

Package ‘quantregGrowth’

February 26, 2021

Type Package

Title Growth Charts via Smooth Regression Quantiles with Automatic Smoothness Estimation and Additive Terms

Version 1.0-0

Date 2021-02-26

Maintainer Vito M. R. Muggeo <vito.muggeo@unipa.it>

Description Fits non-crossing regression quantiles as a function of linear covariates and multiple smooth terms via B-splines with L1-norm difference penalties. The smoothing parameters are estimated as part of the model fitting. Monotonicity and concavity constraints on the fitted curves are allowed. See Muggeo, Sciandra, Tomasello and Calvo (2013) <doi:10.1007/s10651-012-0232-1> and <doi:10.13140/RG.2.2.12924.85122> for some code examples. Smoothing parameter selection with additive terms is discussed in Muggeo and others (2020) <doi:10.1177/1471082X20929802>.

Depends quantreg, splines

License GPL

NeedsCompilation no

Author Vito M. R. Muggeo [aut, cre] (<<https://orcid.org/0000-0002-3386-4054>>)

Repository CRAN

Date/Publication 2021-02-26 13:50:02 UTC

R topics documented:

quantregGrowth-package	2
gcrq	3
growthData	8
ncross.rq.fitXB	8
plot.gcrq	10
predict.gcrq	13
print.gcrq	14
ps	15
summary.gcrq	17
vcov.gcrq	18

quantregGrowth-package

Growth Charts via Smooth Regression Quantiles with Automatic Smoothness Estimation and Additive Terms

Description

Fits non-crossing regression quantiles as a function of linear covariates and smooth terms via B-splines with difference penalties. Automatic smoothness estimation for several spline terms is allowed.

Details

Package: quantregGrowth
Type: Package
Version: 1.0-0
Date: 2021-02-26
License: GPL

Package `quantregGrowth` allows estimation of growth charts via quantile regression. Given a set of percentiles, `gcrq` estimates non-crossing quantile curves as a flexible function of quantitative covariates (typically age), and possibly additional linear terms. To ensure flexibility, B-splines with a difference L_1 penalty are employed to estimate nonparametrically the curves; additionally monotonicity and concavity constraints may be also set. Multiple smooth terms are allowed and the amount of smoothness for each term is efficiently included in the model fitting algorithm, see Muggeo et al. (2020). `plot.gcrq` displays the fitted lines along with observations and pointwise confidence intervals.

Author(s)

Vito M.R. Muggeo

Maintainer: Vito M.R. Muggeo <vito.muggeo@unipa.it>

References

Muggeo VMR, Torretta F, Eilers PHC, Sciandra M, Attanasio M (2020). Multiple smoothing parameters selection in additive regression quantiles, *Statistical Modelling*, to appear.

Muggeo VMR, Sciandra M, Tomasello A, Calvo S (2013). Estimating growth charts via nonparametric quantile regression: a practical framework with application in ecology, *Environ Ecol Stat*, **20**, 519-531.

Muggeo VMR (2018). Using the R package `quantregGrowth`: some examples.
<https://www.researchgate.net/publication/323573492>

Some references on growth charts (the first two papers employ the so-called LMS method)

Cole TJ, Green P (1992) Smoothing reference centile curves: the LMS method and penalized likelihood. *Statistics in Medicine* **11**, 1305-1319.

Rigby RA, Stasinopoulos DM (2004) Smooth centile curves for skew and kurtotic data modelled using the Box-Cox power exponential distribution. *Statistics in Medicine* **23**, 3053-3076.

Wei Y, Pere A, Koenker R, He X (2006) Quantile regression methods for reference growth charts. *Statistics in Medicine* **25**, 1369-1382.

Some references on regression quantiles

Koenker R (2005) Quantile regression. Cambridge University Press, Cambridge.

Cade BS, Noon BR (2003) A gentle introduction to quantile regression for ecologists. *Front Ecol Environ* **1**, 412-420.

See Also

[gcrq](#), [rq](#) in package `quantreg`

Examples

```
#see ?gcrq for some examples
```

gcrq	<i>Growth charts regression quantiles with automatic smoothness estimation</i>
------	--

Description

Modelling unspecified nonlinear relationships between covariates and quantiles of the response conditional distribution. Typical example is estimation nonparametric growth charts (via quantile regression). Quantile curves are estimated via B-splines with a L_1 penalty on the spline coefficient differences, while non-crossing and possible monotonicity and concavity restrictions are set to obtain estimates more biologically plausible. Linear terms can be specified in the model formula. Multiple smooth terms with automatic selection of corresponding smoothing parameters are allowed.

Usage

```
gcrq(formula, tau=c(.1,.25,.5,.75,.9), data, subset, weights, na.action,
      transf=NULL, y=TRUE, n.boot=0, eps=0.0001, display=FALSE,
      method=c("REML","ML"), df.opt=2, lambda0=.1, h=0.8, lambda.max=2000,
      tol=0.01, it.max=20, single.lambda=TRUE, foldid=NULL, nfolds=10,
      lambda.ridge=0, contrasts=NULL, ...)
```

Arguments

formula	a standard R formula to specify the response in the left hand side, and the co-variates in the right hand side. See Details.
tau	a numeric vector to specify the quantile curves of interest. Default to probability values (.1, .25, .5, .75, .9).
data	the dataframe where the variables required by the formula, subset and weights arguments are stored.
subset	optional. A vector specifying a subset of observations to be used in the fitting process.
weights	optional. A numeric vector specifying weights to be assigned to the observations in the fitting process. Currently unimplemented.
na.action	a function which indicates how the possible 'NA's are handled.
transf	an optional character string (with "y" as argument) meaning a function to apply to the response variable before fitting. E.g. "log(y+0.1)". It can be useful to guarantee fitted values within a specified range. If provided, the resulting object fit will include the corresponding inverse (numerically computed) to be used to back transform predictions (see argument transform in predict.gcrq and plot.gcrq).
y	logical. If TRUE (default) the returned object includes also the responses vector.
n.boot	Number of nonparametric (cases resampling) bootstrap samples to be used. The covariance matrix estimates is obtained as empirical covariance matrix of the bootstrap distributions. Notice that the smoothing parameter (if relevant) is assumed fixed. Namely it does change throughout the bootstrap replicates. Set n.boot>0 if you plan to plot the fitted quantiles along with pointwise confidence intervals.
eps	A small positive constant to ensure noncrossing curves (i.e. the minimum distance between two consecutive curves). Use it at your risk! If eps is large, the resulting fitted quantile curves could appear unreasonable.
display	Logical. Should the iterative process be printed? Ignored if all the smoothing parameters specified in ps terms are fixed.
method	character, "ML" or "REML" affecting the smoothing parameter estimation. Default is "REML" which appears to provide better performance in simulation studies. Ignored if no smoothing parameter has to be estimated.
df.opt	How the model and term-specific degrees of freedom have to be computed. df.opt=1 means via the null penalized coefficients, and df.opt=2 via the trace of the approximate hat matrix. Ignored if no smoothing parameter has to be estimated.
lambda0	the starting value for the lambdas to be estimated. Ignored if all the smoothing parameters specified in ps terms are fixed.
h	The step halving factor affecting estimation of the smoothing parameters. Lower values lead to slower updates in the lambda values. Ignored if all the smoothing parameters specified in ps terms are fixed.
lambda.max	The upper bound for lambda estimation. Ignored if all the smoothing parameters specified in ps terms are fixed.

<code>tol</code>	The tolerance value to declare convergence. Ignored if all the smoothing parameters specified in <code>ps</code> terms are fixed.
<code>it.max</code>	The maximum number of iterations in <code>lambdas</code> estimation. Ignored if all the smoothing parameters specified in <code>ps</code> terms are fixed.
<code>single.lambda</code>	Logical. Should the smoothing parameter to be the same across the quantile curves being estimated? Ignored when just a single quantile curve is being estimated. Currently only <code>single.lambda=TRUE</code> is allowed.
<code>foldid</code>	optional. A numeric vector identifying the group labels to perform cross validation to select the smoothing parameter. Ignored if the <code>lambda</code> argument in <code>ps()</code> is not a vector.
<code>nfolds</code>	optional. If <code>foldid</code> is not provided, it is scalar specifying the number of ‘folds’ (groups) which should be used to perform cross validation to select the smoothing parameter. Default to 10, but it is ignored if the <code>lambda</code> argument in <code>ps()</code> is not a vector.
<code>lambda.ridge</code>	logical. If <code>TRUE</code> (default) and there is a single smooth term wherein <code>lambda</code> is a vector, the returned object includes also the matrix <code>cv</code> having number of rows equal Ignored if the <code>lambda</code> argument in <code>ps()</code> is <i>not</i> a vector.
<code>contrasts</code>	an optional list. See argument <code>contrasts.arg</code> in <code>model.matrix.default</code> .
<code>...</code>	further arguments.

Details

The function fits regression quantiles at specified percentiles given in `tau` as a function of covariates specified in the `formula` argument. The `formula` may include linear terms and one or several `ps` terms to model nonlinear relationships with quantitative covariates, usually age in growth charts. When the `lambda` argument in `ps()` is a negative scalar, the smoothing parameter is estimated iteratively as discussed in Muggeo et al. (2020). If a positive scalar, it represents the actual smoothing parameter value. When it is a vector, ‘K-fold’ cross validation is performed to select the ‘optimal’ `lambda` value and the model is fitted at such selected `lambda` value. To select the smoothing parameter via CV, `foldid` or `nfolds` may be supplied. If provided `foldid` overwrites `nfolds`, otherwise `foldid` is obtained via random extraction, namely `sample(rep(seq(nfolds), length = n))`. However selection of smoothing parameter via CV is allowed only with a unique `ps()` term in the formula.

Value

This function returns an object of class `gcrq`, that is a list with the following components

<code>coefficients</code>	The matrix of estimated regression parameters; the number of columns equals the number of the fitted quantile curves.
<code>x</code>	the design matrix of the final fit (including the dummy rows used by penalty).
<code>df</code>	a vector reporting the <code>df</code> values for each quantile curve. See the section ‘Warning’ below.
<code>rho</code>	a vector including the values of the objective functions at the solution for each quantile curve.
<code>fitted.values</code>	a matrix of fitted quantiles (a column for each <code>tau</code> value)

<code>residuals</code>	a matrix of residuals (a column for each tau value)
<code>D.matrix</code>	the penalty matrix (multiplied by the smoothing parameter value).
<code>D.matrix.nolambda</code>	the penalty matrix.
<code>pLin</code>	number of linear covariates in the model.
<code>info.smooth</code>	some information on the smoothing term (if included in the formula via <code>ps</code>).
<code>BB</code>	further information on the smoothing term (if present in the formula via <code>ps</code>), including stuff useful for plotting via <code>plot.gcrq()</code> .
<code>Bderiv</code>	if the smooth term is included, the first derivative of the B spline basis.
<code>boot.coef</code>	The array including the estimated coefficients at different bootstrap samples (provided that <code>n.boot>0</code> has been set).
<code>y</code>	the response vector (if <code>gcrq()</code> has been called with <code>y=TRUE</code>).
<code>contrasts</code>	the contrasts used, when the model contains a factor.
<code>xlevels</code>	the levels of the factors (when included) used in fitting.
<code>taus</code>	a vector of values between 0 and 1 indicating the estimated quantile curves.
<code>call</code>	the matched call.

Warning

Currently, standard errors are obtained via the nonparametric bootstrap (case resampling) which ignores uncertainty in the smoothing parameter selection.

Note

This function is based upon the package `quantreg` by R. Koenker. Currently methods specific to the class "gcrq" are `print.gcrq`, `summary.gcrq`, `vcov.gcrq`, `plot.gcrq`, and `predict.gcrq`.

If the sample is not large, and/or the basis rank is large (i.e. a large number of columns) and/or there are relatively few distinct values in the covariate distribution, the fitting algorithm may fail returning error messages like the following

```
> Error info = 20 in stepy2: singular design
```

To remedy it, it suffices to change some arguments in `ps()`: to decrease `ndx` or `deg` (even by a small amount) or to increase (even by a small amount) the `lambda` value. Sometimes even by changing slightly the tau probability value (for instance from 0.80 to 0.79) can bypass the aforementioned errors.

Author(s)

Vito M. R. Muggeo, <vito.muggeo@unipa.it>

References

V.M.R. Muggeo, F. Torretta, P.H.C. Eilers, M. Sciandra, M. Attanasio (2020). Multiple smoothing parameters selection in additive regression quantiles, *Statistical Modelling*, to appear.

V. M. R. Muggeo, M. Sciandra, A. Tomasello, S. Calvo (2013). Estimating growth charts via nonparametric quantile regression: a practical framework with application in ecology, *Environ Ecol Stat*, 20, 519-531.

V. M. R. Muggeo (2018). Using the R package `quantregGrowth`: some examples. <https://www.researchgate.net/publication/323573492>

See Also

[ps](#), [plot.gcrq](#), [predict.gcrq](#)

Examples

```
## Not run:
#An additive examples.. from ?mgcv::gam
d<-mgcv::gamSim(n=200, eg=1)
o<-gcrq(y ~ ps(x0) + ps(x1)+ ps(x2) + ps(x3), data=d, tau=.5, n.boot=50)
plot(o, res=TRUE, col=2, conf.level=.9, shade=TRUE, split=TRUE)

#some simple examples involving just a single smooth
data(growthData) #load data
tauss<-seq(.1,.9,by=.1) #fix the percentiles of interest

m1<-gcrq(y~ps(x), tau=tauss, data=growthData) #lambda estimated..

m2<-gcrq(y~ps(x, lambda=0), tau=tauss, data=growthData) #unpenalized.. very wiggly curves
#strongly penalized models
m3<-gcrq(y~ps(x, lambda=1000, d=2), tau=tauss, data=growthData) #linear
m4<-gcrq(y~ps(x, lambda=1000, d=3), tau=tauss, data=growthData) #quadratic

#penalized model with monotonicity restrictions
m5<-gcrq(y~ps(x, mon=1, lambda=10), tau=tauss, data=growthData)

#monotonicity constraints,lambda estimated, and varying penalty
m6<-gcrq(y~ps(x, mon=1, lambda=10, var.pen="(1:k)"), tau=tauss, data=growthData)

par(mfrow=c(2,3))
plot(m1, pch=20, res=TRUE)
plot(m2, pch=20, res=TRUE)
plot(m3, add=TRUE, lwd=2)
plot(m4, pch=20, res=TRUE)
plot(m5, pch=20, res=TRUE, legend=TRUE, col=2)
plot(m6, add=TRUE, lwd=2, col=3)

#select lambda via 'K-fold' CV (only with a single smooth term)
m7<-gcrq(y~ps(x, lambda=seq(0.02,100,l=20)), tau=tauss, data=growthData)
par(mfrow=c(1,2))
plot(m7, cv=TRUE) #display CV score versus lambda values
plot(m7, res=TRUE, grid=list(x=5, y=8), col=4) #fitted curves at the best lambda value

## End(Not run)
```

growthData	<i>Simulated data to illustrate capabilities of the package</i>
------------	---

Description

The growthData data frame has 200 rows and 3 columns.

Usage

```
data(growthData)
```

Format

A data frame with 200 observations on the following 3 variables.

x the supposed 'age' variable.

y the supposed growth variable (e.g. weight).

z an additional variable to be considered in the model.

Details

Simulated data to illustrate capabilities of the package.

Examples

```
data(growthData)
with(growthData, plot(x,y))
```

ncross.rq.fitXB	<i>Estimation of noncrossing regression quantiles with monotonicity restrictions.</i>
-----------------	---

Description

These are internal functions of package quantregGrowth and should be not called by the user.

Usage

```
ncross.rq.fitXB(y, x, B=NULL, X=NULL, taus, monotone=FALSE, concave=FALSE,
  nomiBy=NULL, byVariabili=NULL, ndx=10, deg=3, dif=3, lambda=0, eps=.0001,
  var.pen=NULL, penMatrix=NULL, lambda.ridge=0,
  dropcList=FALSE, decomList=FALSE, dropvcList=FALSE, ...)
```

```
ncross.rq.fitX(y, X = NULL, taus, lambda.ridge = 0, eps = 1e-04, ...)
```

```
gcrq.rq.cv(y, B, X, taus, monotone, concave, ndx, lambda, deg, dif, var.pen=NULL,
  penMatrix=NULL, lambda.ridge=0, dropcList=FALSE, decomList=FALSE,
  dropvcList=FALSE, nfolds=10, foldid=NULL, eps=.0001, ...)
```


Arguments

y	the responses vector. see gcrq
x	the covariate supposed to have a nonlinear relationship.
B	the B-spline basis.
X	the design matrix for the linear parameters.
taus	the percentiles of interest.
monotone	numerical value (-1/0/+1) to define a non-increasing, unconstrained, and non-decreasing flexible fit, respectively.
concave	numerical value (-1/0/+1) to possibly define concave or convex fits.
nomiBy	useful for VC models (when B is not provided).
byVariabili	useful for VC models (when B is not provided).
ndx	number of internal intervals within the covariate range, see ndx in ps .
deg	spline degree, see ps .
dif	difference order of the spline coefficients in the penalty term.
lambda	smoothing parameter value(s), see lambda in ps .
eps	tolerance value.
var.pen	Varying penalty, see ps .
penMatrix	Specified penalty matrix, see pen.matrix in ps .
lambda.ridge	a (typically very small) value, see lambda.ridge gcrq .
dropcList	see dropc in ps .
decomList	see decompose in ps .
dropvcList	see ps .
foldid	vector (optional) to perform cross validation, see the same arguments in gcrq .
nfolds	number of folds for crossvalidation, see the same arguments in gcrq .
cv	returning cv scores; see the same arguments in gcrq .
...	optional.

Details

These functions are called by [gcrq](#) to fit growth charts based on regression quantiles with non-crossing and monotonicity restrictions. The computational methods are based on the package [quantreg](#) by R. Koenker and details are described in the reference paper.

Value

A list of fit information.

Author(s)

Vito M. R. Muggeo

See Also[gcrq](#)**Examples**

##See ?gcrq

`plot.gcrq`*Plot method for gcrq objects*

Description

Displaying the estimated growth charts from a gcrq fit.

Usage

```
## S3 method for class 'gcrq'
plot(x, term=NULL, add = FALSE, res = FALSE, conf.level=0, interc=TRUE,
     legend = FALSE, select.tau, deriv = FALSE, cv = FALSE, transf=NULL,
     lambda0=FALSE, shade=FALSE, overlap=NULL, rug=FALSE, n.points=100,
     edf.ylab=NULL, overall.eff=TRUE, grid=NULL, smoos=NULL, split=FALSE, ...)
```

Arguments

<code>x</code>	a fitted "gcrq" object.
<code>term</code>	the smooth variable name entering the model via <code>ps</code> . Relevant fitted quantile curves (as specified by <code>select.tau</code>) will be plotted. If <code>NULL</code> , all smooth terms are plotted according to the <code>split</code> argument.
<code>interc</code>	Should the smooth term be plotted along with the intercept (provided it is included in the model)? If the smooth term is a varying coefficient, <code>interc</code> refers to its intercept, and not the overall model intercept. Of course such argument is ignored if the smooth term has been called via <code>ps(, dropc=FALSE)</code> and the plot always includes implicitly the 'intercept'.
<code>add</code>	logical. If <code>TRUE</code> the fitted quantile curves are added on the current plot.
<code>res</code>	logical. If <code>TRUE</code> 'partial residuals' are also displayed on the plot. Borrowing terminology from GLM, partial residuals for covariate X_j are defined as fitted values corresponding to X_j + residuals (from the actual fit). If there is a single covariate, the partial residuals correspond to observed data. If multiple quantile curves have been estimated, the fitted values coming from the 'middle' quantile curve are employed to compute the partial residuals. 'Middle' means 'corresponding to the τ_k closest to 0.50'. I don't know if that is the best choice.
<code>conf.level</code>	logical. If larger than zero, pointwise confidence intervals for the fitted quantile curve are also shown (at the confidence level specified by <code>conf.level</code>). Such confidence intervals are independent of the possible intercept accounted for via the intercept argument.

legend	logical. If TRUE a legend is drawn on the right side of the plot.
select.tau	an optional numeric vector to draw only some of the fitted quantiles. Percentile values or integers 1 to length(tau) may be supplied.
deriv	logical. If TRUE the first derivative of the curve is displayed.
cv	logical. If TRUE and the "gcrq" object contains a single smooth term wherein lambda has been selected via CV, then the cross-validation scores against the lambda values are plotted.
transf	An optional character string (with "y" as argument) meaning a function to apply to the predicted values (and possibly residuals) before plotting. E.g. "(exp(y)-0.1)". If NULL (default) it is taken as the inverse of function transf (*if*) supplied in gcrq. See argument "transf" in gcrq(). If transf has been specified in gcrq(), use transf="y" to force plotting on the transformed scale, i.e. without back transforming.
lambda0	logical. If cv=TRUE, should the CV plot include also the first CV value? Usually the first CV value is at lambda=0, and typically it is much bigger than the other values making the plot not easy to read. Default to FALSE not to display the first CV value in the plot.
shade	logical. If TRUE and conf.level>0, the pointwise confidence intervals are portrayed via shaded areas.
overlap	If provided and different from NULL, it represents the abscissa value (on the covariate scale) where the legend (i.e. the probability values) of each curve is set. If unspecified (i.e. overlap=NULL), the legend is placed outside the fitted lines on the right side. Ignored if legend=FALSE.
rug	logical. If TRUE, the covariate distribution is displayed as a rug plot at the foot of the plot. Default to FALSE.
n.points	numeric. Number of values used to plot the fitted curves. Large values provide smoother curves.
edf.ylab	Should the edf value to be reported as y label? If NULL, edf.ylab is set to TRUE only if there is a single quantile curve to be plotted.
overall.eff	logical. If the smooth term has been called via ps(...,decompose=TRUE), by specifying overall.eff=TRUE the overall smooth effect is drawn.
grid	if provided, a grid of horizontal and vertical lines is drawn. grid has to be a list with the following components x,y,col,lty,lwd. If x (y) is a vector, the vertical (horizontal) lines are drawn at these locations. If x (y) is a scalar, the vertical (horizontal) lines are drawn at x (y) equispaced values. col,lty,lwd refer to the lines to be drawn.
smoos	logical, indicating if the residuals (provided that res=TRUE) will be drawn using a <i>smoothed</i> scatterplot. If NULL (default) the smoothed scatterplot will be employed when the number of observation is larger than 10000.
split	logical. If there are multiple smooth terms and split=TRUE, plot.gcrq() tries to split the plotting area in 2 columns and number of rows depending on the number of smooths. If split=FALSE, the plots are produced on the current device according to the current graphics settings. Ignored if there is single smooth term.

... Additional graphical parameters:
 xlab, ylab, ylim, and xlim (effective when add=FALSE);
 lwd, lty, and col for the fitted quantile lines; col<0 means color palette for the
 different curves;
 cex for the legend (if legend=TRUE);
 cex.p, col.p, and pch.p for the points (if res=TRUE).

Details

Takes a "gcrq" object and displays the fitted quantile curves. If `conf.level>0` pointwise confidence intervals are also displayed. When the object contains the component `cv`, `plot.gcrq` can display cross-validation scores against the lambda values, see argument `cv`.

Value

The function simply generates a new plot or adds fitted curves to an existing one.

Author(s)

Vito M. R. Muggeo

See Also

[gcrq](#), [predict.gcrq](#)

Examples

```
## Not run:
## use the fits from ?gcrq
## The additive model
plot(o, res=TRUE, col=2, conf.level=.9, shade=TRUE, split=TRUE)

par(mfrow=c(2,2))
plot(m5, select.tau=c(.1,.5,.9), overlap=0.6, legend=TRUE)
plot(m5, grid=list(x=8,y=5), lty=1) #a 8 times 5 grid.
plot(m7, cv=TRUE) #display CV score versus lambda values
plot(m7, res=TRUE, grid=list(x=5, y=8), col=4) #fitted curves at the best lambda value

## End(Not run)
```

predict.gcrq	<i>Prediction for "gcrq" objects</i>
--------------	--------------------------------------

Description

Takes a "gcrq" objects and computes fitted values

Usage

```
## S3 method for class 'gcrq'
predict(object, newdata, se.fit=FALSE, transf=NULL, xreg, ...)
```

Arguments

object	a fitted "gcrq" object.
newdata	a dataframe including <i>all</i> the covariates of the model. The smooth term is represented by a covariate and proper basis functions will be build accordingly. If omitted, the fitted values are used. Ignored if xreg is provided.
se.fit	logical. If TRUE, standard errors of the fitted quantiles are computed using the bootstrap covariance matrix.
transf	An optional character string (with "y" as argument) meaning a function to apply to the predicted values. E.g. " $\exp(y) - 0.1$ ". If NULL (default) it is taken as the inverse of function transf (*if*) supplied in gcrq. The standard errors (provided se.fit=TRUE has been set) are adjusted accordingly via the Delta method. See argument "transf" in gcrq(). If transf has been specified in gcrq(), use transf="y" to force predictions on the transformed scale, i.e. without back transforming.
xreg	the design matrix for which predictions are requested. Note xreg has to include the basis functions of the B-spline.
...	arguments passed to other functions

Details

predict.gcrq computes fitted quantiles as a function of observations included in newdata or xreg. Either newdata or xreg have to be supplied, but newdata is ignored when xreg is provided.

Value

If se.fit=FALSE, a matrix of fitted values with number of rows equal to number of rows of input data and number of columns depending on the number of fitted quantile curves (i.e length of taus).
If se=TRUE, a list of matrices (fitted values and standard errors).

Author(s)

Vito M.R. Muggeo

See Also

[gcrq](#), [plot.gcrq](#)

Examples

```
##see ?gcrq
## predict(m1, newdata=data.frame(x=c(.3,.7)))
```

print.gcrq

Print method for the gcrq class

Description

Printing the most important features of a gcrq model.

Usage

```
## S3 method for class 'gcrq'
print(x, digits = max(3, getOption("digits") - 4), ...)
```

Arguments

x	object of class gcrq
digits	number of digits to be printed
...	arguments passed to other functions

Author(s)

Vito M.R. Muggeo

See Also

[summary.gcrq](#)

ps *Specifying a smooth term in the gcrq formula.*

Description

Function used to define the smooth term (via P-splines) within the gcrq formula. The function actually does not evaluate a (spline) smooth, but simply it passes relevant information to proper fitter functions.

Usage

```
ps(..., lambda = -1, d = 3, by=NULL, ndx = NULL, deg = 3, knots=NULL,
    monotone = 0, concave = 0, var.pen = NULL, pen.matrix=NULL, dropc=TRUE,
    center=TRUE, K=2, ridge=FALSE, decompose=FALSE)
```

Arguments

...	The quantitative covariate supposed to have a nonlinear relationships with the quantiles. In growth charts this variable is typically the age.
lambda	A supplied smoothing parameter for the smooth term. If it is negative scalar, the smoothing parameter is estimated iteratively as discussed in Muggeo et al. (2020). If a positive scalar, it represents the actual smoothing parameter. If it is a vector, cross validation is performed to select the ‘best’ value. See Details in ?gcrq.
d	The difference order of the penalty. Default to 3.
by	if different from NULL, a numeric or factor variable of the same dimension as the covariate in ... In the numeric vector case the elements multiply the smooth, evaluated at the corresponding covariate values (i.e. a varying coefficient model).
ndx	The number of intervals of the covariate range used to build the B-spline basis. Non-integer values are rounded by round(). If NULL, default, the empirical rule of Ruppert is used, namely $\min(n/4, 40)$. It could be reduced (but no less than 10, say) if the sample size is not large, and the default value leads to some error in the fitting procedure, see section Note in gcrq.
deg	The degree of the spline polynomial. Default to 3.
knots	The knots locations. If NULL, equispaced knots are set.
monotone	Numeric value to set up monotonicity restrictions on the first derivative of fitted smooth function <ul style="list-style-type: none"> • ‘0’ = no constraint (default); • ‘1’ = non-decreasing smooth function; • ‘-1’ = non-increasing smooth function.
concave	Numeric value to set up monotonicity restrictions on the second derivative of fitted smooth function <ul style="list-style-type: none"> • ‘0’ = no constraint (default);

- '1' = concave smooth function;
- '-1' = convex smooth function.

var.pen	A character indicating the varying penalty. See Details.
pen.matrix	if provided, a penalty matrix A , say, such that $A^T A$ is the penalty matrix actually used in the estimation process. It overwrites ridge.
dropc	logical. Should the first column of the B-spline basis be dropped for the basis identifiability? Default to TRUE. Note, if dropc=FALSE is set, it is necessary to omit the model intercept AND center=FALSE or to set lambda.ridge>0 to fit the model.
center	logical. If TRUE the smooth effects are 'centered' over the covariate values, i.e. $\sum_i \hat{f}(x_i) = 0$.
K	A factor tuning selection of wiggleness of the smoothed curve. The larger K, the smoother the curve. Simulations suggest K=2. See details.
ridge	logical. If TRUE a ridge penalty is employed.
decompose	logical. If TRUE, the B-spline is decomposed into truncated power functions such as $[x, \dots, x^{d-1}, Z]$. See Details.

Details

When lambda=0 an unpenalized fit is obtained. At 'middle' lambda values, the fitted curve is a *piecewise* polynomial of order d-1. The fit gets smoother as lambda increases, and for a very large value of lambda, it approaches to a polynomial of degree d-1. It is also possible to put a varying penalty to set a different amount of smoothing. Namely for a constant smoothing (var.pen=NULL) the penalty is $\lambda \sum_k |\Delta_k^d|$ where Δ_k^d is the k-th difference (of order d) of the spline coefficients. For instance if $d = 1$, $|\Delta_k^1| = |b_k - b_{k-1}|$ where the b_k are the spline coefficients. When a varying penalty is set, the penalty becomes $\lambda \sum_k |\Delta_k^d| w_k$. The weights w_k depend on var.pen; for instance var.pen="(1:k)^2" results in $w_k = k^2$. See model m5 in examples of [gcrq](#).

Value

The function simply returns the covariate with added attributes relevant to smooth term.

Author(s)

Vito M. R. Muggeo

References

Muggeo VMR, Torretta F, Eilers PHC, Sciandra M, Attanasio M (2020). Multiple smoothing parameters selection in additive regression quantiles, *Statistical Modelling*, to appear.

For a general discussion on using B-spline and penalties in regression model see

Eilers PHC, Marx BD. (1996) Flexible smoothing with B-splines and penalties. *Statistical Sciences*, 11:89-121.

See Also

[gcrq](#)

Examples

```
##see ?gcrq

##gcrq(y ~ ps(x),..) #it works (default: center = TRUE, dropc = TRUE)
##gcrq(y ~ 0 + ps(x, center = TRUE, dropc = FALSE)) #it does NOT work
##gcrq(y ~ 0 + ps(x, center = FALSE, dropc = FALSE)) #it works
```

summary.gcrq

*Summarizing model fits for growth charts regression quantiles***Description**

summary and print methods for class gcrq

Usage

```
## S3 method for class 'gcrq'
summary(object, digits = max(3, getOption("digits") - 3),
        signif.stars =getOption("show.signif.stars"), ...)
```

Arguments

object	An object of class "gcrq".
digits	controls number of digits printed in output.
signif.stars	Should significance stars be printed?
...	further arguments.

Details

This method is at a preliminary stage. `summary.gcrq` returns some information on the fitted quantile curve at different probability values, such as the estimates, standard errors, values of check (objective) function values at solution. Currently there is no `print.summary.gcrq` method, so `summary.gcrq` itself prints results.

Author(s)

Vito M.R. Muggeo

See Also

[gcrq](#)

Examples

```
## see ?gcrq
##summary(o)
```

vcov.gcrq

Variance-Covariance Matrix for a Fitted 'cgrq' Model

Description

Returns the (currently only bootstrap-based) variance-covariance matrix of the parameter estimates of a fitted gcrq model object.

Usage

```
## S3 method for class 'gcrq'
vcov(object, term, type=c("boot", "sandw"), ...)
```

Arguments

object	a fitted model object of class "gcrq" returned by gcrq().
term	if specified the returned covariance matrix includes entries relevant to parameter estimates for that 'term' only. If missing, the returned matrices refer to all model parameter estimates.
type	Which cov matrix should be returned? Currently, only 'boot' is allowed.
...	additional arguments.

Details

If the "gcrq" object includes results from bootstrap runs (namely the component `boot.coef` is not `NULL`), `vcov.gcrq()` computes the covariance matrix for the parameter estimates of each quantile curve. If not, 100 bootstrap replicates are run.

Value

A list (with length equal to the length of `tau` specified in `gcrq`) of square matrices. Namely the list includes the covariance matrices of the parameter estimates for each regression quantile curve.

Author(s)

Vito Muggeo

See Also

[summary.gcrq](#)

Index

- * **datasets**
 - growthData, 8
 - * **models**
 - print.gcrq, 14
 - quantregGrowth-package, 2
 - * **model**
 - gcrq, 3
 - * **nonlinear**
 - ncross.rq.fitXB, 8
 - plot.gcrq, 10
 - predict.gcrq, 13
 - summary.gcrq, 17
 - * **package**
 - quantregGrowth-package, 2
 - * **regression**
 - gcrq, 3
 - ncross.rq.fitXB, 8
 - plot.gcrq, 10
 - predict.gcrq, 13
 - ps, 15
 - quantregGrowth-package, 2
 - summary.gcrq, 17
 - vcov.gcrq, 18
 - * **smooth**
 - gcrq, 3
 - ps, 15
- gcrq, 3, 3, 9, 10, 12, 14–17
- gcrq.rq.cv(ncross.rq.fitXB), 8
- growthData, 8
- ncross.rq.fitX(ncross.rq.fitXB), 8
- ncross.rq.fitXB, 8
- plot.gcrq, 7, 10, 14
- predict.gcrq, 7, 12, 13
- print.gcrq, 14
- ps, 5, 7, 9, 15
- quantregGrowth
(quantregGrowth-package), 2
- quantregGrowth-package, 2
- rq, 3
- summary.gcrq, 14, 17, 18
- vcov.gcrq, 18