

Package ‘quantregGrowth’

September 20, 2018

Type Package

Title Growth Charts via Regression Quantiles

Version 0.4-3

Date 2018-09-18

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. Monotonicity constraints on the fitted curves are allowed. See Muggeo, Scian-dra, Tomasello and Calvo (2013) <doi:10.1007/s10651-012-0232-1> and <doi:10.13140/RG.2.2.12924.85122> for some code example.

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 2018-09-20 16:30:03 UTC

R topics documented:

quantregGrowth-package	2
gcrq	3
growthData	7
ncross.rq.fitXB	8
plot.gcrq	9
predict.gcrq	11
print.gcrq	13
ps	13
summary.gcrq	15
vcov.gcrq	16
Index	17

quantregGrowth-package

Growth Charts via Regression Quantiles

Description

Fits non-crossing regression quantiles as a function of linear covariates and smooth terms via B-splines with difference penalties.

Details

Package: quantregGrowth
Type: Package
Version: 0.4-3
Date: 2018-09-18
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 constraints may be also set. `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, 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` *Growth charts regression quantiles*

Description

Estimation of 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 monotonicity restrictions are set to obtain estimates more biologically plausible. Linear terms are allowed in the model specification.

Usage

```
gcrq(formula, tau = c(0.1, 0.25, 0.5, 0.75, 0.9), data, subset, weights,
      na.action, transf=NULL, y = TRUE, interc=FALSE,
      foldid = NULL, n folds = 10, cv = TRUE, n.boot=0, eps=.0001, ...)
```

Arguments

<code>formula</code>	a standard R formula to specify the response in the left hand side, and the co-variates in the right hand side. See Details.
<code>tau</code>	a numeric vector to specify the percentiles of interest. Default to probability values (.1, .25, .5, .75, .9).
<code>data</code>	the dataframe where the variables required by the formula, subset and weights arguments are stored.
<code>subset</code>	optional. A vector specifying a subset of observations to be used in the fitting process.
<code>weights</code>	optional. A numeric vector specifying weights to be assigned to the observations in the fitting process. Currently unimplemented.
<code>na.action</code>	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.
interc	logical. If formula includes a "ps" term, interc=TRUE means that a model intercept is also estimated. If this is the case, a very small ridge-type penalty is exploited to allow estimation with a design matrix containing both a full B-spline basis and a column of ones. interc=TRUE overwrites the intercept specification in the formula (e.g., ~0+.), and it is ignored if the model does not include a "ps" term.
foldid	optional. A numeric vector identifying the group labels to perform cross validation to select the smoothing parameter. Ignored if the lambda argument in ps() is not a vector.
nfolds	optional. If foldid 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 lambda argument in ps() is not a vector.
cv	logical. If TRUE (default) and there is a single smooth term wherein lambda is a vector, the returned object includes also the matrix cv having number of rows equal to length of lambda and number of columns equal to nfolds. Ignored if the lambda argument in ps() is <i>not</i> a vector.
n.boot	Number of nonparametric (cases resampling) bootstrap samples to be used. Notice that the smoothing parameter (if relevant) 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. Use it at your risk! If eps is large, the resulting fitted quantile curves could appear unreasonable.
...	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 optionally include several `ps` terms to model nonlinear relationships with quantitative covariates, usually age in growth charts. When the lambda argument in `ps()` is scalar, it represents the actual smoothing parameter. 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 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 df 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 tau 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

The function (and underlying method) works pretty well in obtaining point estimates and displaying quantile curves accordingly. Typically this is the main (and unique) goal when dealing with growth charts. However from a statistical viewpoint there are some important limitations affecting the theory and the relevant package,

1. Computation of model degrees of freedom
2. Computation of standard errors, currently implemented via the nonparametric (cases resampling) bootstrap (if `n.boot>0`)

Currently the function does not return standard errors for the parameter estimates (unless `n.boot>0`) and degrees of freedom are roughly computed by summing the 'zero' residuals. Due to noncrossing constraints, the number of zero residuals might not be equal to the number of estimated parameters even in unpenalized models - except for the median, or for models where a single quantile curve has been fitted.

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`, and `plot.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.

Author(s)

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

References

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:
data(growthData) #load data
tauss<-seq(.1,.9,by=.1) #fix the percentiles of interest

m1<-gcrq(y~ps(x, mon=0), tau=tauss, data=growthData) #unpenalized.. very wiggly curves
#strongly penalized models
m2<-gcrq(y~ps(x, mon=0, lambda=1000, d=2), tau=tauss, data=growthData) #linear
m3<-gcrq(y~ps(x, mon=0, lambda=1000, d=3), tau=tauss, data=growthData) #quadratic

#penalized model with monotonicity restrictions
m4<-gcrq(y~ps(x, mon=1, lambda=10), tau=tauss, data=growthData)

#monotonicity constraints with varying penalty
m5<-gcrq(y~ps(x, mon=1, lambda=10, var.pen="(1:k)^3"), tau=tauss, data=growthData)

par(mfrow=c(2,2))
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)
```

```
#select lambda via 'K-fold' CV
m6<-gcrq(y~ps(x, lambda=seq(0.02,100,l=20)), tau=tauss, data=growthData)
par(mfrow=c(1,2))
plot(m6, cv=TRUE) #display CV score versus lambda values
plot(m6, res=TRUE, grid=list(x=5, y=8)) #fitted curves at the best lambda value

## End(Not run)
```

growthData

Simulated data to illustrate capabilities of the package

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, interc=FALSE,
  monotone = FALSE, adj.middle = FALSE, ndx = 10, lambda = 0,
  deg = 3, dif = 3, eps = 1e-04, plott = 0, var.pen = NULL, ...)
```

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

```
gcrq.rq.cv(y, B, X, taus, interc=FALSE, monotone, ndx, lambda, deg, dif,
  var.pen = NULL, cv = TRUE, nolds = 10, foldid = NULL, eps = 1e-04)
```

Arguments

<code>y</code>	the responses vector. see gcrq
<code>x</code>	the covariate supposed to have a nonlinear relationship.
<code>B</code>	the B-spline basis.
<code>X</code>	the design matrix for the linear parameters.
<code>taus</code>	the percentiles of interest.
<code>interc</code>	should the model intercept be estimated? see the same argument in gcrq
<code>monotone</code>	numerical value (-1/0/+1) to define a non-increasing, unconstrained, and non-decreasing flexible fit, respectively.
<code>adj.middle</code>	ignore it!
<code>ndx</code>	number of internal intervals within the covariate range, see ps .
<code>lambda</code>	smoothing parameter, see ps .
<code>deg</code>	spline degree, see ps .
<code>dif</code>	difference order of the spline coefficients in the penalty term.
<code>eps</code>	tolerance value.
<code>plott</code>	Plotting results, unuseful. See plot.gcrq .
<code>var.pen</code>	Varying penalty, 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	<i>Plot method for gcrq objects</i>
-----------	-------------------------------------

Description

Displaying the estimated growth charts from a `gcrq` fit.

Usage

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

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 will be plotted. It may be missing if the model includes a single smooth term.
<code>add</code>	logical. If TRUE the fitted quantile curves are added on the current plot.
<code>res</code>	logical. If TRUE '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. 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>).
<code>legend</code>	logical. If TRUE a legend is drawn on the right side of the plot.
<code>select.tau</code>	an optional numeric vector to draw only some of the fitted quantiles. Percentile values or integers 1 to <code>length(tau)</code> may be supplied.
<code>deriv</code>	logical. If TRUE the first derivative of the curve is displayed.
<code>cv</code>	logical. If TRUE and the "gcrq" object contains the matrix <code>cv</code> , then the cross-validation scores against the lambda values are plotted.
<code>transf</code>	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 <code>transf</code> (*if*) supplied in <code>gcrq</code> . See argument "transf" in <code>gcrq()</code> . If <code>transf</code> has been specified in <code>gcrq()</code> , use <code>transf="y"</code> to force plotting on the transformed scale, i.e. without back transforming.
<code>lambda0</code>	logical. If <code>cv=TRUE</code> , should the CV plot include also the first CV value? Usually the first CV value is at <code>lambda=0</code> , 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.
<code>shade</code>	logical. If TRUE and <code>conf.level>0</code> , the pointwise confidence intervals are portrayed via shaded areas.
<code>overlap</code>	logical. If TRUE, and <code>legend=TRUE</code> , the legend crosses the fitted curve on the right side. Otherwise the legend is placed outside the fitted lines.
<code>rug</code>	logical. If TRUE, the covariate distribution is displayed as a rug plot at the foot of the plot. Default to FALSE.
<code>n.points</code>	numeric. Number of values used to plot the fitted curves. Large values provide smoother curves.
<code>grid</code>	if provided, a grid of horizontal and vertical lines is drawn. <code>grid</code> has to be a list with the following components <code>x</code> , <code>y</code> , <code>col</code> , <code>lty</code> , <code>lwd</code> . If <code>x</code> (<code>y</code>) is a vector, the vertical (horizontal) lines are drawn at these locations. If <code>x</code> (<code>y</code>) is a scalar, the vertical (horizontal) lines are drawn at <code>x</code> (<code>y</code>) equispaced values. <code>col</code> , <code>lty</code> , <code>lwd</code> refer to the lines to be drawn.

smoos logical, indicating if the residuals (provided that `res=TRUE`) will be drawn using a *smoothed* scatterplot. If `NULL` (default) the smoothed scatterplot will be employed when the number of observation is larger than 10000.

... 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 with contribution by Gianluca Sottile (gianluca.sottile@unipa.it)

See Also

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

Examples

```
## see ?gcrq
##plot(m5, conf.level=.8, lty=1)
##plot(m5, lty=1:3, select.tau=c(.1,.5,.9))
##plot(m5, grid=list(x=8,y=5)) #a 8 times 5 grid..
```

predict.gcrq

Prediction for "gcrq" objects

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).

Note

This function is at a preliminary stage and it should be replaced by the method predict.gcrq. Please use it with care. This function is not expect to work if the fitted model includes factors or terms like poly().

Author(s)

Vito M.R. Muggeo

See Also

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

Examples

```
##see ?gcrq
```

print.gcrq	<i>Print method for the gcrq class</i>
------------	--

Description

Printing the most important features of a gcrq model.

Usage

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

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	<i>Specifying a smooth term in the gcrq formula.</i>
----	--

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(x, monotone = 0, lambda = 0, d = 3, ndx = NULL, deg = 3,  
var.pen = NULL)
```

Arguments

x	The quantitative covariate supposed to have a nonlinear relationships with the quantiles. In growth charts this variable is typically the age.
monotone	Numeric value to set up monotonicity restrictions on the fitted smooth function <ul style="list-style-type: none"> • '0' = no constrain; • '1' = non decreasing smooth function; • '-1' = non increasing smooth function.
lambda	A supplied smoothing parameter for the smooth term. If it is a vector, cross validation is performed to select the 'best' value.
d	The difference order of the penalty. Default to 3.
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.
var.pen	A character indicating the varying penalty. See Details.

Details

When $\lambda=0$ an unpenalized fit is obtained. At 'middle' λ values, the fitted curve is a *piecewise* polynomial of order $d-1$. The fit gets smoother as λ increases, and for a very large value of λ , 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 ($\text{var.pen}=\text{NULL}$) 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 $\text{var.pen}="(1:k)^2"$ 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

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
```

```
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), ...)
```

Arguments

object	An object of class "gcrq".
digits	controls number of digits printed in output.
...	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
```

`vcov.gcrq`*Variance-Covariance Matrix for a Fitted 'gcrq' 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, ...)
```

Arguments

<code>object</code>	a fitted model object of class "gcrq" returned by <code>gcrq()</code> .
<code>term</code>	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.
<code>...</code>	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 including the covariance matrices of the parameter estimates for each regression quantile curve.

Author(s)

Vito Muggeo

See Also

[summary.gcrq](#)

Index

- *Topic **datasets**
 - growthData, 7
 - *Topic **models**
 - print.gcrq, 13
 - quantregGrowth-package, 2
 - *Topic **model**
 - gcrq, 3
 - *Topic **nonlinear**
 - ncross.rq.fitXB, 8
 - plot.gcrq, 9
 - predict.gcrq, 11
 - summary.gcrq, 15
 - *Topic **package**
 - quantregGrowth-package, 2
 - *Topic **regression**
 - gcrq, 3
 - ncross.rq.fitXB, 8
 - plot.gcrq, 9
 - predict.gcrq, 11
 - ps, 13
 - quantregGrowth-package, 2
 - summary.gcrq, 15
 - vcov.gcrq, 16
 - *Topic **smooth**
 - gcrq, 3
 - ps, 13
- gcrq, 3, 3, 8, 9, 11, 12, 14, 15
- gcrq.rq.cv(ncross.rq.fitXB), 8
- growthData, 7
- ncross.rq.fitX(ncross.rq.fitXB), 8
- ncross.rq.fitXB, 8
- plot.gcrq, 6, 8, 9, 12
- predict.gcrq, 6, 11, 11
- print.gcrq, 13
- ps, 4, 6, 8, 13
- quantregGrowth
(quantregGrowth-package), 2
- quantregGrowth-package, 2
- rq, 3
- summary.gcrq, 13, 15, 16
- vcov.gcrq, 16