 - level1$). By default `level2` and `level1` are `NULL`, so the differences calculated are for all possible combinations between two factors. Additionally, it is obtained the 95% confidence interval for this difference which let us to make inference about them.

Value

An object is returned with the following elements:

`critical.diff`	a table with a couple of factor's level where it is used to calculate the differences between the critical points, and their 95% confidence interval (for the estimation, first and second derivative).

Author(s)

Marta Sestelo, Nora M. Villanueva and Javier Roca-Pardinas.

References

Sestelo, M. (2013). Development and computational implementation of estimation and inference methods in flexible regression models. Applications in Biology, Engineering and Environment. PhD Thesis, Department of Statistics and O.R. University of Vigo.

Sestelo, M., Villanueva, N.M., Meira-Machado, L., Roca-Pardinas, J. (2017). npregfast: An R Package for Nonparametric Estimation and Inference in Life Sciences. Journal of Statistical Software, 82(12), 1-27.

Examples

```
library(npregfast)
data(barnacle)
fit2 <- frfast(DW ~ RC : F, data = barnacle, seed = 130853, nboot = 100) # with interactions
criticaldiff(fit2)
criticaldiff(fit2, der = 1)
criticaldiff(fit2, der = 1, level1 = "lens", level2 = "barca")
```

`frfast` *Fitting nonparametric models*

Description

This function is used to fit nonparametric models by using local polynomial kernel smoothers or splines. These models can include or not factor-by-curve interactions. Additionally, a parametric model (allometric model) can be estimated (or not).

Usage

```
frfast(
  formula,
  data,
  na.action = "na.omit",
  model = "np",
  smooth = "kernel",
  h0 = -1,
  h = -1,
  nh = 30,
  weights = NULL,
  kernel = "epanech",
  p = 3,
  kbin = 100,
  nboot = 500,
  rankl = NULL,
  ranku = NULL,
  seed = NULL,
  cluster = TRUE,
  ncores = NULL,
  ...
)
```

Arguments

`formula`	An object of class `formula`: a sympbolic description of the model to be fitted. The details of model specification are given under 'Details'.
`data`	An optional data frame, matrix or list required by the formula. If not found in data, the variables are taken from `environment(formula)`, typically the environment from which `frfast` is called.
`na.action`	A function which indicates what should happen when the data contain 'NA's. The default is 'na.omit'.
`model`	Type model used: `model = "np"` for a nonparametric regression model, `model = "allo"` for an allometric model. See details.
`smooth`	Type smoother used: `smooth = "kernel"` for local polynomial kernel smoothers and `smooth = "splines"` for splines using the `mgcv` package.

`h0`	The kernel bandwidth smoothing parameter for the global effect (see references for more details at the estimation). Large values of the bandwidth lead to smoothed estimates; smaller values of the bandwidth lead lo undersmoothed estimates. By default, cross validation is used to obtain the bandwidth.
`h`	The kernel bandwidth smoothing parameter for the partial effects.
`nh`	Integer number of equally-spaced bandwidth in which the `h` is discretised, to speed up computation in the kernel-based regression.
`weights`	Prior weights on the data.
`kernel`	A character string specifying the desired kernel. Defaults to `kernel = "epanech"`, where the Epanechnikov density function kernel will be used. Also, several types of kernel functons can be used: triangular and Gaussian density function, with `"triang"` and `"gaussian"` term, respectively.
`p`	Polynomial degree to be used in the kernel-based regression. Its value must be the value of derivative + 1. The default value is 3, returning the estimation, first and second derivative.
`kbin`	Number of binning nodes over which the function is to be estimated.
`nboot`	Number of bootstrap repeats. Defaults to 500 bootstrap repeats. The wild bootstrap is used when `model = "np"` and the simple bootstrap when `model = "allo"`.
`rankl`	Number or vector specifying the minimum value for the interval at which to search the `x` value which maximizes the estimate, first or second derivative (for each level). The default is the minimum data value.
`ranku`	Number or vector specifying the maximum value for the interval at which to search the `x` value which maximizes the estimate, first or second derivative (for each level). The default is the maximum data value.
`seed`	Seed to be used in the bootstrap procedure.
`cluster`	A logical value. If `TRUE` (default), the bootstrap procedure is parallelized (only for `smooth = "splines"`). Note that there are cases (e.g., a low number of bootstrap repetitions) that R will gain in performance through serial computation. R takes time to distribute tasks across the processors also it will need time for binding them all together later on. Therefore, if the time for distributing and gathering pieces together is greater than the time need for single-thread computing, it does not worth parallelize.
`ncores`	An integer value specifying the number of cores to be used in the parallelized procedure. If `NULL` (default), the number of cores to be used is equal to the number of cores of the machine - 1.
`...`	Other options.

Details

The models fitted by `frfast` function are specified in a compact symbolic form. The ~ operator is basic in the formation of such models. An expression of the form `y ~ model` is interpreted as a specification that the response `y` is modelled by a predictor specified symbolically by `model`. The possible terms consist of a variable name or a variable name and a factor name separated by : operator. Such a term is interpreted as the interaction of the continuous variable and the factor.

However, if `smooth = "splines"`, the formula is based on the function formula.gam of the mgcv package.

According with the `model` argument, if `model = "np"` the estimated regression model will be of the type

$$Y = m(X) + e$$

being m an smooth and unknown function and e the regression error with zero mean. If `model = "allo"`, users could estimate the classical allometric model (Huxley, 1924) with a regression curve

$$m(X) = aX^b$$

being a and b the parameters of the model.

Value

An object is returned with the following elements:

`x`	Vector of values of the grid points at which model is to be estimate.
`p`	Matrix of values of the grid points at which to compute the estimate, their first and second derivative.
`pl`	Lower values of 95% confidence interval for the estimate, their first and second derivative.
`pu`	Upper values of 95% confidence interval for the estimate, their first and second derivative.
`diff`	Differences between the estimation values of a couple of levels (i. e. level 2 - level 1). The same procedure for their first and second derivative.
`diffl`	Lower values of 95% confidence interval for the differences between the estimation values of a couple of levels. It is performed for their first and second derivative.
`diffu`	Upper values of 95% confidence interval for the differences between the estimation values of a couple of levels. It is performed for their first and second derivative.
`nboot`	Number of bootstrap repeats.
`n`	Sample size.
`dp`	Degree of polynomial to be used.
`h0`	The kernel bandwidth smoothing parameter for the global effect.
`h`	The kernel bandwidth smoothing parameter for the partial effects.
`fmod`	Factor's level for each data.
`xdata`	Original x values.
`ydata`	Original y values.
`w`	Weights on the data.
`kbin`	Number of binning nodes over which the function is to be estimated.
`nf`	Number of levels.

`max`	Value of covariate `x` which maximizes the estimate, first or second derivative.
`maxu`	Upper value of 95% confidence interval for the value `max`.
`maxl`	Lower value of 95% confidence interval for the value `max`.
`diffmax`	Differences between the estimation of `max` for a couple of levels (i. e. level 2 - level 1). The same procedure for their first and second derivative.
`diffmaxu`	Upper value of 95% confidence interval for the value `diffmax`.
`diffmaxl`	Lower value of 95% confidence interval for the value `diffmax`.
`repboot`	Matrix of values of the grid points at which to compute the estimate, their first and second derivative for each bootstrap repeat.
`rankl`	Maximum value for the interval at which to search the `x` value which maximizes the estimate, first or second derivative (for each level). The default is the maximum data value.
`ranku`	Minimum value for the interval at which to search the `x` value which maximizes the estimate, first or second derivative (for each level). The default is the minimum data value.
`nmodel`	Type model used: `nmodel = 1` the nonparametric model, `nmodel = 2` the allometric model.
`label`	Labels of the variables in the model.
`numlabel`	Number of labels.
`kernel`	A character specifying the derised kernel.
`a`	Estimated coefficient in the case of fitting an allometric model.
`al`	Lower value of 95% confidence interval for the value of `a`.
`au`	Upper value of 95% confidence interval for the value of `a`.
`b`	Estimated coefficient in the case of fitting an allometric model.
`bl`	Lower value of 95% confidence interval for the value of `b`.
`bu`	Upper value of 95% confidence interval for the value of `b`.
`name`	Name of the variables in the model.
`formula`	A sympbolic description of the model to be fitted.
`nh`	Integer number of equally-spaced bandwidth on which the `h` is discretised.
`r2`	Coefficient of determination (in the case of the allometric model).
`smooth`	Type smoother used.
`cluster`	Is the procedure parallelized? (for splines smoothers).
`ncores`	Number of cores used in the parallelized procedure? (for splines smoothers).

Author(s)

Marta Sestelo, Nora M. Villanueva and Javier Roca-Pardinas.

References

Huxley, J. S. (1924). Constant differential growth-ratios and their significance. Nature, 114:895–896.

Sestelo, M. (2013). Development and computational implementation of estimation and inference methods in flexible regression models. Applications in Biology, Engineering and Environment. PhD Thesis, Department of Statistics and O.R. University of Vigo.

Sestelo, M., Villanueva, N.M., Meira-Machado, L., Roca-Pardinas, J. (2017). npregfast: An R Package for Nonparametric Estimation and Inference in Life Sciences. Journal of Statistical Software, 82(12), 1-27.

Examples

```
library(npregfast)
data(barnacle)

# Nonparametric regression without interactions
fit <- frfast(DW ~ RC, data = barnacle, nboot = 100, smooth = "kernel")
fit
summary(fit)

# using  splines
#fit <- frfast(DW ~ s(RC), data = barnacle, nboot = 100,
#smooth = "splines", cluster = TRUE, ncores = 2)
#fit
#summary(fit)


# Change the number of binning nodes and bootstrap replicates
fit <- frfast(DW ~ RC, data = barnacle, kbin = 200,
               nboot = 100, smooth = "kernel")

# Nonparametric regression with interactions
fit2 <- frfast(DW ~ RC : F, data = barnacle, nboot = 100)
fit2
summary(fit2)

# using  splines
#fit2 <- frfast(DW ~ s(RC, by = F), data = barnacle,
#               nboot = 100, smooth = "splines", cluster = TRUE, ncores = 2)
#fit2
#summary(fit2)


# Allometric model
fit3 <- frfast(DW ~ RC, data = barnacle, model = "allo", nboot = 100)
summary(fit3)

# fit4 <- frfast(DW ~ RC : F, data = barnacle, model = "allo", nboot = 100)
# summary(fit4)
```

`globaltest` *Testing the equality of the* M *curves specific to each level*

Description

This function can be used to test the equality of the M curves specific to each level.

Usage

```
globaltest(
  formula,
  data,
  na.action = "na.omit",
  der,
  smooth = "kernel",
  weights = NULL,
  nboot = 500,
  h0 = -1,
  h = -1,
  nh = 30,
  kernel = "epanech",
  p = 3,
  kbin = 100,
  seed = NULL,
  cluster = TRUE,
  ncores = NULL,
  ...
)
```

Arguments

`formula`	An object of class `formula`: a sympbolic description of the model to be fitted. The details of model specification are given under 'Details'.
`data`	An optional data frame, matrix or list required by the formula. If not found in data, the variables are taken from `environment(formula)`, typically the environment from which `globaltest` is called.
`na.action`	A function which indicates what should happen when the data contain 'NA's. The default is 'na.omit'.
`der`	Number which determines any inference process. By default `der` is `NULL`. If this term is `0`, the testing procedures is applied for the estimate. If it is `1` or `2`, it is designed for the first or second derivative, respectively.
`smooth`	Type smoother used: `smooth = "kernel"` for local polynomial kernel smoothers and `smooth = "splines"` for splines using the `mgcv` package.
`weights`	Prior weights on the data.
`nboot`	Number of bootstrap repeats.

`h0`	The kernel bandwidth smoothing parameter for the global effect (see references for more details at the estimation). Large values of the bandwidth lead to smoothed estimates; smaller values of the bandwidth lead lo undersmoothed estimates. By default, cross validation is used to obtain the bandwidth.
`h`	The kernel bandwidth smoothing parameter for the partial effects.
`nh`	Integer number of equally-spaced bandwidth on which the `h` is discretised, to speed up computation.
`kernel`	A character string specifying the desired kernel. Defaults to `kernel = "epanech"`, where the Epanechnikov density function kernel will be used. Also, several types of kernel funcitons can be used: triangular and Gaussian density function, with `"triang"` and `"gaussian"` term, respectively.
`p`	Degree of polynomial to be used. Its value must be the value of derivative + 1. The default value is 3 due to the function returns the estimation, first and second derivative.
`kbin`	Number of binning nodes over which the function is to be estimated.
`seed`	Seed to be used in the bootstrap procedure.
`cluster`	A logical value. If `TRUE` (default), the bootstrap procedure is parallelized (only for `smooth = "splines"`. Note that there are cases (e.g., a low number of bootstrap repetitions) that R will gain in performance through serial computation. R takes time to distribute tasks across the processors also it will need time for binding them all together later on. Therefore, if the time for distributing and gathering pieces together is greater than the time need for single-thread computing, it does not worth parallelize.
`ncores`	An integer value specifying the number of cores to be used in the parallelized procedure. If `NULL` (default), the number of cores to be used is equal to the number of cores of the machine - 1.
`...`	Other options.

Details

`globaltest` can be used to test the equality of the M curves specific to each level. This bootstrap based test assumes the following null hypothesis:

$$H_0^r : m_1^r(\cdot) = \ldots = m_M^r(\cdot)$$

versus the general alternative

$$H_1^r : m_i^r(\cdot) \neq m_j^r(\cdot) \quad \text{for} \quad \text{some} \quad \text{i,j} \in \{1,\ldots,\text{M}\}.$$

Note that, if H_0 is not rejected, then the equality of critical points will also accepted.

To test the null hypothesis, it is used a test statistic, T, based on direct nonparametric estimates of the curves.

If the null hypothesis is true, the T value should be close to zero but is generally greater. The test rule based on T consists of rejecting the null hypothesis if $T > T^{1-\alpha}$, where T^p is the empirical

p-percentile of T under the null hypothesis. To obtain this percentile, we have used bootstrap techniques. See details in references.

Note that the models fitted by `globaltest` function are specified in a compact symbolic form. The ~ operator is basic in the formation of such models. An expression of the form `y ~ model` is interpreted as a specification that the response `y` is modelled by a predictor specified symbolically by `model`. The possible terms consist of a variable name or a variable name and a factor name separated by : operator. Such a term is interpreted as the interaction of the continuous variable and the factor. However, if `smooth = "splines"`, the formula is based on the function formula.gam of the mgcv package.

Value

The T value and the p-value are returned. Additionally, it is shown the decision, accepted or rejected, of the global test. The null hypothesis is rejected if the p-value< 0.05.

Author(s)

Marta Sestelo, Nora M. Villanueva and Javier Roca-Pardinas.

References

Sestelo, M. (2013). Development and computational implementation of estimation and inference methods in flexible regression models. Applications in Biology, Engineering and Environment. PhD Thesis, Department of Statistics and O.R. University of Vigo.

Sestelo, M., Villanueva, N.M., Meira-Machado, L., Roca-Pardinas, J. (2017). npregfast: An R Package for Nonparametric Estimation and Inference in Life Sciences. Journal of Statistical Software, 82(12), 1-27.

Examples

```
library(npregfast)
data(barnacle)
globaltest(DW ~ RC : F, data = barnacle, der = 1, seed = 130853, nboot = 100)

# globaltest(height ~ s(age, by = sex), data = children,
# seed = 130853, der = 0, smooth = "splines")
```

`localtest` *Testing the equality of critical points*

Description

This function can be used to test the equality of the M critical points estimated from the respective level-specific curves.

Usage

```
localtest(
  formula,
  data = data,
  na.action = "na.omit",
  der,
  smooth = "kernel",
  weights = NULL,
  nboot = 500,
  h0 = -1,
  h = -1,
  nh = 30,
  kernel = "epanech",
  p = 3,
  kbin = 100,
  rankl = NULL,
  ranku = NULL,
  seed = NULL,
  cluster = TRUE,
  ncores = NULL,
  ci.level = 0.95,
  ...
)
```

Arguments

`formula`	An object of class `formula`: a sympbolic description of the model to be fitted. The details of model specification are given under 'Details'.
`data`	An optional data frame, matrix or list required by the formula. If not found in data, the variables are taken from `environment(formula)`, typically the environment from which `localtest` is called.
`na.action`	A function which indicates what should happen when the data contain 'NA's. The default is 'na.omit'.
`der`	Number which determines any inference process. By default `der` is `NULL`. If this term is `0`, the testing procedures is applied for the estimate. If it is `1` or `2`, it is designed for the first or second derivative, respectively.
`smooth`	Type smoother used: `smooth = "kernel"` for local polynomial kernel smoothers and `smooth = "splines"` for splines using the `mgcv` package.
`weights`	Prior weights on the data.
`nboot`	Number of bootstrap repeats.
`h0`	The kernel bandwidth smoothing parameter for the global effect (see references for more details at the estimation). Large values of the bandwidth lead to smoothed estimates; smaller values of the bandwidth lead lo undersmoothed estimates. By default, cross validation is used to obtain the bandwidth.
`h`	The kernel bandwidth smoothing parameter for the partial effects.

`nh`	Integer number of equally-spaced bandwidth on which the `h` is discretised, to speed up computation.
`kernel`	A character string specifying the desired kernel. Defaults to `kernel = "epanech"`, where the Epanechnikov density function kernel will be used. Also, several types of kernel funcitons can be used: triangular and Gaussian density function, with `"triang"` and `"gaussian"` term, respectively.
`p`	Degree of polynomial to be used. Its value must be the value of derivative + 1. The default value is 3 due to the function returns the estimation, first and second derivative.
`kbin`	Number of binning nodes over which the function is to be estimated.
`rankl`	Number or vector specifying the minimum value for the interval at which to search the `x` value which maximizes the estimate, first or second derivative (for each level). The default is the minimum data value.
`ranku`	Number or vector specifying the maximum value for the interval at which to search the `x` value which maximizes the estimate, first or second derivative (for each level). The default is the maximum data value.
`seed`	Seed to be used in the bootstrap procedure.
`cluster`	A logical value. If `TRUE` (default), the bootstrap procedure is parallelized (only for `smooth = "splines"`. Note that there are cases (e.g., a low number of bootstrap repetitions) that R will gain in performance through serial computation. R takes time to distribute tasks across the processors also it will need time for binding them all together later on. Therefore, if the time for distributing and gathering pieces together is greater than the time need for single-thread computing, it does not worth parallelize.
`ncores`	An integer value specifying the number of cores to be used in the parallelized procedure. If `NULL` (default), the number of cores to be used is equal to the number of cores of the machine - 1.
`ci.level`	Level of bootstrap confidence interval. Defaults to 0.95 (corresponding to 95%). Note that the function accepts a vector of levels.
`...`	Other options.

Details

`localtest` can be used to test the equality of the M critical points estimated from the respective level-specific curves. Note that, even if the curves and/or their derivatives are different, it is possible for these points to be equal.

For instance, taking the maxima of the first derivatives into account, interest lies in testing the following null hypothesis

$$H_0 : x_{01} = \ldots = x_{0M}$$

versus the general alternative

$$H_1 : x_{0i} \neq x_{0j} \quad \text{for} \quad \text{some} \quad i, j \in \{1, \ldots, M\}.$$

The above hypothesis is true if $d = x_{0j} - x_{0k} = 0$ where

$$(j,k) = argmax \quad (l,m) \quad \{1 \le l < m \le M\} \quad |x_{0l} - x_{0m}|,$$

otherwise H_0 is false. It is important to highlight that, in practice, the true x_{0j} are not known, and consequently neither is d, so an estimate $\hat{d} = \hat{x}_{0j} - \hat{x}_{0k}$ is used, where, in general, $\hat{x}_{0l}$ are the estimates of x_{0l} based on the estimated curves $\hat{m}_l$ with $l = 1, \ldots, M$.

Needless to say, since $\hat{d}$ is only an estimate of the true d, the sampling uncertainty of these estimates needs to be acknowledged. Hence, a confidence interval (a, b) is created for d for a specific level of confidence (95%). Based on this, the null hypothesis is rejected if zero is not contained in the interval.

Note that if this hypothesis is rejected (and the factor has more than two levels), one option could be to use the `maxp.diff` function in order to obtain the differences between each pair of factor's levels.

Note that the models fitted by `localtest` function are specified in a compact symbolic form. The ~ operator is basic in the formation of such models. An expression of the form `y ~ model` is interpreted as a specification that the response `y` is modelled by a predictor specified symbolically by `model`. The possible terms consist of a variable name or a variable name and a factor name separated by : operator. Such a term is interpreted as the interaction of the continuous variable and the factor. However, if `smooth = "splines"`, the formula is based on the function formula.gam of the mgcv package.

Value

The estimate of d value is returned and its confidence interval for a specific-level of confidence, i.e. 95%. Additionally, it is shown the decision, accepted or rejected, of the local test. Based on the null hypothesis is rejected if a zero value is not within the interval.

Author(s)

Marta Sestelo, Nora M. Villanueva and Javier Roca-Pardinas.

References

Sestelo, M. (2013). Development and computational implementation of estimation and inference methods in flexible regression models. Applications in Biology, Engineering and Environment. PhD Thesis, Department of Statistics and O.R. University of Vigo.

Sestelo, M., Villanueva, N.M., Meira-Machado, L., Roca-Pardinas, J. (2017). npregfast: An R Package for Nonparametric Estimation and Inference in Life Sciences. Journal of Statistical Software, 82(12), 1-27.

Examples

```
library(npregfast)
data(barnacle)
localtest(DW ~ RC : F, data = barnacle, der = 1, seed = 130853, nboot = 100)

# localtest(height ~ s(age, by = sex), data = children, seed = 130853,
# der = 1, smooth = "splines")
```

`plot.frfast` *Visualization of* `frfast` *objects with the base graphics*

Description

Useful for drawing the estimated regression function, first and second derivative (for each factor's level). Additionally, with the `diffwith` argument it is possible to draw the differences between two factor's levels.

Usage

```
## S3 method for class 'frfast'
plot(
  x = model,
  y,
  fac = NULL,
  der = NULL,
  diffwith = NULL,
  points = TRUE,
  xlab = model$name[2],
  ylab = model$name[1],
  ylim = NULL,
  main = NULL,
  col = "black",
  CIcol = "black",
  pcol = "grey80",
  ablinecol = "red",
  abline = TRUE,
  type = "l",
  CItype = "l",
  lwd = 2,
  CIlwd = 1,
  lty = 1,
  CIlty = 2,
  cex = 0.6,
  ...
)
```

Arguments

`x`	`frfast` object.
`y`	NULL.
`fac`	Vector which determines the level to take into account in the plot. By default is `NULL`.
`der`	Number or vector which determines any inference process. By default `der` is `NULL`. If this term is `0`, the plot shows the initial estimate. If it is `1` or `2`, it is designed for the first or second derivative, respectively.

`diffwith`	Factor’s level used for drawing the differences respect to the level specified in the `fac` argument. By default, `NULL`. The differences are computed for the r-th derivative speciefied in the `der` argument.
`points`	Draw the original data into the plot. By default it is `TRUE`.
`xlab`	A title for the `x` axis.
`ylab`	A title for the `y` axis.
`ylim`	The `y` limits of the plot.
`main`	An overall title for the plot.
`col`	A specification for the default plotting color.
`CIcol`	A specification for the default confidence intervals plotting color.
`pcol`	A specification for the points color.
`ablinecol`	The color to be used for `abline`.
`abline`	Draw an horizontal line into the plot of the second derivative of the model.
`type`	What type of plot should be drawn. Possible types are, `p` for points, `l` for lines, `o` for overplotted, etc. See details in `par`.
`CItype`	What type of plot should be drawn for confidence intervals. Possible types are, `p` for points, `l` for lines, `o` for overplotted.
`lwd`	The line width, a positive number, defaulting to 1. See details in `par`.
`CIlwd`	The line width for confidence intervals, a positive number, defaulting to 1.
`lty`	The line type. Line types can either be specified as an integer (0 = blank, 1 = solid (default), 2 = dashed, 3 = dotted, 4 = dotdash, 5 = longdash, 6 = twodash). See details in `par`.
`CIlty`	The line type for confidence intervals. Line types can either be specified as an integer (0 = blank, 1 = solid (default), 2 = dashed, 3 = dotted, 4 = dotdash, 5 = longdash, 6 = twodash).
`cex`	A numerical value giving the amount by which plotting symbols should be magnified relative to the default. See details in `par`.
`...`	Other options.

Value

Simply produce a plot.

Author(s)

Marta Sestelo, Nora M. Villanueva and Javier Roca-Pardinas.

Examples

```
library(npregfast)
data(barnacle)

# Nonparametric regression without interactions
fit <- frfast(DW ~ RC, data = barnacle, nboot = 100)
```

```
plot(fit)
plot(fit, der = 0)
plot(fit, der = 0, points = FALSE)
plot(fit, der = 1, col = "red", CIcol = "blue")

# Nonparametric regression with interactions
fit2 <- frfast(DW ~ RC : F, data = barnacle, nboot = 100)
plot(fit2)
plot(fit2, der = 0, fac = "lens")
plot(fit2, der = 1, col = "grey", CIcol = "red")
plot(fit2, der = c(0,1), fac = c("barca","lens"))

# Visualization of the differences between two factor's levels
plot(fit2, fac = "barca", diffwith = "lens")
plot(fit2, fac = "barca", diffwith = "lens", der = 1)
```

`predict.frfast` *Prediction from fitted* `frfast` *model*

Description

Takes a fitted `frfast` object and produces predictions (with their 95% confidence intervals) from a fitted model with interactions or without interactions.

Usage

```
## S3 method for class 'frfast'
predict(object = model, newdata, fac = NULL, der = NULL, seed = NULL, ...)
```

Arguments

`object`	A fitted `frfast` object as produced by `frfast()`.
`newdata`	A data frame containing the values of the model covariates at which predictions are required. If newdata is provided, then it should contain all the variables needed for prediction: a warning is generated if not.
`fac`	Factor's level to take into account. By default is `NULL`.
`der`	Number which determines any inference process. By default `der` is `NULL`. If this term is `0`, the function returns the initial estimate. If it is `1` or `2`, it is designed for the first or second derivative, respectively.
`seed`	Seed to be used in the bootstrap procedure.
`...`	Other options.

Value

`predict.frfast` computes and returns a list containing predictions of the estimates, first and second derivative, with their 95% confidence intervals.

Author(s)

Marta Sestelo, Nora M. Villanueva and Javier Roca-Pardinas.

Examples

```
library(npregfast)
data(barnacle)

# Nonparametric regression without interactions
fit <- frfast(DW ~ RC, data = barnacle, nboot = 100)
nd <- data.frame(RC = c(10, 14, 18))
predict(fit, newdata = nd)

# Nonparametric regression with interactions
# fit2 <- frfast(DW ~ RC : F, data = barnacle, nboot = 100)
# nd2 <- data.frame(RC = c(10, 15, 20))
# predict(fit2, newdata = nd2)
# predict(fit2, newdata = nd2, der = 0, fac = "barca")
```

`runExample` *Run npregfast example*

Description

Launch a Shiny app that shows a demo of what can be done with the package.

Usage

```
runExample()
```

Details

This example is also available online.

Examples

```
## Only run this example in interactive R sessions
if (interactive()) {
  runExample()
}
```

`summary.frfast` *Summarizing fits of* `frfast` *class*

Description

Takes a fitted `frfast` object produced by `frfast()` and produces various useful summaries from it.

Usage

```
## S3 method for class 'frfast'
summary(object = model, ...)
```

Arguments

`object`	a fitted `frfast` object as produced by `frfast()`.
`...`	additional arguments affecting the predictions produced.

Details

`print.frfast` tries to be smart about `summary.frfast`.

Value

`summary.frfast` computes and returns a list of summary information for a fitted `frfast` object.

`model`	type of model: nonparametric or allometric.
`smooth`	type of smoother: kernel or splines.
`h`	the kernel bandwidth smoothing parameter.
`dp`	degree of the polynomial.
`nboot`	number of bootstrap repeats.
`kbin`	number of binning nodes over which the function is to be estimated.
`n`	sample size.
`fmod`	factor’s levels.
`coef`	if `model = "allo"`, coefficients of the model.

Author(s)

Marta Sestelo, Nora M. Villanueva and Javier Roca-Pardinas.

References

Sestelo, M. (2013). Development and computational implementation of estimation and inference methods in flexible regression models. Applications in Biology, Engineering and Environment. PhD Thesis, Department of Statistics and O.R. University of Vigo.

Sestelo, M., Villanueva, N.M., Meira-Machado, L., Roca-Pardinas, J. (2017). npregfast: An R Package for Nonparametric Estimation and Inference in Life Sciences. Journal of Statistical Software, 82(12), 1-27.

Examples

```
library(npregfast)
data(barnacle)

# Nonparametric regression without interactions
fit <- frfast(DW ~ RC, data = barnacle, nboot = 100)
fit
summary(fit)

# Nonparametric regression with interactions
fit2 <- frfast(DW ~ RC : F, data = barnacle, nboot = 100)
fit2
summary(fit2)

# Allometric model
fit3 <- frfast(DW ~ RC, data = barnacle, model = "allo", nboot = 100)
fit3
summary(fit3)
```

Index