Package 'gamlss'

January 2, 2012

Description The library for fitting GAMLSS models.

Title Generalized Additive Models for Location Scale and Shape.

LazyLoad yes

Version 4.1-1

Date 2011-11-07

Depends R (>= 2.4.0), graphics, stats, splines, utils, gamlss.dist,gamlss.data, nlme

Imports MASS, survival

Author Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk> with contributions from Calliope Akantziliotou

Maintainer Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

License GPL-2 | GPL-3

URL http://www.gamlss.org/

Repository CRAN

Date/Publication 2011-11-08 06:54:42

R topics documented:

`gamlss-package` *The GAMLSS library and datasets*

Description

This a collection of functions to fit Generalized Additive Models for Location Scale and Shape(GAMLSS) and handled gamlss objects.

GAMLSS were introduced by Rigby and Stasinopoulos (2005). GAMLSS is a general framework for univariate regression type statistical problems using new ways of dealing with overdispersion, skewness and kurtosis in the response variable. In GAMLSS the exponential family distribution assumption used in Generalized Linear Model (GLM) and Generalized Additive Model (GAM),(see Nelder and Wedderburn, 1972 and Hastie and Tibshirani, 1990, respectively) is relaxed and replaced by a very general distribution family including highly skew and kurtotic discrete and continuous distributions. The systematic part of the model is expanded to allow modelling not only the mean (or location) but other parameters of the distribution of the response variable as linear parametric, nonlinear parametric or additive non-parametric functions of explanatory variables and/or random effects terms. Maximum (penalized) likelihood estimation is used to fit the models.

Details

Package:	gamlss
Type:	Package
Version:	1.5-0
Date:	2006-12-13
License:	GPL (version 2 or later) See file LICENSE

This package allow the user to model the distribution of the response variable using a variety of one, two, three and four parameter families of distributions. The distributions implemented currently can be found in `gamlss.family`. Other distributions can be easily added. In the current implementation of GAMLSS several additive terms have been implemented including regression splines, smoothing splines, penalized splines, varying coefficients, fractional polynomials and random effects. Other additive terms can be easily added.

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk> with contributions from Calliope Akantziliotou.

Maintainer: Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References

Nelder, J. A. and Wedderburn, R. W. M. (1972). Generalized linear models. *J. R. Statist. Soc.* A., **135** 370-384.

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized Additive Models*. Chapman and Hall, London.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

Examples

```
data(abdom)
mod<-gamlss(y~pb(x),sigma.fo=~pb(x),family=BCT, data=abdom, method=mixed(1,20))
plot(mod)
rm(mod)
```

`additive.fit` *Implementing Backfitting in GAMLSS*

Description

This function is not to be used on its own. It is used for backfitting in the GAMLSS fitting algorithms and it is based on the equivalent function written by Trevor Hastie in the gam() S-plus implementation, (Chambers and Hastie, 1991).

Usage

```
additive.fit(x, y, w, s, who, smooth.frame, maxit = 30, tol = 0.001,
             trace = FALSE, se = TRUE, ...)
```

Arguments

`x`	the linear part of the explanatory variables
`y`	the response variable
`w`	the weights
`s`	the matrix containing the smoothers
`who`	the current smoothers
`smooth.frame`	the data frame used for the smoothers
`maxit`	maximum number of iterations in the backfitting
`tol`	the tolerance level for the backfitting
`trace`	whether to trace the backfitting algorithm
`se`	whether standard errors are required
`...`	for extra arguments

Details

This function should not be used on its own

Value

Returns a list with the linear fit plus the smothers

Author(s)

Mikis Stasinopoulos

References

Chambers, J. M. and Hastie, T. J. (1991). *Statistical Models in S*, Chapman and Hall, London.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss

---

bfp *Functions to fit fractional polynomials in GAMLSS*

Description

The function bfp generate a power polynomial basis matrix which (for given powers) can be used to fit power polynomials in one x-variable. The function fp takes a vector and returns it with several attributes. The vector is used in the construction of the model matrix. The function fp() is not used for fitting the fractional polynomial curves but assigns the attributes to the vector to aid gamlss in the fitting process. The function doing the fitting is gamlss.fp() which is used at the backfitting function additive.fit (but never used on its own). The (experimental) function pp can be use to fit power polynomials as in $a + b_1 x^{p_1} + b_2 x^{p_2}$., where p1 and p2 have arbitrary values rather restricted as in the fp function.

Usage

```
bfp(x, powers = c(1, 2), shift = NULL, scale = NULL)
fp(x, npoly = 2, shift = NULL, scale = NULL)
pp(x, start = list(), shift = NULL, scale = NULL)
```

Arguments

x	the explanatory variable to be used in functions bfp() or fp(). Note that this is different from the argument x use in gamlss.fp (a function used in the backfitting but not by straight by the user)
powers	a vector containing as elements the powers in which the x has to be raised
shift	a number for shifting the x-variable. The default values is zero, if x is positive, or the minimum of the positive difference in x minus the minimum of x
scale	a positive number for scalling the x-variable. The default values is $10^(sign(log10(range)))*trunc(abs(log10(range)))$
npoly	a positive indicating how many fractional polynomials should be considered in the fit. Can take the values 1, 2 or 3 with 2 as default
start	a list containing the starting values for the non-linear maximization to find the powers. The results from fitting the equivalent fractional polynomials can be used here

Details

The above functions are an implementation of the fractional polynomials introduced by Royston and Altman (1994). The three functions involved in the fitting are loosely based on the fractional polynomials implementation in S-plus written by Gareth Amber. The function bfp generates the right design matrix for the fitting a power polynomial of the type $a + b_1 x^{p_1} + b_2 x^{p_2} + \ldots + b_k x_k^p$. For given powers $p_1, p_2, \ldots, p_k$ given as the argument powers in bfp() the function can be used to fit power polynomials in the same way as the functions poly() or bs() (of package splines) are used to fit orthogonal or piecewise polynomials respectively. The function fp(), which is working as a smoother in gamlss, is used to fit the best fractional polynomials within a set of power values. Its argument npoly determines whether one, two or three fractional polynomials should used in the fitting. For a fixed number npoly the algorithm looks for the best fitting fractional polynomials in the list c(-2, -1, -0.5, 0, 0.5, 1, 2, 3) . Note that npolu=3 is rather slow since it fits all possible combinations 3-way combinations at each backfitting interaction. The function gamlss.fp() is an internal function of GAMLSS allowing the fractional polynomials to be fitted in the backfitting cycle of gamlss, and should be not used on its own.

Value

The function bfp returns a matrix to be used as part of the design matrix in the fitting.

The function fp returns a vector with values zero to be included in the design matrix but with attributes useful in the fitting of the fractional polynomials algorithm in gamlss.fp.

Warning

Since the model constant is included in both the design matrix X and in the backfitting part of fractional polynomials, its values is wrongly given in the summary. Its true values is the model constant minus the constant from the fractional polynomial fitting ??? What happens if more that one fractional polynomials are fitted?

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <b.rigby@londonmet.ac.uk>

References

Amber G. (1999) Fracial polynomials in S-plus, http://lib.stat.cmu.edu/S/fracpoly.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Royston, P. and Altman, D. G., (1994). Regression using fractional polynomials of continuous covariates: parsimonious parametric modelling (with discussion), *Appl. Statist.*, **43**, 429-467.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, gamlss.family

Examples

```
data(abdom)
#fits polynomials with power 1 and .5
mod1<-gamlss(y~bfp(x,c(1,0.5)),data=abdom)
# fit the best of one fractional polynomial
m1<-gamlss(y~fp(x,1),data=abdom)
# fit the best of two fractional polynomials
m2<-gamlss(y~fp(x,2),data=abdom)
# fit the best of three fractional polynomials
m3<-gamlss(y~fp(x,3),data=abdom)
# get the coefficient for the second model
m2$mu.coefSmo
# now power polynomials using the best 2 fp c()
 m4 <- gamlss(y ~ pp(x, c(1,3)), data = abdom)
# This is not good idea in this case because
# if you look at the fitted values you see what it went wrong
plot(y~x,data=abdom)
lines(fitted(m2,"mu")~abdom$x,col="red")
lines(fitted(m4,"mu")~abdom$x,col="blue")
```

`centiles` *Plots the centile curves for a GAMLSS object*

Description

This function `centiles()` plots centiles curves for distributions belonging to the GAMLSS family of distributions. The function also tabulates the sample percentages below each centile curve (for comparison with the model percentages given by the argument `cent`.) The function `centiles.fan()` plots a fan-chart of the centile curves. A restriction of the functions is that it applies to models with one explanatory variable only.

Usage

```
centiles(obj, xvar = NULL, cent = c(0.4, 2, 10, 25, 50, 75, 90, 98, 99.6),
         legend = TRUE, ylab = "y", xlab = "x", main = NULL,
         main.gsub = "@", xleg = min(xvar), yleg = max(obj$y),
         xlim = range(xvar), ylim = range(obj$y), save = FALSE,
         plot = TRUE, points = TRUE, pch = "+", col = "blue",
       col.centiles = 1:length(cent) + 2, lty.centiles = 1, lwd.centiles = 1, ...)
centiles.fan(obj, xvar = NULL, cent = c(0.4, 2, 10, 25, 50, 75, 90, 98, 99.6),
         ylab = "y", xlab = "x", main = NULL, main.gsub = "@",
         xleg = min(xvar), yleg = max(obj$y), xlim = range(xvar),
         ylim = range(obj$y), points = FALSE,  median = TRUE,  pch = "+",
         col = "blue",
         colors = c("cm", "gray", "rainbow", "heat", "terrain", "topo"), ...)
```

Arguments

obj	a fitted gamlss object from fitting a gamlss distribution
xvar	the unique explanatory variable
cent	a vector with elements the % centile values for which the centile curves have to be evaluated
legend	whether a legend is required in the plot or not, the default is legent=TRUE
ylab	the y-variable label
xlab	the x-variable label
main	the main title here as character. If NULL the default title "centile curves using NO" (or the relevant distributions name) is shown
main.gsub	if the main.gsub (with default "@") appears in the main title then it is substituted with the default title.
xleg	position of the legend in the x-axis
yleg	position of the legend in the y-axis
xlim	the limits of the x-axis
ylim	the limits of the y-axis
save	whether to save the sample percentages or not with default equal to FALSE. In this case the sample percentages are printed but are not saved
plot	whether to plot the centiles. This option is useful for centile.split
pch	the character to be used as the default in plotting points see par
col	plotting colour see par
col.centiles	Plotting colours for the centile curves
lty.centiles	line type for the centile curves
lwd.centiles	The line width for the centile curves
colors	the different colour schemes to be used for the fan-chart. The following are available c("cm","gray", "rainbow", "heat", "terrain", "topo"),

points	whether the data points should be plotted, default is TRUE for centiles() and FALSE for centiles.fan()
median	whether the median should be plotted (only in centiles.fan())
...	for extra arguments

Details

Centiles are calculated using the fitted values in obj and xvar must correspond exactly to the predictor in obj to plot correctly.

col.centiles, lty.centiles and lwd.centiles may be vector arguments and are recycled to the length cent if necessary.

Value

A centile plot is produced and the sample centiles below each centile curve are printed (or saved)

Warning

This function is appropriate only when one continuous explanatory variable is fitted in the model

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk> with contribution from Steve Ellison

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, centiles.split , centiles.com

Examples

```
data(abdom)
h<-gamlss(y~pb(x), sigma.formula=~pb(x), family=BCT, data=abdom)
# default plot
centiles(h,xvar=abdom$x)
# control of colours and lines
centiles(h, xvar=abdom$x,  col.cent=c(2,3,4,5,1,5,4,3,2,1),
             lwd.cent=c(1,1,1,1,2,1,1,1,1))
```

```
#Control line types
centiles(h, xvar=abdom$x,  col.cent=1, cent=c(.5,2.5,50,97.5,99.5),
             lty.centiles=c(3,2,1,2,3),lwd.cent=c(1,1,2,1,1))
# control of the main title
centiles(h, xvar=abdom$x,  main="Abdominal data \n @")
# the fan-chart
centiles.fan(h,xvar=abdom$x, colors="rainbow")
rm(h)
```

`centiles.com` *Comparing centiles from different GAMLSS models*

Description

This function compares centiles curves for more than one GAMLSS objects.It is based on the `centiles` function. The function also tabulates the sample percentages below each centile curve (for comparison with the model percentages given by the argument `cent`.) A restriction of the function is that it applies to models with one explanatory variable only

Usage

```
centiles.com(obj, ..., xvar = NULL, cent = c(0.4, 10, 50, 90, 99.6),
             legend = TRUE, ylab = "y", xlab = "x", xleg = min(xvar),
             yleg = max(obj$y), xlim = range(xvar), ylim = NULL,
             no.data = FALSE, color = TRUE, main = NULL, plot = TRUE)
```

Arguments

`obj`	a fitted gamlss object from fitting a gamlss continuous distribution
`...`	optionally more fitted GAMLSS model objects
`xvar`	the unique explanatory variable
`cent`	a vector with elements the % centile values for which the centile curves have to be evaluated
`legend`	whether a legend is required in the plot or not, the default is `legent=TRUE`
`ylab`	the y-variable label
`xlab`	the x-variable label
`xleg`	position of the legend in the x-axis
`yleg`	position of the legend in the y-axis
`xlim`	the limits of the x-axis
`ylim`	the limits of the y-axis
`no.data`	whether the data should plotted, default `no.data=FALSE` or not `no.data=TRUE`
`color`	whether the fitted centiles are shown in colour, `color=TRUE` (the default) or not `color=FALSE`
`main`	the main title
`plot`	whether to plot the centiles

Value

Centile plots are produced for the different fitted models and the sample centiles below each centile curve are printed

Warning

This function is appropriate only when one continuous explanatory variable is fitted in the model

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk> and Bob Rigby <r.rigby@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M.(2005). Generalized additive models for location, scale and shape, (with discussion),*Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

See Also

gamlss, centiles , centiles.split

Examples

```
data(abdom)
h1<-gamlss(y~cs(x,df=3), sigma.formula=~cs(x,1),family=BCT, data=abdom)
h2<-gamlss(y~lo(x,span=0.4), sigma.formula=~lo(x,span=0.4), family=BCT, data=abdom )
centiles.com(h1,h2,xvar=abdom$x)
rm(h1,h2)
```

---

centiles.pred *Creating predictive centiles values*

Description

This function creates predictive centiles curves for new x-values given a GAMLSS fitted model. The function has three options: i) for given new x-values and given percentage centiles calculates a matrix containing the centiles values for y, ii) for given new x-values and standard normalized centile values calculates a matrix containing the centiles values for y, iii) for given new x-values and new y-values calculates the z-scores. A restriction of the function is that it applies to models with only one explanatory variable.

Usage

```
centiles.pred(obj, type = c("centiles", "z-scores", "standard-centiles"),
              xname = NULL, xvalues = NULL, power = NULL, yval = NULL,
              cent = c(0.4, 2, 10, 25, 50, 75, 90, 98, 99.6),
              dev = c(-4, -3, -2, -1, 0, 1, 2, 3, 4),
              plot = FALSE, legend = TRUE,
              ...)
```

Arguments

obj	a fitted gamlss object from fitting a gamlss continuous distribution
type	the default, "centiles", gets the centiles values given in the option cent. type="standard-centiles" gets the standard centiles given in the dev. type="z-scores" gets the z-scores for given y and x new values
xname	the name of the unique explanatory variable (it has to be the same as in the original fitted model)
xvalues	the new values for the explanatory variable where the prediction will take place
power	if power transformation is needed (but read the note below)
yval	the response values for a given x required for the calculation of "z-scores"
cent	a vector with elements the % centile values for which the centile curves have to be evaluated
dev	a vector with elements the standard normalized values for which the centile curves have to be evaluated in the option type="standard-centiles"
plot	whether to plot the "centiles" or the "standard-centiles", the default is plot=FALSE
legend	whether a legend is required in the plot or not, the default is legent=TRUE
...	for extra arguments

Value

a vector (for option type="z-scores") or a matrix for options type="centiles" or type="standard-centiles" containing the appropriate values

Warning

See example below of how to use the function when power transofrmation is used for the x-variables

Note

The power option should be only used if the model

Author(s)

Mikis Stasinopoulos , <d.stasinopoulos@londonmet.ac.uk>, based on ideas of Elaine Borghie from the World Health Organization

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, centiles, centiles.split

Examples

```
# bring the data and fit the model
data(abdom)
a<-gamlss(y~pb(x),sigma.fo=~pb(x), data=abdom, family=BCT)
#plot the centiles
centiles(a,xvar=abdom$x)
# calculate the centiles at new x values
newx<-seq(12,40,2)
mat <- centiles.pred(a, xname="x", xvalues=newx )
mat
# now plot the centiles
 mat <- centiles.pred(a, xname="x",xvalues=newx, plot=TRUE )
# calculate standard-centiles for new x values using the fitted model
newx <- seq(12,40,2)
 mat <- centiles.pred(a, xname="x",xvalues=newx, type="standard-centiles" )
 mat
# now plot the centiles
mat <- centiles.pred(a, xname="x",xvalues=newx, type="s", plot = TRUE )
# create new y and x values and plot them in the previous plot
newx <- c(20,21.2,23,20.9,24.2,24.1,25)
newy <- c(130,121,123,125,140,145,150)
for(i in 1:7) points(newx[i],newy[i],col="blue")
# now calculate their z-scores
znewx <- centiles.pred(a, xname="x",xvalues=newx,yval=newy, type="z-scores" )
znewx
# now with transformed x-variable within the formula
aa<-gamlss(y~pb(x^0.5),sigma.fo=~pb(x^0.5), data=abdom, family=BCT)
centiles(aa,xvar=abdom$x)
mat <- centiles.pred(aa, xname="x",xvalues=c(30) )
xx<-rep(mat[,1],9)
yy<-mat[,2:10]
points(xx,yy,col="red")
# now with x-variable previously transformed
nx<-abdom$x^0.5
aa<-gamlss(y~pb(nx),sigma.fo=~pb(nx), data=abdom, family=BCT)
centiles(aa, xvar=abdom$x)
```

```
newd<-data.frame( abdom, nx=abdom$x^0.5)
mat <-  centiles.pred(aa, xname="nx", xvalues=c(30), power=0.5, data=newd)
xxx<-rep(mat[,1],9)
yyy<-mat[,2:10]
points(xxx,yyy,col="red")
```

`centiles.split` *Plots centile curves split by x for a GAMLSS object*

Description

This function plots centiles curves for separate ranges of the unique explanatory variable x. It is similar to the `centiles` function but the range of x is split at a user defined values `xcut.point` into r separate ranges. The functions also tabulates the sample percentages below each centile curve for each of the r ranges of x (for comparison with the model percentage given by cent) The model should have only one explanatory variable.

Usage

```
centiles.split(obj, xvar = NULL, xcut.points = NULL, n.inter = 4,
               cent = c(0.4, 2, 10, 25, 50, 75, 90, 98, 99.6),
               legend = FALSE, main = NULL, main.gsub = "@",
               ylab = "y", xlab = "x", ylim = NULL, overlap = 0,
               save = TRUE, plot = TRUE, ...)
```

Arguments

`obj`	a fitted gamlss object from fitting a gamlss continuous distribution
`xvar`	the unique explanatory variable
`xcut.points`	the x-axis cut off points e.g. `c(20,30)`. If `xcut.points=NULL` then the `n.inter` argument is activated
`n.inter`	if `xcut.points=NULL` this argument gives the number of intervals in which the x-variable will be splited, with default 4
`cent`	a vector with elements the % centile values for which the centile curves are to be evaluated
`legend`	whether a legend is required in the plots or not, the default is `legent=FALSE`
`main`	the main title as character. If NULL the default title (shown the intervals) is shown
`main.gsub`	if the `main.gsub` (with default "@") appears in the `main` title then it is substituted with the default title.
`ylab`	the y-variable label
`xlab`	the x-variable label
`ylim`	the range of the y-variable axis

overlap	how much overlapping in the xvar intervals. Default value is overlap=0 for non overlapping intervals
save	whether to save the sample percentages or not with default equal to TRUE. In this case the functions produce a matrix giving the sample percentages for each interval
plot	whether to plot the centles. This option is usefull if the sample statistics only are to be used
...	for extra arguments

Value

Centile plots are produced and the sample centiles below each centile curve for each of the r ranges of x can be saved into a matrix.

Warning

This function is appropriate when only one continuous explanatory variable is fitted in the model

Author(s)

Mikis Stasinopoulos, <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>, with contributions from Elaine Borghie

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2003) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss centiles, centiles.com

Examples

```
data(abdom)
h<-gamlss(y~pb(x), sigma.formula=~pb(x), family=BCT, data=abdom)
mout <- centiles.split(h,xvar=abdom$x)
mout
rm(h,mout)
```

coef.gamlss *Extract Model Coefficients in a GAMLSS fitted model*

Description

coef.gamlss is the GAMLSS specific method for the generic function coef which extracts model coefficients from objects returned by modelling functions. 'coefficients' is an alias for coef.

Usage

```
## S3 method for class 'gamlss'
coef(object, what = c("mu", "sigma", "nu", "tau"), ... )
```

Arguments

object	a GAMLSS fitted model
what	which parameter coefficient is required, default what="mu"
...	for extra arguments

Value

Coefficients extracted from the GAMLSS model object.

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References

RRigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, deviance.gamlss, fitted.gamlss

Examples

```
data(aids)
h<-gamlss(y~poly(x,3)+qrt, family=PO, data=aids) #
coef(h)
rm(h)
```

cs *Specify a Smoothing Cubic Spline Fit in a GAMLSS Formula*

Description

The functions cs() scs() and vc are using cubic smoothing splines. They take a vector and return it with several attributes. The vector is used in the construction of the model matrix. The functions do no smoothing, but assigns the attributes to the vector to aid gamlss in the smoothing. The function doing the smoothing are gamlss.cs(), gamlss.scs() and gamlss.vc(). The gamlss.cs() is a modified version of the R function smooth.spline() which is used by the backfitting function additive.fit. The function gamlss.scs() just uses smooth.spline(). It differs from the function cs() in that allows cross valdation of the smoothing parameters. The (experimental) function vc can be use to fit varying coefficient models, see Hastie and Tibshirani (1993).

Usage

```
cs(x, df = 3, spar = NULL, c.spar = NULL)
vc(r, x, df = 3, spar = NULL, c.spar = NULL)
scs(x, df = NULL, spar = NULL, control.spar = NULL, all.knots = TRUE, nknots = NULL, penalty =1.4)
```

Arguments

x	the univariate predictor, (or expression, that evaluates to a numeric vector). For the function vc the x argument is the vector which has its (linear) coefficient change with r
df	the desired equivalent number of degrees of freedom (trace of the smoother matrix minus two for the constant and linear fit). The real smoothing parameter (spar below) is found such that df=tr(S)-2, where S is the implicit smoother matrix. Values for df should be greater than 0, with 0 implying a linear fit.
spar	smoothing parameter, typically (but not necessarily) in (0,1]. The coefficient lambda of the integral of the squared second derivative in the fit (penalized log likelihood) criterion is a monotone function of 'spar', see the details in smooth.spline.
c.spar	This is an option to be used when the degrees of freedom of the fitted gamlss object are different from the ones given as input in the option df. The default values used are the ones given the option control.spar in the R function smooth.spine() and they are c.spar=c(-1.5, 2). For very large data sets e.g. 10000 observations, the upper limit may have to increase for example to c.spar=c(-1.5, 2.5). Use this option if you have received the warning 'The output df are different from the input, change the control.spar'. c.spar can take both vectors or lists of length 2, for example c.spar=c(-1.5, 2.5) or c.spar=list(-1.5, 2.5) would have the same effect.
r	for the function vc, r represent the vector of the explanatory variable which effects the coefficients of x i.e. beta(r)*x. Both the x and r vectors should be adjusted by subtracting the their mean

control.spar	see above c.spar or the equivalent argument in the function smooth.spline.
all.knots	if TRUE, all distinct points in x are used as knots. If FALSE (default), a subset of x is used, see the same argument for function smooth.spline
nknots	integer giving the number of knots to use when all.knots=FALSE. Per default, this is less than n, the number of unique x values for n > 49
penalty	the penalty applied to cross validation

Details

Note that cs itself does no smoothing; it simply sets things up for the function gamlss() which in turn uses the function additive.fit() for backfitting which in turn uses gamlss.cs()

Note that cs() and s() functions behave differently at their default values that is if df and lambda are not specified. cs(x) by default will use 3 extra degrees of freedom for smoothing for x. ps(x) by default will estimate lambda (and the degrees of freedom) automatically using generalised cross validation (GCV). Note that if GCV is used the convergence of the gamlss model can be less stable compared to a model where the degrees of freedom are fixed. This can be true especially for small data sets.

Value

the vector x is returned, endowed with a number of attributes. The vector itself is used in the construction of the model matrix, while the attributes are needed for the backfitting algorithms additive.fit(). Since smoothing splines includes linear fits, the linear part will be efficiently computed with the other parametric linear parts of the model.

Warning

For a user who wishes to compare the gamlss() results with the equivalent gam() results in S-plus: make sure when using S-plus that the convergence criteria epsilon and bf.epsilon in control.gam() are decreased sufficiently to ensure proper convergence in S-plus. Also note that the degrees of freedom are defined on top of the linear term in gamlss, but on top of the constant term in S-plus, (so use an extra degrees of freedom in S-plus in order to obtain comparable results to those in galmss).

Change the upper limit of spar if you received the warning 'The output df are different from the input, change the control.spar'.

For large data sets do not use expressions, e.g. cs(x^0.5) inside the gamlss function command but evaluate the expression, e.g. nx=$x^0.5$, first and then use cs(nx).

Note

The degrees of freedom df are defined differently from that of the gam() function in S-plus. Here df are the additional degrees of freedom excluding the constant and the linear part of x. For example df=4 in gamlss() is equivalent to df=5 in gam() in S-plus

Author(s)

Mikis Stasinopoulos and Bob Rigby

References

Hastie, T. J. and Tibshirani, R. J. (1993), Varying coefficient models (with discussion),J. R. Statist. Soc. B., **55**, 757-796.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, gamlss.cs, lo

Examples

```
# cubic splines example
data(aids)
# fitting a smoothing cubic spline with 7 degrees of freedom
# plus the a quarterly  effect
aids1<-gamlss(y~cs(x,df=7)+qrt,data=aids,family=PO) #
aids2<-gamlss(y~scs(x,df=5)+qrt,data=aids,family=PO) #
aids3<-gamlss(y~scs(x)+qrt,data=aids,family=PO) # using GCV
with(aids, plot(x,y))
lines(aids$x,fitted(aids1), col="red")
lines(aids$x,fitted(aids3), col="green")
rm(aids1, aids2, aids3)
# varying-coefficient example
data(rent)
attach(rent)
# adjusting the variables
Flbar<-Fl-mean(Fl)
Abar<-A-mean(A)
# additive model
 m1<-gamlss(R~cs(Flbar, df=3)+cs(Abar))
# varying-coefficient model
 m2<-gamlss(R~cs(Flbar, df=3)+cs(Abar)+vc(r=Abar,x=Flbar))
AIC(m1,m2)
detach(rent)
```

---

deviance.gamlss *Global Deviance of a GAMLSS model*

Description

Returns the global, -2*log(likelihood), or the penalized, -2*log(likelihood)+ penalties, deviance of a fitted GAMLSS model object.

Usage

```
## S3 method for class 'gamlss'
deviance(object,  what = c("G", "P"), ...)
```

Arguments

object	a GAMLSS fitted model
what	put "G" for Global or "P" for Penalized deviance
...	for extra arguments

Details

deviance is a generic function which can be used to extract deviances for fitted models. deviance.gamlss is the method for a GAMLSS object.

Value

The value of the global or the penalized deviance extracted from a GAMLSS object.

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss.family, coef.gamlss, fitted.gamlss

Examples

```
data(aids)
h<-gamlss(y~poly(x,3)+qrt, family=PO, data=aids) #
deviance(h)
rm(h)
```

dtop *Detrended transformed Owen's plot*

Description

Provides single or multiple detrended transformed Owen's plot, Owen (1995), for a GAMLSS fitted objects or any other fitted object which has the method resid(). This is a diagnostic tool for checking whether the normalised quantile residuals are coming from a normal distribution or not. This could be true if the horizontal line is within the confidence intervals.

Usage

```
dtop(object = NULL, xvar = NULL, resid = NULL,
      conf.level = c("95", "99"), n.inter = 4,
      xcut.points = NULL, overlap = 0,
      show.given = TRUE, cex = 1, pch = 21,
      line = TRUE, ...)
```

Arguments

object	a GAMLSS fitted object or any other fitted object which has the method resid().
xvar	the explanatory variable against which the detrended Owen's plots will be plotted
resid	if the object is not specified the residual vector can be given here
conf.level	95 (default) or 99 percent confidence interval for the plots
n.inter	he number of intervals in which the explanatory variable xvar will be cut
xcut.points	the x-axis cut off points e.g. c(20,30). If xcut.points=NULL then the n.inter argument is activated
overlap	how much overlapping in the xvar intervals. Default value is overlap=0 for non overlapping intervals
show.given	whether to show the x-variable intervals in the top of the graph, default is show.given=TRUE
cex	the cex plotting parameter with default cex=1
pch	the pch plotting parameter with default pch=21
line	whether the detrended empirical cdf should be plotted or not
...	for extra arguments

Details

If the xvar argument is not specified then a single detrended Owen's plot is used, see Owen (1995). In this case the plot is a detrended nonparametric likelihood confidence band for a distribution function. That is, if the horizontal lines lies within the confidence band then the normalised residuals could have come from a Normal distribution and consequently the assumed response variable distribution is reasonable. If the xvar is specified then we have as many plots as n.iter. In this case the

x-variable is cut into n.iter intervals with an equal number observations and detrended Owen's plots for each interval are plotted. This is a way of highlighting failures of the model within different ranges of the explanatory variable.

Value

A plot is returned.

Author(s)

Mikis Stasinopoulos, Bob Rigby and Vlassios Voudouris

References

Owen A. B. (1995) Nonparametric Confidence Bands for a Distribution Function. Journal of the American Statistical Association Vol. 90, No 430, pp. 516-521.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), Appl. Statist., 54, part 3, 1-38.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. Journal of Statistical Software, Vol. 23, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

wp

Examples

```
data(abdom)
a<-gamlss(y~pb(x),sigma.fo=~pb(x,1),family=LO,data=abdom)
dtop(a)
dtop(a, xvar=abdom$x)
rm(a)
```

---

edf *Effective degrees of freedom from gamlss model*

Description

The functions `edf()` and `edfAll()` can be used to obtained the effective degrees of freedom for different additive terms for the distribution parameters in a gamlss model.

Usage

```
edf(obj, what = c("mu", "sigma", "nu", "tau"), print = TRUE, ...)
edfAll(obj, ...)
```

Arguments

obj	A gamlss fitted model
what	which of the four parameters mu, sigma, nu or tau.
print	whether to print the label
...	for extra arguments

Value

The function edfAll() re turns a list of edf for all the fitted parameters. The function edf() a vector of edf.

Note

The edf given are the ones fitted in the backfitting so the usually contained (depending on the additive term) the contatnt and the linear part.

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss

Examples

```
library(gamlss.data)
data(usair)
m1<- gamlss(y~pb(x1)+pb(x2)+pb(x3)+pb(x6), data=usair)
edfAll(m1)
edf(m1)
```

find.hyper *A function to select values of hyperparameters in a GAMLSS model*

Description

This function selects the values of hyper parameters and/or non-linear parameters in a GAMLSS model. It uses the R function optim which then minimised the generalized Akaike information criterion (GAIC) with a user defined penalty.

Usage

```
find.hyper(model = NULL, parameters = NULL, other = NULL, penalty = 2,
        steps = c(0.1), lower = -Inf, upper = Inf, method = "L-BFGS-B",
        ...)
```

Arguments

model	this is a GAMLSS model. e.g. model=gamlss(y~cs(x,df=p[1]),sigma.fo=~cs(x,df=p[2]),data=abdom) where p[1] and p[2] denore the parametets to be estimated
parameters	the starting values in the search of the optimum hyperparameters and/or non-linear parameters e.g. parameters=c(3,3)
other	this is used to optimize other non-parameters, for example a transformation of the explanatory variable of the kind $x^{p[3]}$, others=quote(nx<-x^p[3]) where nx is now in the model formula
penalty	specifies the penalty in the GAIC, (the default is 2 e.g. penalty=3
steps	the steps taken in the optimization procedure [see the ndeps option in optim()], by default is set to 0.1 for all hyper parameters and non-linear parameters
lower	the lower permissible level of the parameters i.e. lower=c(1,1) this does not apply if a method other than the default method "L-BFGS-B" is used
upper	the upper permissible level of the parameters i.e. upper=c(30,10), this is not apply if a method other than the default method "L-BFGS-B" is used
method	the method used in optim() to numerically minimize the GAIC over the hyperparameters and/or non-linear parameters. By default this is "L-BFGS-B" to allow box-restriction on the parameters
...	for extra arguments to be passed to the R function optim() used in the optimization

Details

This is an experimental function which appears to work well for the search of the optimum degrees of freedom and non-linear parameters (e.g. power parameter λ used to transform x to x^{λ}). Further investigation will check whether this function is reliable in general.

Value

The function turns the same output as the function `optim()`

`par`	the optimum hyperparameter values
`value`	the minimized value of the GAIC
`counts`	A two-element integer vector giving the number of calls to 'fn' and 'gr' respectively
`convergence`	An integer code. '0' indicates successful convergence. see the function `optim()` for other errors
`message`	A character string giving any additional information returned by the optimizer, or 'NULL'

Warning

It may be slow to find the optimum

Author(s)

Mikis Stasinopoulos

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, plot.gamlss, optim

Examples

```
data(abdom)
attach(abdom)
# declare the model
mod1<-quote(gamlss(y~cs(nx,df=p[1]),family=BCT,data=abdom,
                        control=gamlss.control(trace=FALSE)))
# we want also to check for a transformation in x
# so we use the other option
op<-find.hyper(model=mod1, other=quote(nx<-x^p[2]), parameters=c(3,0.5),
            lower=c(1,0.001), steps=c(0.1,0.001))
# the optimum parameters found are
# p=(p[1],p[2]) = (2.944836 0.001000) = (df for mu, lambda)
```

```
# so it needs df = 3 on top of the constant and linear
# in  the cubic spline model for mu since p[1] is approximately  3
# and log transformation for x since p[2] is approximately  0
op
rm(op)
```

`fitDist` *Fits Different Parametric* `gamlss.family` *distributions to data*

Description

This function is using the function `gamlssML()` to fit all relevant parametric `gamlss.family` distributions to a data vector. The final model is the one which is selected by the generalised Akaike information criterion with penalty `k`.

Usage

```
fitDist(y, k = 2,
      type = c("realAll", "realline", "realplus", "real0to1", "counts", "binom"),
  try.gamlss = FALSE, extra = NULL, data = NULL)
```

Arguments

`y`	the data vector
`k`	the penalty for the GAIC with default values `k=2` the standard AIC
`type`	the tipe of distribution to be tried see details
`try.gamlss`	if `gamlssML()` failed whether should try `gamlss` instead
`extra`	whether extra distribution should be tried which are not in the `type` list
`data`	the data frame where `y` ca be found

Details

The following are the different `type` argument:

- `realAll` all the `gamlss.family` continuous distributions defined on the real line, i.e. `realline` plus `realplus`
- `reallinethe` `gamlss.family` continuous distributions : "GU", "RG" ,"LO", "NET", "TF", "PE", "SHASH", "EGB2", "JSU", "SEP1", "SEP2", "SEP3", "SEP4","ST1", "ST2", "ST3", "ST4", "ST5", "GT"
- `realplus` the `gamlss.family` continuous distributions in the positive leal line: "EXP","GA","IG","LNO", "WEI3", "BCCGo", "exGAUS", "GG", "GIG", "BCTo", "BCPEo"
- `real0to1`the `gamlss.family` continuous distributions from 0 to 1: "BE", "BEINF", "BEINF0", "BEINF1", "BEOI", "BEZI", "GB1"
- `countsthe` `gamlss.family` distributions for counts: "PO", "LG", "NBI", "NBII", "PIG", "DEL", "SI", "ZIP", "ZAP", "ZALG", "ZANBI", "ZIP2", "ZIPIG"
- `binomthe` `gamlss.family` distributions for binomial type data :"BI", "BB", "ZIBI", "ZIBB", "ZABI", "ZABB"

Value

A gamlssML object with two extra components:

fits	an ordered list according to the GAIC of the fitted distributioon
failed	the distributions where the gamlssML)() (or gamlss()) fits have failed

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>, Vlasis Voudouris <v.voiudouris@londonmet.ac.uk> and Majid Djennad <m.djennad.londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss,gamlssML

Examples

```
y <- rt(100, df=1)
m1<-fitDist(y, type="realline")
m1$fits
m1$failed
# an example of using  extra
## Not run:
library(gamlss.tr)
data(tensile)
gen.trun(par=1,family="GA", type="right")
gen.trun(par=1,"LOGNO", type="right")
gen.trun(par=c(0,1),"TF", type="both")
ma<-fitDist(str, type="real0to1", extra=c("GAtr", "LOGNOtr", "TFtr"), data=tensile)

## End(Not run)
```

`fitted.gamlss` *Extract Fitted Values For A GAMLSS Model*

Description

`fitted.gamlss` is the GAMLSS specific method for the generic function `fitted` which extracts fitted values for a specified parameter from a GAMLSS objects. `fitted.values` is an alias for it. The function `fv()` is similar to `fitted.gamlls()` but allows the argument `what` not to be character

Usage

```
## S3 method for class 'gamlss'
fitted(object, what = c("mu", "sigma", "nu", "tau"), ...)
fv(obj, what = "mu", ...)
```

Arguments

`object`	a GAMLSS fitted model
`obj`	a GAMLSS fitted model
`what`	which parameter fitted values are required, default `what="mu"`
`...`	for extra arguments

Value

Fitted values extracted from the GAMLSS object for the given parameter.

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

`print.gamlss`, `summary.gamlss`, `fitted.gamlss`, `coef.gamlss`, `residuals.gamlss`, `update.gamlss`, `plot.gamlss`, `deviance.gamlss`, `formula.gamlss`

Examples

```
data(aids)
h<-gamlss(y~poly(x,3)+qrt, family=PO, data=aids) #
fitted(h)
rm(h)
```

`fittedPlot` *Plots The Fitted Values of a GAMLSS Model*

Description

This function, applicable only to a models with a single explanatory variable, plots the fitted values for all the parameters of a GAMLSS model against the (one) explanatory variable. It is also useful for comparing the fits for more than one model.

Usage

```
fittedPlot(object, ..., x = NULL, color = TRUE, line.type = FALSE, xlab = NULL)
```

Arguments

`object`	a fitted GAMLSS model object(with only one explanatory variable)
`...`	optionally more fitted GAMLSS model objects
`x`	The unique explanatory variable
`color`	whether the fitted lines plots are shown in colour, `color=TRUE` (the default) or not `color=FALSE`
`line.type`	whether the line type should be different or not. The default is `color=FALSE`
`xlab`	the x-label

Value

A plot of the fitted values against the explanatory variable

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk> and Calliope Akantziliotou

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, centiles, centiles.split

Examples

```
data(abdom)
h1<-gamlss(y~pb(x), sigma.formula=~x, family=BCT, data=abdom)
h2<-gamlss(y~pb(x), sigma.formula=~pb(x), family=BCT, data=abdom)
fittedPlot(h1,h2,x=abdom$x)
rm(h1,h2)
```

---

formula.gamlss *Extract the Model Formula in a GAMLSS fitted model*

Description

formula.gamlss is the GAMLSS specific method for the generic function formula which extracts the model formula from objects returned by modelling functions.

Usage

```
## S3 method for class 'gamlss'
formula(x, what = c("mu", "sigma", "nu", "tau"), ... )
```

Arguments

x	a GAMLSS fitted model
what	which parameter coefficient is required, default what="mu"
...	for extra arguments

Value

Returns a model formula

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, deviance.gamlss, fitted.gamlss

Examples

```
data(aids)
h<-gamlss(y~poly(x,3)+qrt, family=PO, data=aids) #
formula(h,"mu")
rm(h)
```

---

gamlss *Generalized Additive Models for Location Scale and Shape*

Description

Returns an object of class "gamlss", which is a generalized additive model for location scale and shape (GAMLSS). The function gamlss() is very similar to the gam() function in S-plus (now also in R in package gam), but can fit more distributions (not only the ones belonging to the exponential family) and can model all the parameters of the distribution as functions of the explanatory variables (e.g. using linear, non-linear, smoothing, loess and random effects terms).

This implementation of gamlss() allows modelling of up to four parameters in a distribution family, which are conventionally called mu, sigma, nu and tau.

The function gamlssNews() shows what is new in the current implementation.

Usage

```
gamlss(formula = formula(data), sigma.formula = ~1,
        nu.formula = ~1, tau.formula = ~1, family = NO(),
        data = sys.parent(), weights = NULL,
        contrasts = NULL, method = RS(),  start.from = NULL,
        mu.start = NULL,  sigma.start = NULL,
        nu.start = NULL, tau.start = NULL,
        mu.fix = FALSE, sigma.fix = FALSE, nu.fix = FALSE,
        tau.fix = FALSE, control = gamlss.control(...),
        i.control = glim.control(...), ...)
is.gamlss(x)
gamlssNews()
```

Arguments

formula	a formula object, with the response on the left of an ~ operator, and the terms, separated by + operators, on the right. Nonparametric smoothing terms are indicated by pb() for penalised beta splines, cs for smoothing splines, lo for loess smooth terms and random or ra for random terms, e.g. y~cs(x,df=5)+x1+x2*x3. Additional smoothers can be added by creating the appropriate interface. Interactions with nonparametric smooth terms are not fully supported, but will not produce errors; they will simply produce the usual parametric interaction

sigma.formula	a formula object for fitting a model to the sigma parameter, as in the formula above, e.g. sigma.formula=~cs(x,df=5). It can be abbreviated to sigma.fo=~cs(x,df=5).
nu.formula	a formula object for fitting a model to the nu parameter, e.g. nu.fo=~x
tau.formula	a formula object for fitting a model to the tau parameter, e.g. tau.fo=~cs(x,df=2)
family	a gamlss.family object, which is used to define the distribution and the link functions of the various parameters. The distribution families supported by gamlss() can be found in gamlss.family. Functions such as BI() (binomial) produce a family object. Also can be given without the parentheses i.e. BI. Family functions can take arguments, as in BI(mu.link=probit)
data	a data frame containing the variables occurring in the formula. If this is missing, the variables should be on the search list. e.g. data=aids
weights	a vector of weights. Note that this is not the same as in the glm() or gam() function. Here weights can be used to weight out observations (like in subset) or for a weighted likelihood analysis where the contribution of the observations to the likelihood differs according to weights. The length of weights must be the same as the number of observations in the data. By default, the weight is set to one. To set weights to vector w use weights=w
contrasts	list of contrasts to be used for some or all of the factors appearing as variables in the model formula. The names of the list should be the names of the corresponding variables. The elements should either be contrast-type matrices (matrices with as many rows as levels of the factor and with columns linearly independent of each other and of a column of ones), or else they should be functions that compute such contrast matrices.
method	the current algorithms for GAMLSS are RS(), CG() and mixed(). i.e. method=RS() will use the Rigby and Stasinopoulos algorithm, method=CG() will use the Cole and Green algorithm and mixed(2,10) will use the RS algorithm twice before switching to the Cole and Green algorithm for up to 10 extra iterations
start.from	a fitted GAMLSS model which the fitted values will be used as staring values for the current model
mu.start	vector or scalar of initial values for the location parameter mu e.g. mu.start=4
sigma.start	vector or scalar of initial values for the scale parameter sigma e.g. sigma.start=1
nu.start	vector or scalar of initial values for the parameter nu e.g. nu.start=3
tau.start	vector or scalar of initial values for the location parameter tau e.g. tau.start=2
mu.fix	whether the mu parameter should be kept fixed in the fitting processes e.g. mu.fix=FALSE
sigma.fix	whether the sigma parameter should be kept fixed in the fitting processes e.g. sigma.fix=FALSE
nu.fix	whether the nu parameter should be kept fixed in the fitting processes e.g. nu.fix=FALSE
tau.fix	whether the tau parameter should be kept fixed in the fitting processes e.g. tau.fix=FALSE
control	this sets the control parameters of the outer iterations algorithm. The default setting is the gamlss.control function
i.control	this sets the control parameters of the inner iterations of the RS algorithm. The default setting is the glim.control function

...	for extra arguments
x	an object

Details

The Generalized Additive Model for Location, Scale and Shape is a general class of statistical models for a univariate response variable. The model assumes independent observations of the response variable y given the parameters, the explanatory variables and the values of the random effects. The distribution for the response variable in the GAMLSS can be selected from a very general family of distributions including highly skew and/or kurtotic continuous and discrete distributions, see gamlss.family. The systematic part of the model is expanded to allow modelling not only of the mean (or location) parameter, but also of the other parameters of the distribution of y, as linear parametric and/or additive nonparametric (smooth) functions of explanatory variables and/or random effects terms. Maximum (penalized) likelihood estimation is used to fit the (non)parametric models. A Newton-Raphson/Fisher scoring algorithm is used to maximize the (penalized) likelihood. The additive terms in the model are fitted using a backfitting algorithm.

`is.gamlss` is a short version is `is(object,"gamlss")`

Value

Returns a gamlss object with components

family	the distribution family of the gamlss object (see gamlss.family)
parameters	the name of the fitted parameters i.e. mu, sigma, nu, tau
call	the call of the gamlss function
y	the response variable
control	the gamlss fit control settings
weights	the vector of weights
G.deviance	the global deviance
N	the number of observations in the fit
rqres	a function to calculate the normalized (randomized) quantile residuals of the object
iter	the number of external iterations in the fitting process
type	the type of the distribution or the response variable (continuous or discrete)
method	which algorithm is used for the fit, RS(), CG() or mixed()
converged	whether the model fitting has have converged
residuals	the normalized (randomized) quantile residuals of the model
mu.fv	the fitted values of the mu model, also sigma.fv, nu.fv, tau.fv for the other parameters if present
mu.lp	the linear predictor of the mu model, also sigma.lp, nu.lp, tau.lp for the other parameters if present
mu.wv	the working variable of the mu model, also sigma.wv, nu.wv, tau.wv for the other parameters if present

mu.wt	the working weights of the mu model, also sigma.wt, nu.wt, tau.wt for the other parameters if present
mu.link	the link function for the mu model, also sigma.link, nu.link, tau.link for the other parameters if present
mu.terms	the terms for the mu model, also sigma.terms, nu.terms, tau.terms for the other parameters if present
mu.x	the design matrix for the mu, also sigma.x, nu.x, tau.x for the other parameters if present
mu.qr	the QR decomposition of the mu model, also sigma.qr, nu.qr, tau.qr for the other parameters if present
mu.coefficients	the linear coefficients of the mu model, also sigma.coefficients, nu.coefficients, tau.coefficients for the other parameters if present
mu.formula	the formula for the mu model, also sigma.formula, nu.formula, tau.formula for the other parameters if present
mu.df	the mu degrees of freedom also sigma.df, nu.df, tau.df for the other parameters if present
mu.nl.df	the non linear degrees of freedom, also sigma.nl.df, nu.nl.df, tau.nl.df for the other parameters if present
df.fit	the total degrees of freedom use by the model
df.residual	the residual degrees of freedom left after the model is fitted
aic	the Akaike information criterion
sbc	the Bayesian information criterion

Warning

Respect the parameter hierarchy when you are fitting a model. For example a good model for mu should be fitted before a model for sigma is fitted

Note

The following generic functions can be used with a GAMLSS object: print, summary, fitted, coef, residuals, update, plot, deviance, formula

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk> and Calliope Akantziliotou

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss.family, pdf.plot, find.hyper

Examples

```
data(abdom)
mod<-gamlss(y~pb(x),sigma.fo=~pb(x),family=BCT, data=abdom, method=mixed(1,20))
plot(mod)
rm(mod)
```

---

gamlss.control *Auxiliary for Controlling GAMLSS Fitting*

Description

Auxiliary function as user interface for gamlss fitting. Typically only used when calling gamlss function with the option control.

Usage

```
gamlss.control(c.crit = 0.001, n.cyc = 20, mu.step = 1, sigma.step = 1, nu.step = 1,
               tau.step = 1, gd.tol = 5, iter = 0, trace = TRUE, autostep = TRUE,
                save = TRUE, ...)
```

Arguments

c.crit	the convergence criterion for the algorithm
n.cyc	the number of cycles of the algorithm
mu.step	the step length for the parameter mu
sigma.step	the step length for the parameter sigma
nu.step	the step length for the parameter nu
tau.step	the step length for the parameter tau
gd.tol	global deviance tolerance level
iter	starting value for the number of iterations, typically set to 0 unless the function refit is used
trace	whether to print at each iteration (TRUE) or not (FALSE)
autostep	whether the steps should be halved automatically if the new global deviance is greater that the old one, the default is autostep=TRUE

save	save=TRUE, (the default), saves all the information on exit. save=FALSE saves only limited information as the global deviance and AIC. For example fitted values, design matrices and additive terms are not saved. The latest is useful when gamlss() is called several times within a procedure.
...	for extra arguments

Details

The step length for each of the parameters mu, sigma, nu or tau is very useful to aid convergence if the parameter has a fully parametric model. However using a step length is not theoretically justified if the model for the parameter includes one or more smoothing terms, (even thought it may give a very approximate result).

The c.crit can be increased to speed up the convergence especially for a large set of data which takes longer to fit. When 'trace' is TRUE, calls to the function cat produce the output for each outer iteration.

Value

A list with the arguments as components.

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss

Examples

```
data(aids)
h<-gamlss(y~poly(x,3)+qrt, family=PO, data=aids) #
con<-gamlss.control(mu.step=0.1)
h<-gamlss(y~poly(x,3)+qrt, family=PO, data=aids, control=con) #
rm(h,con)
```

gamlss.cs *Support for Function cs(), s() and vc()*

Description

This is support for the functions cs(), s() and vc(). It is not intended to be called directly by users. The function gamlss.cs is based on the R function smooth.spline

Usage

```
gamlss.cs(x, y = NULL, w = NULL, df = 5, spar = NULL, cv = FALSE,  nknots = NULL,
         all.knots = TRUE, df.offset = 0, penalty = 1,
         control.spar = list(low = -1.5, high = 2), xeval = NULL)
gamlss.scs(x, y = NULL, w = NULL, df = NULL, spar = NULL, cv = FALSE, all.knots = FALSE,
          nknots = NULL, keep.data = TRUE, df.offset = 0, penalty = 1.4,
          control.spar = list(low = -1.5, high = 2), xeval = NULL)
gamlss.vc(x, y = NULL, w = NULL, df = 5, spar = NULL, cv = FALSE, all.knots = TRUE,
          df.offset = 0, penalty = 1, control.spar = list(low = -1.5, high = 2),
          xeval = NULL)
```

Arguments

x	the design matrix
y	the response variable
w	prior weights
df	effective degrees of freedom
spar	spar the smoothing parameter
cv	options for the smooth.spline function not to use here
nknots	options for the smooth.spline function not to use here
all.knots	options for the smooth.spline function not to use here
df.offset	options for the smooth.spline function not to use here
keep.data	whether to keep data
penalty	options for the smooth.spline function not to use here
control.spar	control for spar. It can be changed through cs
xeval	used in prediction

Value

Returns a class "smooth.spline" object with

residuals	The residuals of the fit
fitted.values	The smoothing values
var	the variance for the fitted smoother

lambda	the final value for spar
nl.df	the smoothing degrees of freedom excluding the constant and linear terms, i.e. (df-2)
coefSmo	this is a list containing among others the knots and the coefficients
...	

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>

See Also

gamlss, cs

---

gamlss.fp *Support for Function fp()*

Description

Those are support for the functions fp() and pp. It is not intended to be called directly by users.

Usage

```
gamlss.fp(x, y, w, npoly = 2, xeval = NULL)
gamlss.pp(x, y, w)
```

Arguments

x	the x for function gamlss.fp is referred to the design matric of the specific parameter model (not to be used by the user)
y	the y for function gamlss.fp is referred to the working variable of the specific parameter model (not to be used by the user)
w	the w for function gamlss.fp is referred to the iterative weight variable of the specific parameter model (not to be used by the user)
npoly	a positive indicating how many fractional polynomials should be considered in the fit. Can take the values 1, 2 or 3 with 2 as default
xeval	used in prediction

Value

Returns a list with

`fitted.values`	fitted
`residuals`	residuals
`var`	
`nl.df`	the trace of the smoothing matrix
`lambda`	the value of the smoothing parameter
`coefSmo`	the coefficients from the smoothing fit
`varcoeff`	the variance of the coefficients

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, fp

`gamlss.lo` *Support for Function lo()*

Description

This is support for the `loess` function `lo()`. It is not intended to be called directly by users. The function `gamlss.lo` is based on the R function `loess`

Usage

```
gamlss.lo(x, y, w = NULL, span, df = NULL, degree = 1, ncols =
                FALSE, wspan = TRUE, parametric = FALSE, drop.square
                = FALSE, normalize = FALSE, family = "gaussian",
                method = "loess", control = loess.control(...), xeval
                = NULL, ...)
```

Arguments

x	the design matrix
y	the response variable
w	prior weights
span	the smoothing parameter
df	effective degrees of freedom
degree	the order of the polynomial
ncols	the number of columns of the x matrix
wspan	argument for the loess function not to use here
parametric	argument for the loess function not to use here
drop.square	argument for the loess function not to use here
normalize	argument for the loess function not to use here
family	argument for the loess function not to use here
method	argument for the loess function not to use here
control	argument for the loess function not to use here
xeval	used in prediction
...	further arguments passed to or from other methods.

Value

Returns an object with

fitted	the smooth values
residuals	the residuals
var	the variance of the smoother
nl.df	the non-linear degrees of freedom
coefSmo	with value NULL
lambda	the value of span

Author(s)

Mikis Stasinopoulos based on Brian Ripley loess function in R

See Also

gamlss, lo

gamlss.ps *Support for Functions ps(), pb(), cy() and ri()*

Description

Those functions are support for the functions `ps()` and `pb()`. The functions are not intended to be called directly by users.

Usage

```
gamlss.ps(x, y, w, xeval = NULL, ...)
gamlss.pb(x, y, w, xeval = NULL, ...)
gamlss.ridge(x, y, w, xeval = NULL, ...)
gamlss.ri(x, y, w, xeval = NULL, ...)
gamlss.cy(x, y, w, xeval = NULL, ...)
gamlss.pvc(x, y, w, xeval = NULL, ...)
```

Arguments

`x`	the `x` for function `gamlss.fp` is referred to the design matric of the specific parameter model (not to be used by the user)
`y`	the `y` for function `gamlss.fp` is referred to the working variable of the specific parameter model (not to be used by the user)
`w`	the `w` for function `gamlss.fp` is referred to the iterative weight variable of the specific parameter model (not to be used by the user)
`xeval`	used in prediction
`...`	further arguments passed to or from other methods.

Value

`comp1`	Description of 'comp1'
`comp2`	Description of 'comp2'
...	

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2003) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, ps, pb

`gamlss.ra` *Support for Function ra()*

Description

This is support for the random effect smoother function ra() . It is not intended to be called directly by users. The function `gamlss.ra` is similar to the GAMLSS function `gamlss.random`. Both functions can be used with the same effect.

Usage

```
gamlss.ra(x, y, w, df = NULL)
```

Arguments

`x`	the explanatory design matrix
`y`	the response variable
`w`	iterative weights
`df`	effective degrees of freedom

Value

Returns an list with

`fitted.values`	fitted values
`residuals`	residuals
`var`	variances of the fitted values
`nl.df`	the trace of the smoothing matrix
`lambda`	the value of the smoothing parameter
`coefSmo`	the coefficients from the smoothing fit
`varcoeff`	the variance of the coefficients

Note

This is an experimental function and should be used with care

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>

References

RRigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, ra, random

---

gamlss.random *Support for Function random()*

Description

This is support for the function random(). It is not intended to be called directly by users. The function gamlss.radom is similar to the GAMLSS function gamlss.ra.

Usage

```
gamlss.random(x, y, w)
```

Arguments

x	the explanatory design matrix
y	the response variable
w	iterative weights

Value

Returns a list with

y	the fitted values
residuals	the residuals
var	the variance of the fitted values
lambda	the final lambda, the smoothing parameter
coefSmo	with value NULL

Author(s)

Mikis Stasinopoulos, based on Trevor Hastie function `gam.random`

References

Chambers, J. M. and Hastie, T. J. (1991). *Statistical Models in S*, Chapman and Hall, London.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, random, ra

`gamlss.rc` *Support for Function rc()*

Description

This is support for the function rc(). It is not intended to be called directly by users.

Usage

```
gamlss.rc(x, y, w, df = NULL)
```

Arguments

`x`	the explanatory design matrix
`y`	the response variable
`w`	iterative weights
`df`	effective degrees of freedom

Value

Returns a list with

`fitted.values`	fitted
`residuals`	residuals
`var`	
`nl.df`	the trace of the smoothing matrix

lambda	the value of the smoothing parameter
coefSmo	the coefficients from the smoothing fit
varcoeff	the variance of the coefficients

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, rc

---

gamlss.scope *Generate a Scope Argument for Stepwise GAMLSS*

Description

Generates a scope argument for a stepwise GAMLSS.

Usage

```
gamlss.scope(frame, response = 1, smoother = "cs", arg = NULL, form = TRUE)
```

Arguments

frame	a data or model frame
response	which variable is the response; the default is the first
smoother	what smoother to use; default is cs
arg	any additional arguments required by the smoother
form	should a formula be returned (default), or else a character version of the formula

Details

Each formula describes an ordered regimen of terms, each of which is eligible on their own for inclusion in the gam model. One of the terms is selected from each formula by step.gam. If a 1 is selected, that term is omitted.

Value

a list of formulas is returned, one for each column in frame (excluding the response). For a numeric variable, say x1, the formula is

~ 1 + x1 + cs(x1)

If x1 is a factor, the last smooth term is omitted.

Author(s)

Mikis Stasinopoulos: a modified function from Statistical Models in S

References

Chambers, J. M. and Hastie, T. J. (1991). *Statistical Models in S*, Chapman and Hall, London.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

stepGAIC

Examples

```
data(usair)
gs1<-gamlss.scope(model.frame(y~x1+x2+x3+x4+x5+x6, data=usair))
gs2<-gamlss.scope(model.frame(usair))
gs1
gs2
gs3<-gamlss.scope(model.frame(usair), smooth="fp", arg="3")
gs3
```

---

gamlssML *Maximum Likelihood estimation of a simple GAMLSS model*

Description

This is a function for fitting a `gamlss.family` distribution to sigle data set using a non linear maximisation algorithm in R. This is relevant only when there are not explanatory variables.

Usage

```
gamlssML(y, family = NO, weights = NULL, mu.start = NULL,
       sigma.start = NULL, nu.start = NULL, tau.start = NULL,
       mu.fix = FALSE, sigma.fix = FALSE, nu.fix = FALSE,
       tau.fix = FALSE, data = NULL, ...)
```

Arguments

y	a vector of data requiring the fit of a gamlss.family distribution
family	gamlss.family object, which is used to define the distribution and the link functions of the various parameters. The distribution families supported by gamlssML() can be found in gamlss.family
weights	a vector of weights. Here weights can be used to weight out observations (like in subset) or for a weighted likelihood analysis where the contribution of the observations to the likelihood differs according to weights. The length of weights must be the same as the number of observations in the data. By default, the weight is set to one. To set weights to vector w use weights=w
mu.start	a scalar of initial values for the location parameter mu e.g. mu.start=4
sigma.start	a scalar of initial values for the scale parameter sigma e.g. sigma.start=1
nu.start	scalar of initial values for the parameter nu e.g. nu.start=3
tau.start	scalar of initial values for the parameter tau e.g. tau.start=3
mu.fix	whether the mu parameter should be kept fixed in the fitting processes e.g. mu.fix=FALSE
sigma.fix	whether the sigma parameter should be kept fixed in the fitting processes e.g. sigma.fix=FALSE
nu.fix	whether the nu parameter should be kept fixed in the fitting processes e.g. nu.fix=FALSE
tau.fix	whether the tau parameter should be kept fixed in the fitting processes e.g. tau.fix=FALSE
data	a data frame containing the variable y. If this is missing, the variable should be on the search list. e.g. data=aids
...	for extra arguments

Details

This function which fits a gamlss.family distribution to a single data set is using a non linear maximisation. in fact it uses the internal function MLE() which is a copy of the mle() function of package stat4. The function gamlssML() could be for large data faster than the equivalent gamlss() function which is designed for regression type of models.

Value

Returns a gamlssML object which behaves like a gamlss fitted objected

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>, Vlasis Voudouris <v.voiudouris@londonmet.ac.uk> and Majid Djennad <m.djennad.londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss.family, gamlss

Examples

```
#-------- negative binomial 1000 observations
y<- rNBI(1000)
 system.time(m1<-gamlss(y~1, family=NBI))
  system.time(m1a<-gamlss(y~1, family=NBI, trace=FALSE))
system.time(m11<-gamlssML(y, family=NBI))
AIC(m1,m1a,m11, k=0)
# neg. binomial   n=10000
 y<- rNBI(10000)
 system.time(m1<-gamlss(y~1, family=NBI))
  system.time(m1a<-gamlss(y~1, family=NBI, trace=FALSE))
system.time(m11<-gamlssML(y, family=NBI))
AIC(m1,m1a,m11, k=0)
```

---

glim.control *Auxiliary for Controlling the inner algorithm in a GAMLSS Fitting*

Description

Auxiliary function used for the inner iteration of gamlss algorithm. Typically only used when calling gamlss function through the option i.control.

Usage

```
glim.control(cc = 0.001, cyc = 50,  glm.trace = FALSE,
             bf.cyc = 30, bf.tol = 0.001, bf.trace = FALSE,
             ...)
```

Arguments

cc	the convergence criterion for the algorithm
cyc	the number of cycles of the algorithm
glm.trace	whether to print at each iteration (TRUE) or not (FALSE)
bf.cyc	the number of cycles of the backfitting algorithm
bf.tol	the convergence criterion (tolerance level) for the backfitting algorithm
bf.trace	whether to print at each iteration (TRUE) or not (FALSE, the default)
...	for extra arguments

Value

A list with the arguments as components

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape, (with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss

Examples

```
data(aids)
con<-glim.control(glm.trace=TRUE)
h<-gamlss(y~poly(x,3)+qrt, family=PO, data=aids, i.control=con) #
rm(h,con)
```

`histDist` *This function plots the histogram and a fitted (GAMLSS family) distribution to a variable*

Description

This function fits constants to the parameters of a GAMLSS family distribution and them plot the histogram and the fitted distribution.

Usage

```
histDist(y, family = NO, freq = NULL,
        density = FALSE, nbins = 10, xlim = NULL,
        ylim = NULL, main = NULL, xlab = NULL,
        ylab = NULL, data = NULL, ...)
```

Arguments

`y`	a vector for the response variable
`family`	a `gamlss.family` distribution
`freq`	the frequencies of the data in `y` if exist. `freq` is used as `weights` in the `gamlss` fit
`density`	default value is FALSE. Change to TRUE if you would like a non-parametric density plot together with the parametric fitted distribution plot (for continuous variable only)
`nbins`	The suggested number of bins (argument passed to `truehist()` of package MASS). Either a positive integer, or a character string naming a rule: "Scott" or "Freedman-Diaconis" or "FD". (Case is ignored.)
`xlim`	the minimum and the maximum x-axis value (if the default values are out of range)
`ylim`	the minimum and the maximum y-axis value (if the default values are out of range)
`main`	the main title for the plot
`xlab`	the label in the x-axis
`ylab`	the label in the y-axis
`data`	the data.frame
`...`	for extra arguments to be passed to the `gamlss` function

Details

This function first fits constants for each parameters of a GAMLSS distribution family using the `gamlss` function and them plots the fitted distribution together with the appropriate plot according to whether the `y` variable is of a continuous or discrete type. Histogram is plotted for continuous and barplot for discrete variables. The function `truehist` of Venables and Ripley's MASS package is used for the histogram plotting.

Value

returns a plot

Author(s)

Mikis Stasinopoulos

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, gamlss.family

Examples

```
data(abdom)
attach(abdom)
histDist(y,family="NO")
# use the ymax argument of the of the truehist()
histDist(y,family="NO",ymax=0.005)
# bad fit use PE
histDist(y,family="PE",ymax=0.005)
detach(abdom)
# discere data counts
#  Hand at al. p150  Leptinotarsa decemlineata
y <- c(0,1,2,3,4,6,7,8,10,11)
freq <- c(33,12,5,6,5,2,2,2,1,2)
histDist(y, "NBI", freq=freq)
# the same as
histDist(rep(y,freq), "NBI")
```

IC *Gives the GAIC for a GAMLSS Object*

Description

IC is a function to calculate the Generalized Akaike information criterion (GAIC) for a given penalty k for a fitted GAMLSS object. The function AIC.gamlss is the method associated with a GAMLSS object of the generic function AIC. The function GAIC is a synonymous of the function AIC.gamlss. The function extractAIC is a the method associated a GAMLSS object of the generic function extractAIC and it is mainly used in the stepAIC function.

Usage

```
IC(object, k = 2)
## S3 method for class 'gamlss'
AIC(object, ..., k = 2)
GAIC(object, ..., k = 2 )
## S3 method for class 'gamlss'
extractAIC(fit, scale, k = 2, ...)
```

Arguments

object	an gamlss fitted model
fit	an gamlss fitted model
...	allows several GAMLSS object to be compared using a GAIC
k	the penalty with default k=2.5
scale	this argument is not used in gamlss

Value

The function IC returns the GAIC for given penalty k of the GAMLSS object. The function AIC returns a matrix contains the df's and the GAIC's for given penalty k. The function GAIC returns identical results to AIC. The function extractAIC returns vector of length two with the degrees of freedom and the AIC criterion.

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss

Examples

```
data(abdom)
mod1<-gamlss(y~pb(x),sigma.fo=~pb(x),family=BCT, data=abdom)
IC(mod1)
mod2<-gamlss(y~pb(x),sigma.fo=~x,family=BCT, data=abdom)
AIC(mod1,mod2,k=3)
GAIC(mod1,mod2,k=3)
extractAIC(mod1,k=3)
rm(mod1,mod2)
```

---

lo *Specify a loess fit in a GAMLSS formula*

Description

Allows the user to specify a loess fit in a GAMLSS formula. This function is similar to the lo function in the gam implementation of S-plus

Usage

```
lo(..., span = 0.5, df = NULL, degree = 1)
```

Arguments

...	the unspecified ... can be a comma-separated list of numeric vectors, numeric matrix, or expressions that evaluate to either of these. If it is a list of vectors, they must all have the same length.
span	the number of observations in a neighborhood. This is the smoothing parameter for a loess fit.
df	the effective degrees of freedom can be specified instead of span, e.g. df=5
degree	the degree of local polynomial to be fit; can be 1 or 2.

Details

Note that lo itself does no smoothing; it simply sets things up for the function gamlss.lo() which is used by the backfitting function gamlss.add().

Value

a numeric matrix is returned. The simplest case is when there is a single argument to lo and degree=1; a one-column matrix is returned, consisting of a normalized version of the vector. If degree=2 in this case, a two-column matrix is returned, consisting of a 2d-degree orthogonal-polynomial basis. Similarly, if there are two arguments, or the single argument is a two-column matrix, either a two-column matrix is returned if degree=1, or a five-column matrix consisting of powers and products up to degree 2. Any dimensional argument is allowed, but typically one or two vectors are used in practice. The matrix is endowed with a number of attributes; the matrix itself is used in the construction of the model matrix, while the attributes are needed for the backfitting algorithms all.wam or lo.wam (weighted additive model). Local-linear curve or surface fits reproduce linear responses, while local-quadratic fits reproduce quadratic curves or surfaces. These parts of the loess fit are computed exactly together with the other parametric linear parts of the model.

Warning

For user wanted to compare the `gamlss()` results with the equivalent `gam()` results in S-plus: make sure that the convergence criteria epsilon and bf.epsilon in S-plus are decreased sufficiently to ensure proper convergence in S-plus

Note

Note that `lo` itself does no smoothing; it simply sets things up for `gamlss.lo()` to do the backfitting.

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>, based on the Trevor Hastie S-plus `lo` function

References

Chambers, J. M. and Hastie, T. J. (1991). *Statistical Models in S*, Chapman and Hall, London.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

cs, random,

Examples

```
data(aids)
attach(aids)
# fitting a loess curve with span=0.4 plus the a quarterly  effect
aids1<-gamlss(y~lo(x,span=0.4)+qrt,data=aids,family=PO) #
plot(x,y)
lines(x,fitted(aids1))
rm(aids1)
detach(aids)
```

---

lpred *Extract Linear Predictor Values and Standard Errors For A GAMLSS Model*

Description

`lpred` is the GAMLSS specific method which extracts the linear predictor and its (approximate) standard errors for a specified parameter from a GAMLSS objects. The `lpred` can be also used to extract the fitted values (with its approximate standard errors) or specific terms in the model (with its approximate standard errors) in the same way that the `predict.lm()` and `predict.glm()` functions can be used for `lm` or `glm` objects. The function `lp` extract only the linear predictor. If prediction is required for new data values then use the function `predict.gamlss()`.

Usage

```
lpred(obj, what = c("mu", "sigma", "nu", "tau"),
           type = c("link", "response", "terms"),
           terms = NULL, se.fit = FALSE, ...)
lp(obj, what = "mu", ...)
```

Arguments

`obj`	a GAMLSS fitted model
`what`	which distribution parameter is required, default `what="mu"`
`type`	`type="link"` (the default) gets the linear predictor for the specified distribution parameter. `type="response"` gets the fitted values for the parameter while `type="terms"` gets the fitted terms contribution
`terms`	if `type="terms"`, which terms to be selected (default is all terms)
`se.fit`	if TRUE the approximate standard errors of the appropriate type are extracted
`...`	for extra arguments

Value

If `se.fit=FALSE` a vector (or a matrix) of the appropriate `type` is extracted from the GAMLSS object for the given parameter in `what`. If `se.fit=TRUE` a list containing the appropriate `type`, `fit`, and its (approximate) standard errors, `se.fit`.

Author(s)

Mikis Stasinopoulos

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

predict.gamlss

Examples

```
data(aids)
mod<-gamlss(y~poly(x,3)+qrt, family=PO, data=aids) #
mod.t <- lpred(mod, type = "terms", terms= "qrt")
mod.t
mod.lp <- lp(mod)
mod.lp
rm(mod, mod.t,mod.lp)
```

LR.test *Likelihood Ratio test for nested GAMLSS models*

Description

The function performs a likelihood ration test for two nested fitted model.

Usage

```
LR.test(null, alternative, print = TRUE)
```

Arguments

null	The null hypothesis (simpler) fitted model
alternative	The alternative hypothesis (more complex) fitted model
print	whether to print or save the result

Details

Warning: no checking whether the models are nested is performed.

Value

If `print=FALSE` a list with `chi`, `df` and `p.val` is produced.

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, dropterm

Examples

```
data(usair)
m0<-gamlss(y~x1+x2, data=usair)
m1<-gamlss(y~x1+x2+x3+x4, data=usair)
LR.test(m0,m1)
```

`model.frame.gamlss` *Extract a model.frame, a model matrix or terms from a GAMLSS object for a given distributional parameter*

Description

`model.frame.gamlss`, `model.matrix.gamlss` and `terms.gamlss` are the gamlss versions of the generic functions `model.frame`, `model.matrix` and `terms` respectively.

Usage

```
## S3 method for class 'gamlss'
model.frame(formula, what = c("mu", "sigma", "nu", "tau"), ...)
## S3 method for class 'gamlss'
terms(x, what = c("mu", "sigma", "nu", "tau"), ...)
## S3 method for class 'gamlss'
model.matrix(object, what = c("mu", "sigma", "nu", "tau"), ...)
```

Arguments

formula	a gamlss object
x	a gamlss object
object	a gamlss object
what	for which parameter to extract the model.frame, terms or model.frame
...	for extra arguments

Value

a model.frame, a model.matrix or terms

Author(s)

Mikis Stasinopoulos

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss

Examples

```
data(aids)
mod<-gamlss(y~poly(x,3)+qrt, family=PO, data=aids) #
model.frame(mod)
model.matrix(mod)
terms(mod, "mu")
rm(mod)
```

`par.plot` *A function to plot parallel plot for repeated measurement data*

Description

This function can be used to plot parallel plots for each individual in a repeated measurement study. It is based on the `coplot()` function of `R`.

Usage

```
par.plot(formula = NULL, data = NULL, subjects = NULL,
          color = TRUE, show.given = TRUE, ...)
```

Arguments

`formula`	a formula describing the form of conditioning plot. A formula of the form `y ~ x | a` indicates that plots of `y` versus `x` should be produced conditional on the variable `a`. A formula of the form `y ~ x| a * b` indicates that plots of `y` versus `x` should be produced conditional on the two variables `a` and `b`.
`data`	a data frame containing values for any variables in the formula. By default the environment where `par.plot` was called from is used.
`subjects`	a factor which distinguish between the individual participants
`color`	whether the parallel plot are shown in colour, `color=TRUE` (the default) or not `color=FALSE`
`show.given`	logical (possibly of length 2 for 2 conditioning variables): should conditioning plots be shown for the corresponding conditioning variables (default 'TRUE')
`...`	for extra arguments

Value

It returns a plot.

Note

Note that similar plot can be fount in the library `nlme` by Pinheiro and Bates

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss

Examples

```
library(nlme)
data(Orthodont)
par.plot(distance~age,data=Orthodont,sub=Subject)
par.plot(distance~age|Sex,data=Orthodont,sub=Subject)
par.plot(distance~age|Subject,data=Orthodont,sub=Subject,show.given=FALSE)
```

---

pdf.plot *Plots Probability Distribution Functions for GAMLSS Family*

Description

A function to plot probability distribution functions (pdf) belonging to the gamlss family of distributions. This function allows either plotting of the fitted distributions for up to eight observations or plotting specified distributions belonging in the gamlss family

Usage

```
pdf.plot(obj = NULL, obs = c(1), family = NO(), mu = NULL,
         sigma = NULL, nu = NULL, tau = NULL, min = NULL,
         max = NULL, step = NULL, allinone = FALSE,
         no.title = FALSE, ...)
```

Arguments

obj	An gamlss object e.g. obj=model1 where model1 is a fitted gamlss object
obs	A number or vector of up to length eight indicating the case numbers of the observations for which fitted distributions are to be displayed, e.g. obs=c(23,58) will display the fitted distribution for the 23th and 58th observations

family	This must be a gamlss family i.e. family=NO
mu	The value(s) of the location parameter mu for which the distribution has to be evaluated e.g mu=c(3,7)
sigma	The value(s) the scale parameter sigma for which the distribution has to be evaluated e.g sigma=c(3,7)
nu	The value(s) the parameter nu for which the distribution has to be evaluated e.g. nu=3
tau	The value(s) the parameter tau for which the distribution has be evaluated e.g. tau=5
min	Minimum value of the random variable y e.g. min=0
max	Maximum value of y e.g. max=10
step	Steps for the evaluation of y e.g. step=0.5
allinone	This will go
no.title	Whether you need title in the plot, default is no.title=FALSE
...	for extra arguments

Details

This function can be used to plot distributions of the GAMLSS family. If the first argument obj is specified and it is a GAMLSS fitted object, then the fitted distribution of this model at specified observation values (given by the second argument obs) is plotted for a specified y-variable range (arguments min, max, and step).

If the first argument is not given then the family argument has to be specified and the pdf is plotted at specified values of the parameters mu, sigma, nu, tau. Again the range of the y-variable has to be given.

Value

plot(s) of the required pdf(s) are returned

Warning

The range of some distributions depends on the fitted parameters

Note

The range of the y values given by min, max and step are very important in the plot

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk> and Calliope Akantziliotou

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss

Examples

```
pdf.plot(family=BCT, min=1, max=20, step=.05, mu=10, sigma=0.15, nu=-1, tau=c(4,10,20,40) )
# now using an gamlss object
# library(gamlss)
#data(abdom)
#h<-gamlss(y~pb(x), sigma.formula=~pb(x), family=BCT, data=abdom) # fits
#pdf.plot(obj=h , obs=c(23,67), min=50, max=150, step=.5)
```

---

plot.gamlss *Plot Residual Diagnostics for an GAMLSS Object*

Description

This function provides four plots for checking the normalized (randomized for a discrete response distribution) quantile residuals of a fitted GAMLSS object, referred to as residuals below : a plot of residuals against fitted values, a plot of the residuals against an index or a specific explanatory variable, a density plot of the residuals and a normal Q-Q plot of the residuals. If argument `ts=TRUE` then the first two plots are replaced by the autocorrelation function (ACF) and partial autocorrelation function (PACF) of the residuals

Usage

```
## S3 method for class 'gamlss'
plot(x, xvar = NULL, parameters = NULL, ts = FALSE,
           summaries = TRUE, ...)
```

Arguments

x	a GAMLSS fitted object
xvar	an explanatory variable to plot the residuals against
parameters	plotting parameters can be specified here

`ts`	set this to TRUE if ACF and PACF plots of the residuals are required
`summaries`	set this to FALSE if no summary statistics of the residuals are required
`...`	further arguments passed to or from other methods.

Details

This function provides four plots for checking the normalized (randomized) quantile residuals (called `residuals`) of a fitted GAMLSS object. Randomization is only performed for discrete response variables. The four plots are

- residuals against the fitted values (or ACF of the residuals if `ts=TRUE`)
- residuals against an index or specified x-variable (or PACF of the residuals if `ts=TRUE`)
- kernel density estimate of the residuals
- QQ-normal plot of the residuals

For time series response variables option `ts=TRUE` can be used to plot the ACF and PACF functions of the residuals.

Value

Returns four plots related to the residuals of the fitted GAMLSS model and prints summary statistics for the residuals if the `summary=T`

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk> and Kalliope Akantziliotou

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss

Examples

```
data(aids)
a<-gamlss(y~pb(x)+qrt,family=PO,data=aids)
plot(a)
rm(a)
```

---

polyS *Auxiliary support for the GAMLSS*

Description

These two functions are similar to the poly and polym in R. Are needed for the gamlss.lo function of GAMLSS and should not be used on their own.

Usage

```
polyS(x, ...)
poly.matrix(m, degree = 1)
```

Arguments

x	a variable
m	a variable
degree	the degree of the polynomial
...	for extra arguments

Value

Returns a matrix of orthogonal polynomials

Warning

Not be use by the user

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, gamlss.lo

`predict.gamlss` *Extract Predictor Values and Standard Errors For New Data In a GAMLSS Model*

Description

`predict.gamlss` is the GAMLSS specific method which produce predictors for a new data set for a specified parameter from a GAMLSS objects. The `predict.gamlss` can be used to extract the linear predictors, fitted values and specific terms in the model at new data values in the same way that the `predict.lm()` and `predict.glm()` functions can be used for `lm` or `glm` objects. Note that linear predictors, fitted values and specific terms in the model at the current data values can also be extracted using the function `lpred()` (which is called from predict if new data is NULL).

Usage

```
## S3 method for class 'gamlss'
predict(object, what = c("mu", "sigma", "nu", "tau"),
               newdata = NULL, type = c("link", "response", "terms"),
               terms = NULL, se.fit = FALSE, data = NULL, ...)
predictAll(object, newdata = NULL, type = c("response", "link", "terms"),
               terms = NULL, se.fit = FALSE, ...)
```

Arguments

`object`	a GAMLSS fitted model
`what`	which distribution parameter is required, default `what="mu"`
`newdata`	a data frame containing new values for the explanatory variables used in the model
`type`	the default, gets the linear predictor for the specified distribution parameter. `type="response"` gets the fitted values for the parameter while `type="terms"` gets the fitted terms contribution

terms	if type="terms", which terms to be selected (default is all terms)
se.fit	if TRUE the approximate standard errors of the appropriate type are extracted if exist
data	the data frame used in the original fit if is not defined in the call
...	for extra arguments

Details

The predict function assumes that the object given in newdata is a data frame containing the right x-variables used in the model. This could possible cause problems if transformed variables are used in the fitting of the original model. For example, let us assume that a transformation of age is needed in the model i.e. nage<-age^.5. This could be fitted as mod<-gamlss(y~cs(age^.5),data=mydata) or as nage<-age^.5; mod<-gamlss(y~cs(nage), data=mydata). The later could more efficient if the data are in thousands rather in hundreds. In the first case, the code predict(mod,newdata=data.frame(age=c(34,56) would produce the right results. In the second case a new data frame has to be created containing the old data plus any new transform data. This data frame has to be declared in the data option. The option newdata should contain a data.frame with the new names and the transformed values in which prediction is required, (see the last example).

Value

A vector or a matrix depending on the options.

Note

This function is under development

Author(s)

Mikis Stasinopoulos

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

lp, lpred

Examples

```
data(aids)
a<-gamlss(y~poly(x,3)+qrt, family=PO, data=aids) #
newaids<-data.frame(x=c(45,46,47), qrt=c(2,3,4))
ap <- predict(a, newdata=newaids, type = "response")
ap
# now getting all the parameters
predictAll(a, newdata=newaids)
rm(a, ap)
data(abdom)
# transform x
aa<-gamlss(y~cs(x^.5),data=abdom)
# predict at old values
predict(aa)[610]
# predict at new values
predict(aa,newdata=data.frame(x=42.43))
# now transform x first
nx<-abdom$x^.5
aaa<-gamlss(y~cs(nx),data=abdom)
# create a new data frame
newd<-data.frame( abdom, nx=abdom$x^0.5)
# predict at old values
predict(aaa)[610]
# predict at new values
predict(aaa,newdata=data.frame(nx=42.43^.5), data=newd)
```

---

print.gamlss *Prints a GAMLSS fitted model*

Description

`print.gamlss` is the GAMLSS specific method for the generic function `print` which prints objects returned by modelling functions.

Usage

```
## S3 method for class 'gamlss'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`x`	a GAMLSS fitted model
`digits`	the number of significant digits to use when printing
`...`	for extra arguments

Value

Prints a gamlss object

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk> and Calliope Akantziliotou

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files,(see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, deviance.gamlss, fitted.gamlss

Examples

```
data(aids)
h<-gamlss(y~poly(x,3)+qrt, family=PO, data=aids)
print(h) # or just h
rm(h)
```

---

prof.dev — *Plotting the Profile Deviance for one of the Parameters in a GAMLSS model*

Description

This functions plots the profile deviance of one of the (four) parameters in a GAMLSS model. It can be used if one of the parameters `mu`, `sigma`, `nu` or `tau` is a constant (not a function of explanatory variables) to obtain a profile confidence intervals.

Usage

```
prof.dev(object, which = NULL, min = NULL, max = NULL, step = NULL,
          startlastfit = TRUE, type = "o", plot = TRUE, perc = 95,
          ...)
```

Arguments

object	A fitted GAMLSS model
which	which parameter to get the profile deviance e.g. which="tau"
min	the minimum value for the parameter e.g. min=1
max	the maximum value for the parameter e.g. max=20
step	how often to evaluate the global deviance (defines the step length of the grid for the parameter) e.g. step=1
startlastfit	whether to start fitting from the last fit or not, default value is startlastfit=TRUE
type	what type of plot required. This is the same as in type for plot, default value is type="o", that is, both line and points
plot	whether to plot, plot=TRUE or save the results, plot=FALSE
perc	what % confidence interval is required
...	for extra arguments

Details

This function can be use to provide likelihood based confidence intervals for a parameter for which a constant model (i.e. no explanatory model) is fitted and consequently for checking the adequacy of a particular values of the parameter. This can be used to check the adequacy of one distribution (e.g. Box-Cox Cole and Green) nested within another (e.g. Box-Cox power exponential). For example one can test whether a Box-Cox Cole and Green (Box-Cox-normal) distribution or a Box-Cox power exponential is appropriate by plotting the profile of the parameter tau. A profile deviance showing support for tau=2 indicates adequacy of the Box-Cox Cole and Green (i.e. Box-Cox normal) distribution.

Value

A plot of profile global deviance

Warning

A dense grid (i.e. small step) evaluation of the global deviance can take a long time, so start with a sparse grid (i.e. large step) and decrease gradually the step length for more accuracy.

Author(s)

Calliope Akantziliotou, Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk> and Bob Rigby <r.rigby@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, prof.term

Examples

```
data(abdom)
h<-gamlss(y~pb(x), sigma.formula=~pb(x), family=BCT, data=abdom)
prof.dev(h,"nu",min=-2.000,max=2,step=0.25,type="l")
rm(h)
```

---

prof.term — *Plotting the Profile: deviance or information criterion for one of the terms (or hyper-parameters) in a GAMLSS model*

Description

This functions plots the profile deviance for a chosen parameter included in the linear predictor of any of the mu, sigma, nu or tau models so profile confidence intervals can be obtained. In can also be used to plot the profile of a specified information criterion for any hyperparameter.

Usage

```
prof.term(model = NULL, criterion = "GD", penalty = 2.5, other = NULL,
          min = NULL, max = NULL, step = NULL, type = "o", xlabel = NULL,
          plot = TRUE, term = TRUE, perc = 95, ... )
```

Arguments

model	this is a GAMLSS model, e.g. model=gamlss(y~cs(x,df=this),sigma.fo=~cs(x,df=3),data=abdom), where this indicates the (hyper)parameter to be profiled
criterion	whether global deviance ("GD") or information criterion ("IC") is profiled. The default is criterion="GD"
penalty	The penalty value if information criterion is used in criterion, default penalty=2.5
other	this can be used to evaluate an expression before the actual fitting of the model
min	the minimum value for the parameter e.g. min=1
max	the maximum value for the parameter e.g. max=20
step	how often to evaluate the global deviance (defines the step length of the grid for the parameter) e.g. step=1
type	what type of plot required. This is the same as in type for plot, default value is type="o", that is, both line and points

xlabel	if a label for the axis is required
plot	whether to plot, plot=TRUE or save the results, plot=FALSE
term	this has the value TRUE and it should be changed to FALSE if a profile global deviance is required for a hyperparameter so the IC are suppressed
perc	what % confidence interval is required
...	for extra arguments

Details

This function can be use to provide likelihood based confidence intervals for a parameter involved in terms in the linear predictor(s). These confidence intervals are more accurate than the ones obtained from the parameters' standard errors. The function can also be used to plot a profile information criterion (with a given penalty) against a hyperparameter. This can be used to check the uniqueness in hyperparameter determination using for example find.df.

Value

Return a profile plot (if the argument plot=TRUE) or the values of the parameters and the IC or GD values otherwise

Warning

A dense grid (i.e. small step) evaluation of the global deviance can take a long time, so start with a sparse grid (i.e. large step) and decrease gradually the step length for more accuracy.

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk> and Bob Rigby <r.rigby@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, prof.dev

Examples

```
data(aids)
gamlss(y~x+qrt,family=NBI,data=aids)
mod<-quote(gamlss(y ~ offset(this * x) + qrt, data = aids, family = NBI))
prof.term(mod, min=0.06, max=0.11, step=0.001)
mod1<-quote(gamlss(y ~ cs(x,df=this) + qrt, data = aids, family = NBI))
prof.term(mod1, min=1, max=15, step=1, criterion="IC")
mod2 <- quote(gamlss(y ~ x+I((x>this)*(x-this))+qrt,family=NBI,data=aids))
prof.term(mod2, min=1, max=45, step=1, criterion="GD")
rm(mod,mod1,mod2)
```

---

ps *Specify a Penalised B-Spline Fit in a GAMLSS Formula*

Description

There are several function operating using penalised B-splines: `pb()`, `cy()`, `pvc()` and `ps()`. The functions take a vector and return it with several attributes. The vector is used in the construction of the design matrix X used in the fitting. The functions do not do the smoothing, but assign the attributes to the vector to aid gamlss in the smoothing. The functions doing the smoothing are `gamlss.ps()` `gamlss.pb()`, `gamlss.cy()` and `gamlss.pvc()` which are used in the backfitting function `additive.fit`.

The function `pb()` is more efficient and faster than the original penalized smoothing function `ps()`. `pb()` allows the estimation of the smoothing parameters using different local (performance iterations) methods. The method are "ML", "ML-1", "EM", "GAIC" and "GCV". The function `cy()` fits a cycle penalised beta regression spline such as the last fitted value of the smoother is equal to the first fitted value. The function `pvc()` fits varying coefficient models see Hastie and Tibshirani(1993) and it is more general and flexible than the old `vc()` function which is based on cubic splines.

Usage

```
pb(x, df = NULL, lambda = NULL, control = pb.control(...), ...)
pb.control(inter = 20, degree = 3, order = 2, start = 10, quantiles = FALSE,
               method = c("ML", "GAIC", "GCV", "EM", "ML-1"), k = 2, ...)
cy(x, df = NULL, lambda = NULL, control = cy.control(...), ...)
cy.control(inter = 20, degree = 3, order = 2, start = 10,
          method = c("ML", "GAIC", "GCV", "EM", "ML-1"), k = 2, ts=FALSE, ...)
pvc(x, df = NULL, lambda = NULL, by = NULL, control = pvc.control(...), ...)
pvc.control(inter = 20, degree = 3, order = 2, start = 10, quantiles = FALSE,
             method = c("ML", "GAIC", "GCV", "EM", "ML-1"), k = 2, ...)
ps(x, df = 3, lambda = NULL, ps.intervals = 20, degree = 3, order = 3)
```

Arguments

`x`	the univariate predictor
`df`	the desired equivalent number of degrees of freedom (trace of the smoother matrix minus two for the constant and linear fit)

lambda	the smoothing parameter
control	setting the control parameters
by	a factor, for fitting different smoothing curves to each level of the factor or a continuous explanatory variable in which case the coefficients of the by variable change smoothly according to x i.e. beta(x)*z where z is the by variable.
...	for extra arguments
inter	the no of break points (knots) in the x-axis
degree	the degree of the piecewise polynomial
order	the required difference in the vector of coefficients
start	the lambda starting value if the local methods are used, see below
quantiles	if TRUE the quantile values of x are use to determine the knots
ts	if TRUE assumes that it is a seasonal factor
method	The method used in the (local) performance iterations. Available methods are "ML", "ML-1", "EM", "GAIC" and "GCV"
k	the penalty used in "GAIC" and "GCV"
ps.intervals	the no of break points in the x-axis

Details

The ps() function is based on Brian Marx function which can be found in http://www.stat.lsu.edu/faculty/marx/. The pb(), cy() and pvc() functions are based on Paul Eilers's original R functions. Note that ps() and pb() functions behave differently at their default values if df and lambda are not specified. ps(x) by default uses 3 extra degrees of freedom for smoothing x. pb(x) by default estimates lambda (and therefore the degrees of freedom) automatically using a "local" method. The local (or performance iterations) methods available are: (i) local Maximum Likelihood, "ML", (ii) local Generalized Akaike information criterion, "GAIC", (iii) local Generalized Cross validation "GCV" (iv) local EM-algorithm, "EM" (which is very slow) and (v) a modified version of the ML, "ML-1" which produce identical results with "EM" but faster.

Note that the local (or performance iterations) methods can occasionally make the convergence of gamlss less stable compared to models where the degrees of freedom are fixed.

Value

the vector x is returned, endowed with a number of attributes. The vector itself is used in the construction of the model matrix, while the attributes are needed for the backfitting algorithms additive.fit().

Warning

There are occations where the automatic local methods do not work. One accation which came to our attention is when the range of the response variable values is very large. Scalling the response variable will solve the problem.

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk > and Paul Eilers

References

http://www.stat.lsu.edu/faculty/marx/

Eilers, P. H. C. and Marx, B. D. (1996). Flexible smoothing with B-splines and penalties (with comments and rejoinder). *Statist. Sci*, **11**, 89-121.

Hastie, T. J. and Tibshirani, R. J. (1993), Varying coefficient models (with discussion),J. R. Statist. Soc. B., **55**, 757-796.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, gamlss.ps, cs

Examples

```
#==============================
# pb() and ps() functions
data(aids)
# fitting a smoothing cubic spline with 7 degrees of freedom
# plus the a quarterly  effect
aids1<-gamlss(y~ps(x,df=7)+qrt,data=aids,family=PO) # fix df's
aids2<-gamlss(y~pb(x,df=7)+qrt,data=aids,family=PO) # fix df's
aids3<-gamlss(y~pb(x)+qrt,data=aids,family=PO) # estimate lambda
with(aids, plot(x,y))
with(aids, lines(x,fitted(aids1),col="red"))
with(aids, lines(x,fitted(aids2),col="green"))
with(aids, lines(x,fitted(aids1),col="yellow"))
rm(aids1, aids2, aids3)
#==============================
# cy()
# simulate data
set.seed(555)
x = seq(0, 1, length = 100)
y = sign(cos(1 * x * 2 * pi + pi / 4)) + rnorm(length(x)) * 0.2
plot(y~x)
m1<-gamlss(y~cy(x))
lines(fitted(m1)~x)
rm(y,x,m1)
#==============================
# the pvc() function
# function to generate data
genData <- function(n=200)
 {
f1 <- function(x)-60+15*x-0.10*x^2
f2 <- function(x)-120+10*x+0.08*x^2
set.seed(1441)
x1 <- runif(n/2, min=0, max=55)
```

```
x2 <- runif(n/2, min=0, max=55)
y1 <- f1(x1)+rNO(n=n/2,mu=0,sigma=20)
y2 <- f2(x2)+rNO(n=n/2,mu=0,sigma=30)
 y <- c(y1,y2)
 x <- c(x1,x2)
 f <- gl(2,n/2)
da<-data.frame(y,x,f)
da
}
da<-genData(500)
plot(y~x, data=da, pch=21,bg=c("gray","yellow3")[unclass(f)])
# fitting models
# smoothing x
m1 <- gamlss(y~pb(x), data=da)
# parallel smoothing lines
m2 <- gamlss(y~pb(x)+f, data=da)
# linear interaction
m3 <- gamlss(y~pb(x)+f*x, data=da)
# varying coefficient model
m4 <- gamlss(y~pvc(x, by=f), data=da)
GAIC(m1,m2,m3,m4)
# plotting the fit
lines(fitted(m4)[da$f==1][order(da$x[da$f==1])]~da$x[da$f==1][order(da$x[da$f==1])], col="blue", lwd=2)
lines(fitted(m4)[da$f==2][order(da$x[da$f==2])]~da$x[da$f==2][order(da$x[da$f==2])], col="red", lwd=2)
rm(da,m1,m2,m3,m4)
#==================================
# the rent data
# first with a factor
data(rent)
plot(R~Fl, data=rent, pch=21,bg=c("gray","blue")[unclass(rent$B)])
r1 <- gamlss(R~pb(Fl), data=rent)
# identical to model
r11 <- gamlss(R~pvc(Fl), data=rent)
# now with the factor
r2 <- gamlss(R~pvc(Fl, by=B), data=rent)
lines(fitted(r2)[rent$B==1][order(rent$Fl[rent$B==1])]~rent$Fl[rent$B==1][order(rent$Fl[rent$B==1])], col="blue
lines(fitted(r2)[rent$B==0][order(rent$Fl[rent$B==0])]~rent$Fl[rent$B==0][order(rent$Fl[rent$B==0])], col="red"
# probably not very sensible model
rm(r1,r11,r2)
#-----------
# now with a continuous variable
# additive model
 h1 <-gamlss(R~pb(Fl)+pb(A), data=rent)
# varying-coefficient model
 h2 <-gamlss(R~pb(Fl)+pb(A)+pvc(A,by=Fl), data=rent)
AIC(h1,h2)
rm(h1,h2)
#==================================
```

`Q.stats` *A function to calculate the Q-statistics*

Description

This function calculates and prints the Q-statistics which are useful to test normality of the residuals within a range of an independent variable, for example age in centile estimation, see Royston and Wright (2000).

Usage

```
Q.stats(obj, xvar = NULL, xcut.points = NULL, n.inter = 10, zvals = TRUE,
        save = TRUE, ...)
```

Arguments

`obj`	a GAMLSS object or any other residual vector
`xvar`	a unique explanatory variable
`xcut.points`	the x-axis cut off points e.g. `c(20,30)`. If `xcut.points=NULL` then the `n.inter` argument is activated
`n.inter`	if `xcut.points=NULL` this argument gives the number of intervals in which the x-variable will be split, with default 4
`zvals`	if `TRUE` the output matix contains the individual z's rather that Q statistics
`save`	whether to save the Q-statistics or not with default equal to `TRUE`. In this case the functions produce a matrix giving individual Q (or z) statistics and the final aggregate Q's
`...`	for extra arguments

Value

A matrix containing the individual Q's and the aggregate Q-statistics

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>, with contributions from Elaine Borghie

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Royston P. and Wright E. M. (2000) Goodness of fit statistics for the age-specific reference intervals. *Statistics in Medicine*, 19, pp 2943-2962.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, centiles.split, wp

Examples

```
data(abdom)
h<-gamlss(y~pb(x), sigma.formula=~pb(x), family=BCT, data=abdom)
Q.stats(h,xvar=abdom$x,n.inter=8)
Q.stats(h,xvar=abdom$x,n.inter=8,zvals=FALSE)
rm(h)
```

---

ra *Specify Simple Random Effect In A GAMLSS Formula*

Description

This is an experimental smoother for use with factors in gamlss(). It allows the fitted values for a factor predictor to be shrunk towards the overall mean, where the amount of shrinking depends either on lambda, or on the equivalent degrees of freedom (df).

This function is slightly more general, but considerably slower than the random function .

Usage

```
ra(xfactor, xvector = NULL, df = NULL, lambda = NULL, order = 0,
   estimate = FALSE, expl = NULL, data1 = NULL)
```

Arguments

xfactor	a factor defining the subjects grouping in a one factor random effect model term e.g. xfactor=Subjects
xvector	a variable if interaction with the xfactor is required xvector (experimental)
df	required equivalent degrees of freedom e.g. df=10
lambda	the smoothing parameter which is the reciprocal (i.e. inverse) of the variance of the random effect
order	the order of the difference in the matrix D, order=1 is for simple random effects, order=2 is for random walk order 1 and order=3 is for random walk order 2
estimate	whether to estimate the lambda parameter within the backfitting iterations (very unreliable). Set by default to estimate=FALSE. [The lambda parameter can be more accurately estimated by selecting the corresponding smoothing degrees of freedom using find.hyper]
expl	this allows an explanatory variable at the subject level to be fitted e.g. expl=~x1+x2
data1	the data frame for the subject level variables data1

Value

xfactor is returned with class "smooth", with an attribute named "call" which is to be evaluated in the backfitting `additive.fit()` called by `gamlss()`

Warning

This is experimental and likely to change soon

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

random, gamlss

Examples

```
data(aids)
attach(aids)
# fitting a loess curve with span=0.4 plus the a quarterly  effect
aids1<-gamlss(y~lo(x,span=0.4)+qrt,data=aids,family=PO) #
# now we string the quarterly  effect using random
aids2<-gamlss(y~lo(x,span=0.4)+ra(qrt,df=2),data=aids,family=PO) #
plot(x,y)
lines(x,fitted(aids1),col="red")
lines(x,fitted(aids2),col="purple")
rm(aids1,aids2)
detach(aids)
```

`random` *Specify a simple random effect in a GAMLSS Formula*

Description

Includes random effect terms in an GAMLSS model. The function is based on the original `random()` function of Trevor Hastie in the package `gam`. This version of the function has been modified to allow a "local" maximum likelihood estimation of the smoothing parameter `lambda`. This method is equivalent to the PQL method of Breslow and Clayton (1993) applied at the local iterations of the algorithm. In fact for a GLM model and a simple random effect it is equivalent to `glmmPQL()` function in the package `MASS` see Venables and Ripley (2002). Venables and Ripley (2002) claimed that this iterative method was first introduced by Schall (1991). Note that in order for the "local" maximum likelhood estimation procedure to operate both argument `df` and `lambda` has to be `NULL`.

Usage

```
random(x, df = NULL, lambda = NULL, start=10)
```

Arguments

`x`	a factor
`df`	the target degrees of freedom
`lambda`	the smoothing parameter lambda which can be viewed as a shrinkage parameter.
`start`	starting value for lambda if local Maximul likelihood is used.

Details

This is a smoother for use with factors in gamlss(). It allows the fitted values for a factor predictor to be shrunk towards the overall mean, where the amount of shrinking depends either on lambda, or on the equivalent degrees of freedom. Similar in spirit to smoothing splines, this fitting method can be justified on Bayesian grounds or by a random effects model.

Note that the behavier of the function is different from the original Hastie function. Here the function behaves as follows: i) if both `df` and `lambda` are `NULL` then the PQL method is used ii) if `lambda` is not `NULL`, `lambda` is used for fitting iii) if `lambda` is `NULL` and `df` is not `NULL` then `df` is used for fitting.

Since factors are coded by model.matrix() into a set of contrasts, care has been taken to add an appropriate "contrast" attribute to the output of random(). This zero contrast results in a column of zeros in the model matrix, which is aliased with any column and is hence ignored

Value

x is returned with class "smooth", with an attribute named "call" which is to be evaluated in the backfitting `additive.fit()` called by `gamlss()`

Author(s)

Trevor Hastie (amended by Mikis Stasinopoulos)

References

Breslow, N. E. and Clayton, D. G. (1993) Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association* **88**, 9???25.

Chambers, J. M. and Hastie, T. J. (1991). *Statistical Models in S*, Chapman and Hall, London.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Schall, R. (1991) Estimation in generalized linear models with random effects. *Biometrika* **78**, 719???727.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S*. Fourth edition. Springer.

See Also

gamlss, gamlss.random

Examples

```
data(hodges)
plot(prind~state, data=hodges)
m1<- gamlss(prind~random(state), sigma.fo=~random(state), nu.fo=~random(state), tau.fo=~random(state), family=BC
edfAll(m1)
# radnom effect for nu is not needed
m2<- gamlss(prind~random(state), sigma.fo=~random(state), nu.fo=~random(state),  family=BCT, data=hodges, start.f
edfAll(m2)
plot(m2)
#op<-par(mfrow=c(3,1))
#term.plot(m2, se=TRUE)
#term.plot(m2, se=TRUE, what="sigma")
#term.plot(m2, se=TRUE, what="nu")
#par(op)
# the example from Venable and Ripley (2002)
library(MASS)
data(bacteria)
library(nlme)
summary(glmmPQL(y ~ trt + I(week > 2), random = ~ 1 | ID,
                family = binomial, data = bacteria))
s1 <- gamlss(y ~ trt + I(week > 2)+random(ID), family = BI, data = bacteria)
# the esimate of sigma
sqrt(s1$mu.coefSmo[[1]]$sig2)
# the esimate of random effect  sigma
sqrt(s1$mu.coefSmo[[1]]$tau2)
```

rc *Specify Random Coefficients In A GAMLSS Formula*

Description

Fits Random Coefficients In A GAMLSS Model.

Usage

```
rc(formula, lambda = NULL)
```

Arguments

formula	a model formula to specify the explanatory variable and the subject grouping factor e.g. formula=~x1|Subjects will fit random constant and slopes for x1 for each level of Subjects
lambda	a matrix specifying the variance-covariance matrix for the random coefficients e.g. lambda=diag{2,5}

Details

This is an experimental function and it is likely to change in the near future. For a chosen fixed variance-covariance matrix for the random coefficients, this function finds the MAP estimates for the constant and slope random effects for the different subjects

Value

The variables in the left side of the formula are returned with class "smooth", and attribute named "call" which is to be evaluated in the backfitting additive.fit() called by gamlss()

Warning

This is experimental

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/)

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07. .

See Also

random, ra, gamlss

---

refit *Refit a GAMLSS model*

Description

This function refits a GAMLSS model. It is useful when the algorithm has not converged after 20 outer iteration (the default value)

Usage

```
refit(object, ...)
```

Arguments

object	a GAMLSS fitted model which has not converged
...	for extra arguments

Details

This function is useful when the iterations have reach the maximum value set by the code(n.cyc) of the gamlss.control function and the model has not converged yet

Value

Returns a GAMLSS fitted model

Note

The function update does a very similar job

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, update.gamlss

Examples

```
data(aids)
h<-gamlss(y~poly(x,3)+qrt, family=PO, data=aids) #
refit(h)
rm(h)
```

`residuals.gamlss` *Extract Residuals from GAMLSS model*

Description

`residuals.gamlss` is the GAMLSS specific method for the generic function `residuals` which extracts the residuals for a fitted model. The abbreviated form `resid` is an alias for `residuals`.

Usage

```
## S3 method for class 'gamlss'
residuals(object, what = c("z-scores", "mu", "sigma", "nu", "tau"),
                          type = c("simple", "weighted", "partial"),
                          terms=NULL, ...)
```

Arguments

`object`	a GAMLSS fitted model
`what`	specify whether the standardized residuals are required, called here the "z-scores", or residuals for a specific parameter
`type`	the type of residual if residuals for a parameter are required
`terms`	if type is "partial" this specifies which term is required
`...`	for extra arguments

Details

The "z-scores" residuals saved in a GAMLSS object are the normalized (randomized) quantile residuals (see Dunn and Smyth, 1996). Randomization is only needed for the discrete family distributions, see also `rqres.plot`. Residuals for a specific parameter can be "simple" = (working variable - linear predictor), "weighted"= sqrt(working weights)*(working variable - linear predictor) or "partial"= (working variable - linear predictor)+contribution of specific terms.

Value

a vector or a matrix of the appropriate residuals of a GAMLSS model. Note that when weights are used in the fitting the length of the residuals can be different from `N` the length of the fitted values. Observations with weights equal to zero are not appearing in the residuals. Also observations with frequencies as weights will appear more than once according to their frequencies.

Note

The "weighted" residuals of a specified parameter can be zero and one if the square of first derivative have been used in the fitting of this parameter

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

print.gamlss, summary.gamlss, fitted.gamlss, coef.gamlss, residuals.gamlss, update.gamlss, plot.gamlss, deviance.gamlss, formula.gamlss

Examples

```
data(aids)
h<-gamlss(y~poly(x,3)+qrt, family=NBI, data=aids) #
plot(aids$x,resid(h))
plot(aids$x,resid(h,"sigma") )
rm(h)
```

---

ridge *Specify Ridge regression In A GAMLSS Formula*

Description

The function ri() allow the user to fit a ridge regression within GAMLSS. It allows the coefficients of a set of explanatory variables to be shrunk towards an overall zero, where the amount of shrinking depends either on lambda, or on the equivalent degrees of freedom (df). The function ri() can estimate lambda a local REML-algorithm.

Usage

```
ridge(X, df = NULL, lambda = NULL, order = 0)
ri(X, df = NULL, lambda = NULL, order = 0, start = 10)
```

Arguments

X	A standardized (mean=0, sd-1) matrix X of explanatory variables
df	the effective degrees of freedom df
lambda	the smoothing parameter lambda
order	the order of the difference applied to the coefficients
start	the lambda starting value if the Schall's EM-algorithm is used

Value

x is returned with class "smooth", with an attribute named "call" which is to be evaluated in the backfitting additive.fit() called by gamlss()

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk> and Paul Eilers

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

ra, gamlss

Examples

```
data(usair)
# standarized the x's
for (i in 2:7) usair[,i] <- (usair[,i]-mean(usair[,i]))/sd(usair[,i])
# create a matrix
usair$X <-as.matrix(usair[,c("x1","x2", "x3","x4", "x5", "x6")])
m1<- gamlss(y~ridge(X, df=3), data=usair, family=GA)
m2<- gamlss(y~ridge(X, lambda=10), data=usair, family=GA)
# plotting the coeficients as function of the df
df <- seq(0,6,0.5)
resmat<-matrix(0, nrow=length(df), ncol=6)
for (i in 1:length(df))
{
resmat[i,] <- gamlss(y~ridge(X, df=df[i]), data=usair)$mu.coefSmo[[1]][["coef"]]
}
```

```
colnames(resmat)<-colnames(usair$X)
plot(1:length(df), type="n", xlim=c(-.5,6.5), ylim=c(-23, 38))
for (i in 1:6)
{
lines(resmat[,i]~df, col="purple")
points(resmat[,i]~df)
}
lines(rep(0,length(df))~df, col="black")
# now estimating the lambda
 g1<-gamlss(y~ri(X), data=usair)
  m1$mu.coefSmo
```

`rqres.plot` *Plotting Randomized Quantile Residuals*

Description

This function plots QQ-plots of the normalized randomized quantile residuals (see Dunn and Smyth, 1996) for a model using a discrete GAMLSS family distribution.

Usage

```
rqres.plot(obj = NULL, howmany = 6, all = TRUE, save = FALSE, ...)
```

Arguments

`obj`	a fitted GAMLSS model object from a "discrete" type of family
`howmany`	The number of QQ-plots required up to ten i.e. `howmany=6`
`all`	if TRUE QQ-plots from `howmany` realizations are plotted. If FALSE then a single qq-plot of the median of the `howmany` realizations is plotted
`save`	If TRUE the median residuals can be saved
`...`	for extra arguments

Details

For discrete family distributions, the `gamlss()` function saves on exit one realization of randomized quantile residuals which can be plotted using the generic function `plot` which calls the `plot.gamlss`. Looking at only one realization can be misleading, so the current function creates QQ-plots for several realizations. The function allows up to 10 QQ-plots to be plotted. Occasionally one wishes to create a lot of realizations and then take a median of them (separately for each ordered value) to create a single median realization. The option `all` in combinations with the option `howmany` creates a QQ-plot of the medians of the normalized randomized quantile residuals. These 'median' randomized quantile residuals can be saved using the option (`save=TRUE`).

Value

If `save` it is TRUE then the vector of the median residuals is saved.

Warning

....

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References

Dunn, P. K. and Smyth, G. K. (1996) Randomised quantile residuals, *J. Comput. Graph. Statist.*, **5**, 236–244

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

plot.gamlss, gamlss

Examples

```
data(aids) # fitting a model from a discrete distribution
h<-gamlss(y~pb(x)+qrt, family=NBI, data=aids) #
plot(h)
# plot qq- plots from 6 realization of the randomized quantile residuals
rqres.plot(h)
# a qq-plot from the medians from 40 realizations
rqres.plot(h,howmany=40,all=FALSE) #
```

---

stepGAIC *Choose a model by GAIC in a Stepwise Algorithm*

Description

The function stepGAIC() performs stepwise model selection using a Generalized Akaike Information Criterion. The function stepGAIC() calls one of the two functions stepGAIC.VR() or stepGAIC.CH() depending on the argument additive. The function stepGAIC.VR() is based on the function stepAIC() given in the library MASS of Venables and Ripley (2002). The function stepGAIC.CH is based on the S function step.gam() (see Chambers and Hastie (1991)) and it is more suited for model with smoothing additive terms, (see below comments for the additive function pb()). Both functions have been adapted to work with gamlss objects. The main difference

for the user is the scope argument, see below. If the stepGAIC() is called with the argument additive=FALSE then the stepGAIC.VR() is called else the stepGAIC.CH().

While the functions stepGAIC.VR() and stepGAIC.CH() are used to build models for individual parameters of the distribution of the response variable, the functions stepGAICAll.A() and stepGAICAll.A() are building a model for all the parameters. Both the functions stepGAICAll.A() and stepGAICAll.B() are based on stepGAIC.VR(). The use two different strategies for selecting a appropriate final model. stepGAICAll.A() has the following strategy:

Stategy A:

i) build a model for mu using a forward approach.

ii) given the model for mu build a model for sigma (forward)

iii) given the models for mu and sigma build a model for nu (forward)

iv) given the models for mu, sigma and nu build a model for tau (forward)

v) given the models for mu, sigma, nu and tau check whether the terms for nu are needed using backward elimination.

vi) given the models for mu, sigma, nu and tau check whether the terms for sigma are needed (backward).

vii) given the models for mu, sigma, nu and tau check whether the terms for mu are needed (backward).

Note for this strategy to work the scope argument should be set appropriately.

stepGAICAll.B() uses the same procedure as the function stepGAIC.VR() but each term in the scope is fitted to ALL the parameters of the distribution, rather than the one specified by the argument what of stepGAIC.VR().

Usage

```
stepGAIC.VR(object, scope, direction = c("both", "backward", "forward"),
         trace = T, keep = NULL, steps = 1000, scale = 0,
         what = c("mu", "sigma", "nu", "tau"), k = 2, ...)

stepGAIC.CH(object, scope = gamlss.scope(model.frame(object)),
           direction = c("both", "backward", "forward"), trace = T, keep = NULL,
            steps = 1000, what = c("mu", "sigma", "nu", "tau"), k = 2, ...)


stepGAIC(object, scope = gamlss.scope(model.frame(object)),
          direction = c("both", "backward", "forward"),
          trace = T, keep = NULL, steps = 1000,
          what = c("mu", "sigma", "nu", "tau"), k = 2,
          additive = FALSE, ...)

stepGAICAll.A(object, scope = NULL, sigma.scope = NULL, nu.scope = NULL,
              tau.scope = NULL, mu.try = TRUE, sigma.try = TRUE,
              nu.try = TRUE, tau.try = TRUE, ...)
```

```
stepGAICAll.B(object, scope, direction = c("both", "backward", "forward"),
               trace = T, keep = NULL, steps = 1000, scale = 0, k = 2, ...)

stepTGD(object, scope, newdata, direction = c("both", "backward", "forward"),
              trace = T, keep = NULL, steps = 1000,
              what = c("mu", "sigma", "nu", "tau"), ...)
```

Arguments

object	an gamlss object. This is used as the initial model in the stepwise search.
scope	defines the range of models examined in the stepwise search. For the function stepAIC() this should be either a single formula, or a list containing components upper and lower, both formulae. See the details for how to specify the formulae and how they are used. For the function stepGAIC the scope defines the range of models examined in the step-wise search. It is a list of formulas, with each formula corresponding to a term in the model. A 1 in the formula allows the additional option of leaving the term out of the model entirely. +
direction	the mode of stepwise search, can be one of both, backward, or forward, with a default of both. If the scope argument is missing the default for direction is backward.
trace	if positive, information is printed during the running of stepAIC. Larger values may give more information on the fitting process.
keep	a filter function whose input is a fitted model object and the associated 'AIC' statistic, and whose output is arbitrary. Typically 'keep' will select a subset of the components of the object and return them. The default is not to keep anything.
steps	the maximum number of steps to be considered. The default is 1000 (essentially as many as required). It is typically used to stop the process early.
scale	scale is nor used in gamlss
what	which distribution parameter is required, default what="mu"
k	the multiple of the number of degrees of freedom used for the penalty. Only 'k = 2' gives the genuine AIC: 'k = log(n)' is sometimes referred to as BIC or SBC.
additive	if additive=TRUE then stepGAIC.CH is used else stepGAIC.CH, default value is FALSE
sigma.scope	scope for sigma if different to scope in stepGAICAll.A()
nu.scope	scope for nu if different to scope in stepGAICAll.A()
tau.scope	scope for tau if different to scope in stepGAICAll.A()
mu.try	The default value is is TRUE, set to FALSE if no model for mu is needed
sigma.try	The default value is TRUE, set to FALSE if no model for sigma is needed
nu.try	The default value is TRUE, set to FALSE if no model for nu is needed
tau.try	The default value is TRUE, set to FALSE if no model for tau is needed
newdata	The new data set where the Test Global Deviance (TGD) will be evaluated
...	any additional arguments to 'extractAIC'. (None are currently used.)

Details

The set of models searched is determined by the scope argument.

For the function stepGAIC.VR() the right-hand-side of its lower component is always included in the model, and right-hand-side of the model is included in the upper component. If scope is a single formula, it specifies the upper component, and the lower model is empty. If scope is missing, the initial model is used as the upper model.

Models specified by scope can be templates to update object as used by update.formula.

For the function stepGAIC.CH() each of the formulas in scope specifies a "regimen" of candidate forms in which the particular term may enter the model. For example, a term formula might be

~ x1 + log(x1) + cs(x1, df=3)

This means that x1 could either appear linearly, linearly in its logarithm, or as a smooth function estimated non-parametrically. Every term in the model is described by such a term formula, and the final model is built up by selecting a component from each formula.

The function gamlss.scope similar to the S gam.scope() in Chambers and Hastie (1991) can be used to create automatically term formulae from specified data or model frames.

The supplied model object is used as the starting model, and hence there is the requirement that one term from each of the term formulas of the parameters be present in the formula of the distribution parameter. This also implies that any terms in formula of the distribution parameter not contained in any of the term formulas will be forced to be present in every model considered.

When the smoother used in gamlss modeling belongs to the new generation of smoothers allowing the determination of the smoothing parameters automatically (i.e. pb(), cy()) then the function stepGAIC.VR() can be used for model selection (see example below).

The function stepTGD is a clone function of stepGAIC.VR() where the selection criterion is not anymore a GAIC but the Test Global Deviance, that is, the deviance (-2log(Likelihood)) evaluated at the test data rather than the training ones.

Value

the stepwise-selected model is returned, with up to two additional components. There is an '"anova"' component corresponding to the steps taken in the search, as well as a '"keep"' component if the 'keep=' argument was supplied in the call. The '"Resid. Dev"' column of the analysis of deviance table refers to a constant minus twice the maximized log likelihood

The function stepGAICAll.A() returns with a component "anovaAll" containing all the different anova tables used in the process.

Author(s)

Mikis Stasinopoulos based on functions in MASS library and in Statistical Models in S

References

Chambers, J. M. and Hastie, T. J. (1991). *Statistical Models in S*, Chapman and Hall, London.

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S*. Fourth edition. Springer.

See Also

gamlss.scope

Examples

```
data(usair)
# Note default of additive=FALSE
# fitting all variables linearly
mod1<-gamlss(y~., data=usair, family=GA)
# find the best subset for the mu
mod2<-stepGAIC(mod1)
mod2$anova
# find the best subset for sigma
mod3<-stepGAIC(mod2, what="sigma", scope=~x1+x2+x3+x4+x5+x6)
mod3$anova
# find the best model using pb() smoother
#only three variables are used here for simplicity
mod10 <-gamlss(y~1, data=usair, family=GA)
mod20<-stepGAIC(mod10, scope=list(lower=~1, upper=~pb(x1)+pb(x2)+pb(x5)))
edf(mod20)
# x1 and x2 enter linearly
# now use the stepGAIC.CH function
# creating a scope from the usair model frame
gs<-gamlss.scope(model.frame(y~x1+x2+x3+x4+x5+x6, data=usair))
gs
mod4<-gamlss(y~1, data=usair, family=GA)
mod5<-stepGAIC(mod4,gs, additive=TRUE)
mod5$anova
mod6<-stepGAIC(mod5, what="sigma", scope=~x1+x2+x3+x4+x5+x6)
mod6$anova
mod6
# now stepGAICAll.A
mod7<-stepGAICAll.A(mod4, scope=list(lower=~1,upper=~x1+x2+x3+x4+x5+x6))
# now  stepGAICAll.A
mod8<-stepGAICAll.B(mod4, scope=list(lower=~1,upper=~x1+x2+x3+x4+x5+x6))
# now stepTGD()
data(aep)
# sampling from the data
rand <- sample(2, dim(aep)[1], replace=TRUE, prob=c(0.6,0.4))
# the proportions in the sample
table(rand)/dim(aep)[1]
oldaep<-aep[rand==1,]
```

```
newaep<-aep[rand==2,]
m0<-gamlss(y~ward+year+loglos, data=oldaep, family=BB)
#checking mu
m1 <- stepTGD(m0, newdata=newaep)
#checking sigma
m2 <- stepTGD(m0,scope=~ward+year, newdata=newaep, what="sigma")
```

---

summary.gamlss *Summarizes a GAMLSS fitted model*

Description

summary.gamlss is the GAMLSS specific method for the generic function summary which summarize objects returned by modelling functions.

Usage

```
## S3 method for class 'gamlss'
summary(object, type = c("vcov", "qr"), ...)
```

Arguments

object	a GAMLSS fitted model
type	the default value vcov uses the vcov() method for gamlss to get the variance-covariance matrix of the estimated beta coefficients, see details below. The alternative qr is the original method used in gamlss to estimated the standard errors but it is not reliable since it do not take into the account the inter-correlation between the distributional parameters mu, sigma, nu and tau.
...	for extra arguments

Details

Using the default value type="vcov", the vcov() method for gamlss is used to get the variance covariance matrix (and consequently the standard errors)of the beta parameters. The variance covariance matrix is calculated using the inverse of the numerical second derivatives of the observed information matrix. This is a more reliable method since it take into the account the inter-correlation between the all the parameters. The type="qr" assumes that the parameters are fixed at the estimated values. Note that both methods are not appropriate and should be used with caution if smoothing terms are used in the fitting.

Value

Print summary of a GAMLSS object

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk> and Calliope Akantziliotou

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2003) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss, deviance.gamlss, fitted.gamlss

Examples

```
data(aids)
h<-gamlss(y~poly(x,3)+qrt, family=PO, data=aids) #
summary(h)
rm(h)
```

`term.plot` *Plot regression terms for a specified parameter of a GAMLSS object*

Description

Plots regression terms against their predictors, optionally with standard errors and partial residuals added. It is almost identical to the R function `termplot` suitable changed to apply to GAMLSS objects.

Usage

```
term.plot(object, what = c("mu", "sigma", "nu", "tau"), data = NULL,
          envir = environment(formula(object)),
          partial.resid = FALSE, rug = FALSE,
          terms = NULL, se = FALSE, xlabs = NULL, ylabs = NULL,
          main = NULL, col.term = 2, lwd.term = 1.5,
          col.se = "orange", lty.se = 2, lwd.se = 1,
          col.res = "gray", cex = 1, pch = par("pch"),
          col.smth = "darkred", lty.smth = 2,
          span.smth = 2/3,
          ask = interactive() && nb.fig < n.tms && .Device != "postscript",
          use.factor.levels = TRUE, smooth = NULL,  ylim  = NULL, ...)
```

Arguments

`object`	a GAMLSS object
`what`	the required parameter of the GAMLSS distribution
`data`	data frame in which variables in `object` can be found
`envir`	environment in which variables in `object` can be found
`partial.resid`	logical; should partial residuals be plotted?
`rug`	add rugplots (jittered 1-d histograms) to the axes?
`terms`	which terms to plot (default 'NULL' means all terms)
`se`	plot pointwise standard errors?
`xlabs`	vector of labels for the x axes
`ylabs`	vector of labels for the y axes
`main`	logical, or vector of main titles; if 'TRUE', the model's call is taken as main title, 'NULL' or 'FALSE' mean no titles.
`col.term, lwd.term`	color and width for the "term curve", see 'lines'.
`col.se, lty.se, lwd.se`	color, line type and line width for the "twice-standard-error curve" when 'se = TRUE'.
`col.res, cex, pch`	color, plotting character expansion and type for partial residuals, when 'partial.resid = TRUE', see 'points'.
`lty.smth,col.smth, span.smth`	Passed to 'smooth'
`ask`	logical; if 'TRUE', the user is asked before each plot, see 'par(ask=.)'.
`use.factor.levels`	Should x-axis ticks use factor levels or numbers for factor terms?
`smooth`	'NULL' or a function with the same arguments as 'panel.smooth' to draw a smooth through the partial residuals for non-factor terms
`ylim`	for setting the y axis limits
`...`	other graphical parameters

Details

The function uses the `lpred` function of GAMLSS. The 'data' argument should rarely be needed, but in some cases 'termplot' may be unable to reconstruct the original data frame. Using 'na.action=na.exclude' makes these problems less likely. Nothing sensible happens for interaction terms.

Value

a plot of fitted terms.

Author(s)

Mikis Stasinopoulos based on the existing termplot() function

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

termplot

Examples

```
data(aids)
a<-gamlss(y~pb(x)+qrt,data=aids,family=NBI)
term.plot(a, se=TRUE)
rm(a)
```

---

update.gamlss *Update and Re-fit a GAMLSS Model*

Description

`update.gamlss` is the GAMLSS specific method for the generic function `update` which updates and (by default) refits a GAMLSS model.

Usage

```
## S3 method for class 'gamlss'
update(object, formula., ..., what = c("mu", "sigma", "nu", "tau", "All"),
               evaluate = TRUE)
```

Arguments

`object`	a GAMLSS fitted model
`formula.`	the formula to update
`...`	for updating argument in `gamlss()`
`what`	the parameter in which the formula needs updating for example "mu", "sigma", "nu" "tau" or "All". If "All" all the formulae are updated. Note that the `what` argument has an effect only if only if the argument `formula.` is set
`evaluate`	whether to evaluate the call or not

Value

Returns a GAMLSS call or fitted object.

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

print.gamlss, summary.gamlss, fitted.gamlss, coef.gamlss, residuals.gamlss, plot.gamlss, deviance.gamlss, formula.gamlss

Examples

```
data(aids)
# fit a poisson model
h.po <-gamlss(y~pb(x)+qrt, family=PO, data=aids)
# update with a negative binomial
h.nb <-update(h.po, family=NBI)
# update the smoothing
h.nb1 <-update(h.nb,~cs(x,8)+qrt)
# remove qrt
h.nb2 <-update(h.nb1,~.-qrt)
# put back qrt take log of y and fit a normal distribution
h.nb3 <-update(h.nb1,log(.)~.+qrt, family=NO)
# verify that it is the same
h.no<-gamlss(log(y)~cs(x,8)+qrt,data=aids )
```

VGD *Validation (or test) data global deviance*

Description

These function are useful for model selection. The function VGD helps to validate a GAMLSS model by randomly splitting the data into training and validation sets. It minimizes the global deviance for the training data set and then uses the validation set to calculate the prediction global deviance. The function VGD1 is identical to VGD but the output is a list rather than a values as in VGD. The function VGD2 is identical to VGD1 but it takes as argument the new data, (newdata), rather than a factor which split the combined data in two as in VGD or VGD1. The function TGD takes a fitted gamlss object and produce the test global deviance for the new data. The resulting prediction errors can be used for selecting the distribution of the model, the terms in the model or degrees of freedom for smoothing terms.

Usage

```
VGD(formula = NULL, sigma.formula = ~1, nu.formula = ~1, tau.formula = ~1,
    data = NULL, family = NO,
    control = gamlss.control(trace = FALSE), rand = NULL, ...)
VGD1(formula = NULL, sigma.formula = ~1, nu.formula = ~1, tau.formula = ~1,
    data = NULL, family = NO,
    control = gamlss.control(trace = FALSE), rand = NULL, ...)
VGD2(formula = NULL, sigma.formula = ~1, nu.formula = ~1, tau.formula = ~1,
    data = NULL, family = NO,
    control = gamlss.control(trace = FALSE),
    newdata = NULL, ...)
TGD(object, newdata = NULL, ...)
```

Arguments

object	a gamlss object
formula	a gamlss formula for mu (including the response on the left)
sigma.formula	a formula for sigma
nu.formula	a formula for nu
tau.formula	a formula for tau
data	the data set used for the fitting
newdata	the data set to be used for validation ot test
family	a gamlss.family object
control	gamlss.control to be passed to gamlss
rand	a random vector of one and two indicating whether is the training set (1) or the validation set (2) i.e. created in advance using something like rand <- sample(2, N, replace=T, prob=c(0.6,0.4)) where N is the length of the data
...	for extra arguments to be passed in the gamlss fit

Value

The function VGD returns a validated global deviance, that is, the global deviance evaluated at the validation data. The functions VGD1, VGD2 and TGD are returning a list with component oldGD, newGD, oldPE, newPE. oldGD is the deviance of the training data, newGD is the deviance for the validation or (test) data. oldPE is the mean prediction error for the old data (oldGD divided by the number of observations in the training data). newPEis the mean prediction error for the new data, (newGD divided by the number of observations in the validation (or test) data).

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk> and Bob Rigby <r.rigby@londonmet.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

See Also

gamlss.family, gamlss, deviance.gamlss

Examples

```
data(abdom)
# generate the random split of the data
rand <- sample(2, 610, replace=TRUE, prob=c(0.6,0.4))
# the proportions in the sample
table(rand)/610
# VGD needs a factor
VGD(y~pb(x,df=2),sigma.formula=~pb(x,df=1), data=abdom, family=LO, rand=rand)
# VGD1 needs a factor and has different output
VGD1(y~pb(x,df=2),sigma.formula=~pb(x,df=1), data=abdom, family=LO, rand=rand)
# using VGD2 needs two different sets of data
VGD2(y~pb(x,df=2),sigma.formula=~pb(x,df=1), data=abdom[rand==1,], family=LO,
      newdata=abdom[rand==2,] )
#fit in the old data
olddata<-abdom[rand==1,]
m1<-gamlss(y~pb(x,df=2),sigma.formula=~pb(x,df=1), data=olddata, family=LO)
# get the global deviance in the new data
newdata<-abdom[rand==2,]
TGD(m1, newdata=newdata)
```

wp *Worm plot*

Description

Provides single or multiple worm plots for GAMLSS fitted objects. This is a diagnostic tool for checking the residuals within different ranges (by default not overlapping) of the explanatory variable

Usage

```
wp(object=NULL, xvar = NULL,  resid = NULL, n.inter = 4,
    xcut.points = NULL,
   overlap = 0, xlim.all = 4, xlim.worm = 3.5,
   show.given = TRUE, line = TRUE,
   ylim.all = 12 * sqrt(1/length(resid)),
   ylim.worm = 12 * sqrt(n.inter/length(resid)),
   cex = 1, pch = 21, ...)
```

Arguments

object	a GAMLSS fitted object or any other fitted model where the resid() method works (preferably it should be quantile residuals)
xvar	the explanatory variable against which the worm plots will be plotted
resid	if object is missing this argument can be used to specify the residual vector (again it should a quantile residuals or it be assumed to come from a normal distribution)
n.inter	the number of intervals in which the explanatory variable xvar will be cut
xcut.points	the x-axis cut off points e.g. c(20,30). If xcut.points=NULL then the n.inter argument is activated
overlap	how much overlapping in the xvar intervals. Default value is overlap=0 for non overlapping intervals
xlim.all	for the single plot, this value is the x-variable limit, default is xlim.all=4
xlim.worm	for multiple plots, this value is the x-variable limit, default is xlim.worm=3.5
show.given	whether to show the x-variable intervals in the top of the graph, default is show.given=TRUE
line	whether to plot the polynomial line in the worm plot, default value is line=TRUE
ylim.all	for the single plot, this value is the y-variable limit, default value is ylim.all=12*sqrt(1/length(fitte
ylim.worm	for multiple plots, this values is the y-variable limit, default value is ylim.worm=12*sqrt(n.inter/lengt
cex	the cex plotting parameter with default cex=1
pch	the pch plotting parameter with default pch=21
...	for extra arguments

Details

If the xvar argument is not specified then a single worm plot is used. In this case a worm plot is a detrended normal QQ-plot so departure from normality is highlighted. If the xvar is specified then we have as many worm plot as n.iter. In this case the x-variable is cut into n.iter intervals with an equal number observations and detrended normal QQ (i.e. worm) plots for each interval are plotted. This is a way of highlighting failures of the model within different ranges of the explanatory variable. The fitted coefficients from fitting cubic polynomials to the residuals (within each x-variable interval) can be obtain by e.g. coeffs<-wp(model1,xvar=x,n.iner=9). van Buuren *et al.* (2001) used these residuals to identify regions (intervals) of the explanatory variable within which the model does not fit adequately the data (called "model violation")

Value

For multiple plots the xvar intervals and the coefficients of the fitted cubic polynomials to the residuals (within each xvar interval) are returned.

Author(s)

Mikis Stasinopoulos and Bob Rigby

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, 1-38.

Stasinopoulos D. M., Rigby R.A. and Akantziliotou C. (2006) Instructions on how to use the GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see also http://www.gamlss.org/).

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, http://www.jstatsoft.org/v23/i07.

van Buuren and Fredriks M. (2001) Worm plot: simple diagnostic device for modelling growth reference curves. *Statistics in Medicine*, **20**, 1259–1277

See Also

gamlss, plot.gamlss

Examples

```
data(abdom)
a<-gamlss(y~pb(x),sigma.fo=~pb(x,1),family=LO,data=abdom)
wp(a)
coeff1<-wp(a,abdom$x)
coeff1
rm(a,a1)
```

Index