Package ‘expectreg’

October 13, 2022

Version 0.52

Date 2022-03-15

Title Expectile and Quantile Regression

Author Fabian Otto-Sobotka [cre],
Elmar Spiegel [aut],
Sabine Schnabel [aut],
Linda Schulze Waltrup [aut],
Paul Eilers [ctb],
Thomas Kneib [ths],
Goeran Kauermann [ctb]

Maintainer Fabian Otto-Sobotka <fabian.otto-sobotka@uni-oldenburg.de>

Depends R (>= 3.5.0), stats, parallel, mboost (>= 2.1.0), BayesX (>= 0.2-4), Matrix

Imports Rcpp (>= 0.11.2), splines, quadprog, colorspace (>= 0.97), fields

Suggests SemiPar

LinkingTo Rcpp, RcppEigen

Description Expectile and quantile regression of models with nonlinear effects e.g. spatial, random, ridge using least asymmetric weighed squares / absolutes as well as boosting; also supplies expectiles for common distributions.

License GPL-2

LazyData yes

NeedsCompilation yes

Encoding UTF-8

Repository CRAN

Date/Publication 2022-03-15 16:00:02 UTC

R topics documented:

`expectreg-package` *Expectile and Quantile Regression*

Description

Expectile and quantile regression of models with nonlinear effects e.g. spatial, random, ridge using least asymmetric weighed squares / absolutes as well as boosting; also supplies expectiles for common distributions.

Details

Package:	expectreg
Type:	Package
Version:	0.55
Date:	2022-03-15
License:	GPL-2
LazyLoad:	yes
LazyData:	yes

- This package requires the packages `BayesX`, `mboost`, `splines` and `quadprog`.

Author(s)

Fabian Otto-Sobotka
Carl von Ossietzky University Oldenburg
`https://uol.de`

Elmar Spiegel
Helmholtz Centre Munich
https://www.helmholtz-munich.de

Sabine Schnabel
Wageningen University and Research Centre
https://www.wur.nl

Linda Schulze Waltrup
Ludwig Maximilian University Munich
https://www.lmu.de

with contributions from

Paul Eilers
Erasmus Medical Center Rotterdam
https://www.erasmusmc.nl

Thomas Kneib
Georg August University Goettingen
https://www.uni-goettingen.de

Goeran Kauermann
Ludwig Maximilian University Munich
https://www.lmu.de

Maintainer: Fabian Otto-Sobotka <fabian.otto-sobotka@uni-oldenburg.de>

References

Fenske N and Kneib T and Hothorn T (2009) *Identifying Risk Factors for Severe Childhood Malnutrition by Boosting Additive Quantile Regression* Technical Report 052, University of Munich

He X (1997) *Quantile Curves without Crossing* The American Statistician, 51(2):186-192

Koenker R (2005) *Quantile Regression* Cambridge University Press, New York

Schnabel S and Eilers P (2009) *Optimal expectile smoothing* Computational Statistics and Data Analysis, 53:4168-4177

Schnabel S and Eilers P (2011) *Expectile sheets for joint estimation of expectile curves* (under review at Statistical Modelling)

Sobotka F and Kneib T (2010) *Geoadditive Expectile Regression* Computational Statistics and Data Analysis, doi: 10.1016/j.csda.2010.11.015.

See Also

mboost, BayesX

Examples

```
data(dutchboys)

## Expectile Regression using the restricted approach
ex = expectreg.ls(dist ~ rb(speed),data=cars,smooth="f",lambda=5,estimate="restricted")
names(ex)

## The calculation of expectiles for given distributions
enorm(0.1)
enorm(0.5)

## Introducing the expectiles-meet-quantiles distribution
x = seq(-5,5,length=100)
plot(x,demq(x),type="l")

## giving an expectile analogon to the 'quantile' function
y = rnorm(1000)

expectile(y)

eenorm(y)
```

---

cdf.qp *Calculation of the conditional CDF based on expectile curves*

Description

Estimating the CDF of the response for a given value of covariate. Additionally quantiles are computed from the distribution function which allows for the calculation of regression quantiles.

Usage

```
cdf.qp(expectreg, x = NA, qout = NA, extrap = FALSE, e0 = NA, eR = NA,
       lambda = 0, var.dat = NA)

cdf.bundle(bundle, qout = NA, extrap = FALSE, quietly = FALSE)
```

Arguments

`expectreg, bundle`
: An object of class expectreg or subclass bundle respectively. The number of expectiles should be high enough to ensure accurate estimation. One approach would be to take as many expectiles as data points. Also make sure that extreme expectiles are incuded, e.g. expectiles corresponding to very small and large asymmetrie values.

`x`
: The covariate value where the CDF is estimated. By default the first covariate value.

`qout`
: Vector of quantiles that will be computed from the CDF.

extrap	If TRUE, extreme quantiles will be extrapolated linearly, otherwise the maximum of the CDF is used.
e0	Scalar number which offers the possibility to specify an artificial minimal expectile (for example the minimum of the data) used for the calculation. By default e0 = e1 + (e1 - e2) where e1 is the actual minimal expectile and e2 the second smallest expectile.
eR	Scalar number which offers the possibility to specify an artificial maximal expectile (for example the maximum of the data) used for the calculation. By default eR = eR-1 + (eR-1 - eR-2) where eR-1 is the actual maximal expectile and eR-2 the second largest expectile.
lambda	Positive Scalar. Penalty parameter steering the smoothness of the fitted CDF. By default equal to 0 which means no penalization.
var.dat	Positive Scalar. If a penalization is applied (i.e. lambda unequal to 0), this argument can be used to let the penalty depend on the variance of the expectiles (which is the default).
quietly	If programm should run quietly.

Details

Expectile curves can describe very well the spread and location of a scatterplot. With a set of curves they give good impression about the nature of the data. This information can be used to estimate the conditional density from the expectile curves. The results of the bundle model are especially suited in this case as only one density will be estimated which can then be modulated to over the independent variable x. The density estimation can be formulated as penalized least squares problem that results in a smooth non-negative density. The theoretical values of a quantile regression at this covariate value are also returned for adjustable probabilities qout.

Value

A list consisting of

x	vector of expectiles where the CDF is computed.
cdf	vector of values of the CDF at the expectiles x.
quantiles	vector of quantile values estimated from the CDF.
qout	vector of probabilities for the calculated quantiles.

Author(s)

Goeran Kauermann, Linda Schulze Waltrup
Ludwig Maximilian University Munich
https://www.lmu.de

Fabian Sobotka
Georg August University Goettingen
https://www.uni-goettingen.de

Sabine Schnabel
Wageningen University and Research Centre
https://www.wur.nl

Paul Eilers
Erasmus Medical Center Rotterdam
https://www.erasmusmc.nl

References

Schnabel SK and Eilers PHC (2010) *A location scale model for non-crossing expectile curves* (working paper)

Schulze Waltrup L, Sobotka F, Kneib T and Kauermann G (2014) *Expectile and Quantile Regression - David and Goliath?* Statistical Modelling.

See Also

expectreg.ls, expectreg.qp

Examples

```
d = expectreg.ls(dist ~ rb(speed),data=cars,smooth="f",lambda=5,estimate="restricted",
                 expectiles=c(0.0001,0.001,seq(0.01,0.99,0.01),0.999,0.9999))
e = cdf.qp(d,15,extrap=TRUE)
e
```

`dutchboys` *Data set about the growth of dutch children*

Description

Data from the fourth dutch growth study in 1997.

Usage

```
data(dutchboys)
```

Format

A data frame with 6848 observations on the following 10 variables.

`defnr` identification number

`age` age in decimal years

`hgt` length/height in cm

`wgt` weight in kg

`hc` head circumference in cm

`hgt.z` z-score length/height

`wgt.z` z-score weight

`hc.z` z-score head circumference

`bmi.z` z-score body mass index

`hfw.z` z-score height for weight

z-scores were calculated relative to the Dutch references.

Details

The Fourth Dutch Growth Study is a cross-sectional study that measures growth and development of the Dutch population between ages 0 and 21 years. The study is a follow-up to earlier studies performed in 1955, 1965 and 1980, and its primary goal is to update the 1980 references.

Source

van Buuren S and Fredriks A (2001) *Worm plot: A simple diagnostic device for modeling growth reference curves* Statistics in Medicine, 20:1259-1277

References

Schnabel S and Eilers P (2009) *Optimal expectile smoothing* Computatational Statistics and Data Analysis, 53: 4168-4177

Examples

```
data(dutchboys)

expreg <- expectreg.ls(dutchboys[,3] ~ rb(dutchboys[,2],"pspline"),smooth="f",
                       estimate="restricted",expectiles=c(.05,.5,.95))
plot(expreg)
```

`enorm` *Expectiles of distributions*

Description

Much like the 0.5 quantile of a distribution is the median, the 0.5 expectile is the mean / expected value. These functions add the possibility of calculating expectiles of known distributions. The functions starting with 'e' calculate an expectile value for given asymmetry values, the functions starting with 'pe' calculate vice versa.

Usage

```
enorm(asy, m = 0, sd = 1)
penorm(e, m = 0, sd = 1)

ebeta(asy, a = 1, b = 1)
pebeta(e, a = 1, b = 1)

eunif(asy, min = 0, max = 1)
peunif(e, min = 0, max = 1)

et(asy, df)
pet(e, df)

elnorm(asy, meanlog = 0, sdlog = 1)
pelnorm(e, meanlog = 0, sdlog = 1)

egamma(asy, shape, rate = 1, scale = 1/rate)
pegamma(e, shape, rate = 1, scale = 1/rate)

eexp(asy, rate = 1)
peexp(e, rate = 1)

echisq(asy, df)
pechisq(e, df)
```

Arguments

`asy`	vector of asymmetries with values between 0 and 1.
`e`	vector of expectiles from the respective distribution.
`m, sd`	mean and standard deviation of the Normal distribution.
`a, b`	positive parameters of the Beta distribution.
`min, max`	minimum, maximum of the uniform distribution.
`df`	degrees of freedom of the student t and chi squared distribution.
`meanlog, sdlog`	parameters of the lognormal distribution.
`shape, rate, scale`	parameters of the gamma distribution (with 2 different parametrizations) and parameter of the exponential distribution which is a special case of the gamma with shape=1.

Details

An expectile of a distribution cannot be determined explicitely, but instead is given by an equation. The expectile z for an asymmetry p is: $p = \frac{G(z)-zF(z)}{2(G(z)-zF(z))+z-m}$ where m is the mean, F the cdf and G the partial moment function $G(z) = \int\limits_{-\infty}^{z} uf(u)\mathrm{d}u$.

Value

Vector of the expectiles or asymmetry values for the desired distribution.

Author(s)

Fabian Otto- Sobotka
Carl von Ossietzky University Oldenburg
https://uol.de

Thomas Kneib
Georg August University Goettingen
https://www.uni-goettingen.de

References

Newey W and Powell J (1987) *Asymmetric least squares estimation and testing* Econometrica, 55:819-847

See Also

eemq

Examples

```
x <- seq(0.02,0.98,0.2)

e = enorm(x)
e
penorm(e)
```

---

expectile — *Sample Expectiles*

Description

Expectiles are fitted to univariate samples with least asymmetrically weighted squares for asymmetries between 0 and 1. For graphical representation an expectile - expectile plot is available. The corresponding functions quantile, qqplot and qqnorm are mapped here for expectiles.

Usage

```
expectile(x, probs = seq(0, 1, 0.25), dec = 4)

eenorm(y, main = "Normal E-E Plot",
       xlab = "Theoretical Expectiles", ylab = "Sample Expectiles",
```

```
        plot.it = TRUE, datax = FALSE, ...)

eeplot(x, y, plot.it = TRUE, xlab = deparse(substitute(x)),
       ylab = deparse(substitute(y)), main = "E-E Plot", ...)
```

Arguments

`x, y`	Numeric vector of univariate observations.
`probs`	Numeric vector of asymmetries between 0 and 1 where 0.5 corresponds to the mean.
`dec`	Number of decimals remaining after rounding the results.
`plot.it`	logical. Should the result be plotted?
`datax`	logical. Should data values be on the x-axis?
`xlab, ylab, main`	plot labels. The xlab and ylab refer to the x and y axes respectively if `datax` = TRUE.
`...`	graphical parameters.

Details

In least asymmetrically weighted squares (LAWS) each expectile is fitted independently from the others. LAWS minimizes:

$S = \sum_{i=1}^{n} w_i(p)(x_i - \mu(p))^2$

with

$w_i(p) = p1_{(x_i > \mu(p))} + (1 - p)1_{(x_i < \mu(p))}$.

$\mu(p)$ is determined by iteration process with recomputed weights $w_i(p)$.

Value

Numeric vector with the fitted expectiles.

Author(s)

Fabian Otto-Sobotka
Carl von Ossietzky University Oldenburg
https://uol.de

References

Sobotka F and Kneib T (2010) *Geoadditive Expectile Regression* Computational Statistics and Data Analysis, doi: 10.1016/j.csda.2010.11.015.

See Also

`expectreg.ls`, `quantile`

Examples

```
data(dutchboys)

expectile(dutchboys[,3])

x = rnorm(1000)

expectile(x,probs=c(0.01,0.02,0.05,0.1,0.2,0.5,0.8,0.9,0.95,0.98,0.99))

eenorm(x)
```

`expectreg.boost` *Quantile and expectile regression using boosting*

Description

Generalized additive models are fitted with gradient boosting for optimizing arbitrary loss functions to obtain the graphs of 11 different expectiles for continuous, spatial or random effects.

Usage

```
expectreg.boost(formula, data, mstop = NA, expectiles = NA, cv = TRUE,
BoostmaxCores = 1, quietly = FALSE)

quant.boost(formula, data, mstop = NA, quantiles = NA, cv = TRUE,
BoostmaxCores = 1, quietly = FALSE)
```

Arguments

`formula`	An R formula object consisting of the response variable, '~' and the sum of all effects that should be taken into consideration (see `gamboost`). Each effect can be linear or represented through a nonlinear or spatial base (see `bbs`). Each variable has to be named consistently with `data`.
`data`	data frame (is required).
`mstop`	vector, number of bootstrap iterations for each of the 11 quantiles/expectiles that are fitted. Default is 4000.
`expectiles, quantiles`	In default setting, the expectiles (0.01,0.02,0.05,0.1,0.2,0.5,0.8,0.9,0.95,0.98,0.99) are calculated. You may specify your own set of expectiles in a vector.
`cv`	A cross-validation can determine the optimal amount of boosting iterations between 1 and `mstop`. Uses `cvrisk`. If set to `FALSE`, the results from `mstop` iterations are used.
`BoostmaxCores`	Maximum number of used cores for the different asymmetry parameters
`quietly`	If programm should run quietly.

Details

A (generalized) additive model is fitted using a boosting algorithm based on component-wise univariate base learners. The base learner can be specified via the formula object. After fitting the model a cross-validation is done using `cvrisk` to determine the optimal stopping point for the boosting which results in the best fit.

Value

An object of class 'expectreg', which is basically a list consisting of:

`values`	The fitted values for each observation and all expectiles, separately in a list for each effect in the model, sorted in order of ascending covariate values.
`response`	Vector of the response variable.
`formula`	The formula object that was given to the function.
`asymmetries`	Vector of fitted expectile asymmetries as given by argument `expectiles`.
`effects`	List of characters giving the types of covariates.
`helper`	List of additional parameters like neighbourhood structure for spatial effects or 'phi' for kriging.
`fitted`	Fitted values $\hat{y}$.

`plot`, `predict`, `resid`, `fitted` and `effects` methods are available for class 'expectreg'.

Author(s)

Fabian Otto- Sobotka
Carl von Ossietzky University Oldenburg
https://uol.de

Thomas Kneib, Elmar Spiegel
Georg August University Goettingen
https://www.uni-goettingen.de

References

Fenske N and Kneib T and Hothorn T (2009) *Identifying Risk Factors for Severe Childhood Malnutrition by Boosting Additive Quantile Regression* Technical Report 052, University of Munich

Sobotka F and Kneib T (2010) *Geoadditive Expectile Regression* Computational Statistics and Data Analysis, doi: 10.1016/j.csda.2010.11.015.

See Also

`expectreg.ls`, `gamboost`, `bbs`, `cvrisk`

Examples

```
data("lidar", package = "SemiPar")
ex <- expectreg.boost(logratio ~ bbs(range),lidar, mstop=200,
                      expectiles=c(0.1,0.5,0.95),quietly=TRUE)
plot(ex)
```

---

expectreg.ls — *Expectile regression of additive models*

Description

Additive models are fitted with least asymmetrically weighted squares or quadratic programming to obtain expectiles for parametric, continuous, spatial and random effects.

Usage

```
expectreg.ls(formula, data = NULL, estimate = c("laws", "restricted", "bundle", "sheets"),
smooth = c("schall", "ocv", "gcv", "cvgrid", "aic", "bic", "lcurve", "fixed"),
lambda = 1, expectiles = NA, ci = FALSE, LAWSmaxCores = 1, ...)

expectreg.qp(formula, data  = NULL, id = NA, smooth = c("schall", "acv", "fixed"),
             lambda = 1, expectiles = NA)
```

Arguments

`formula`	An R formula object consisting of the response variable, '~' and the sum of all effects that should be taken into consideration. Each effect has to be given through the function `rb`.
`data`	Optional data frame containing the variables used in the model, if the data is not explicitely given in the formula.
`id`	Potential additional variable identifying individuals in a longitudinal data set. Allows for a random intercept estimation.
`estimate`	Character string defining the estimation method that is used to fit the expectiles. Further detail on all available methods is given below.
`smooth`	There are different smoothing algorithms that should prevent overfitting. The 'schall' algorithm iterates the smoothing penalty `lambda` until it converges (REML). The generalised cross-validation 'gcv',similar to the ordinary cross- validation 'ocv' minimizes a score-function using `nlminb` or with a grid search by 'cvgrid' or the function uses a fixed penalty. The numerical minimisatioin is also possible with AIC or BIC as score. The L-curve is a new experimental grid search by Frasso and Eilers.
`lambda`	The fixed penalty can be adjusted. Also serves as starting value for the smoothing algorithms.

`expectiles`	In default setting, the expectiles (0.01,0.02,0.05,0.1,0.2,0.5,0.8,0.9,0.95,0.98,0.99) are calculated. You may specify your own set of expectiles in a vector. The option may be set to 'density' for the calculation of a dense set of expectiles that enhances the use of `cdf.qp` and `cdf.bundle` afterwards.
`ci`	Whether a covariance matrix for confidence intervals and a `summary` is calculated.
`LAWSmaxCores`	How many cores should maximal be used by parallelization
`...`	Optional value for re-weight the model with estimate weights and combine selected models to one model.

Details

In least asymmetrically weighted squares (LAWS) each expectile is fitted independently from the others. LAWS minimizes:

$S = \sum_{i=1}^{n} w_i(p)(y_i - \mu_i(p))^2$

with

$w_i(p) = p1_{(y_i > \mu_i(p))} + (1-p)1_{(y_i < \mu_i(p))}$.

The restricted version fits the 0.5 expectile at first and then the residuals. Afterwards the other expectiles are fitted as deviation by a factor of the residuals from the mean expectile. This algorithm is based on He(1997). The advantage is that expectile crossing cannot occur, the disadvantage is a suboptimal fit in certain heteroscedastic settings. Also, since the number of fits is significantly decreased, the restricted version is much faster.

The expectile bundle has a resemblence to the restricted regression. At first, a trend curve is fitted and then an iteration is performed between fitting the residuals and calculating the deviation factors for all the expectiles until the results are stable. Therefore this function shares the (dis)advantages of the restricted.

The expectile sheets construct a p-spline basis for the expectiles and perform a continuous fit over all expectiles by fitting the tensor product of the expectile spline basis and the basis of the covariates. In consequence there will be most likely no crossing of expectiles but also a good fit in heteroscedastic scenarios.

The function `expectreg.qp` also fits a sheet over all expectiles, but it uses quadratic programming with constraints, so crossing of expectiles will definitely not happen. So far the function is implemented for one nonlinear or spatial covariate and further parametric covariates. It works with all smoothing methods.

Value

An object of class 'expectreg', which is basically a list consisting of:

`lambda`	The final smoothing parameters for all expectiles and for all effects in a list. For the restricted and the bundle regression there are only the mean and the residual lambda.
`intercepts`	The intercept for each expectile.
`coefficients`	A matrix of all the coefficients, for each base element a row and for each expectile a column.

`values`	The fitted values for each observation and all expectiles, separately in a list for each effect in the model, sorted in order of ascending covariate values.
`response`	Vector of the response variable.
`covariates`	List with the values of the covariates.
`formula`	The formula object that was given to the function.
`asymmetries`	Vector of fitted expectile asymmetries as given by argument `expectiles`.
`effects`	List of characters giving the types of covariates.
`helper`	List of additional parameters like neighbourhood structure for spatial effects or 'phi' for kriging.
`design`	Complete design matrix.
`bases`	Bases components of each covariate.
`fitted`	Fitted values $\hat{y}$.
`covmat`	Covariance matrix, estimated when `ci = TRUE`.
`diag.hatma`	Diagonal of the hat matrix. Used for model selection criteria.
`data`	Original data
`smooth_orig`	Unchanged original type of smoothing.

`plot`, `predict`, `resid`, `fitted`, `effects` and further convenient methods are available for class 'expectreg'.

Author(s)

Fabian Otto-Sobotka
Carl von Ossietzky University Oldenburg
https://uol.de

Thomas Kneib
Georg August University Goettingen
https://www.uni-goettingen.de

Sabine Schnabel
Wageningen University and Research Centre
https://www.wur.nl

Paul Eilers
Erasmus Medical Center Rotterdam
https://www.erasmusmc.nl

Linda Schulze Waltrup, Goeran Kauermann
Ludwig Maximilians University Muenchen
https://www.lmu.de

References

Schnabel S and Eilers P (2009) *Optimal expectile smoothing* Computational Statistics and Data Analysis, 53:4168-4177

Sobotka F and Kneib T (2010) *Geoadditive Expectile Regression* Computational Statistics and Data Analysis, doi: 10.1016/j.csda.2010.11.015.

Schnabel S and Eilers P (2011) *Expectile sheets for joint estimation of expectile curves* (under review at Statistical Modelling)

Frasso G and Eilers P (2013) *Smoothing parameter selection using the L-curve* (under review)

See Also

rb, expectreg.boost

Examples

```
library(expectreg)
ex = expectreg.ls(dist ~ rb(speed),data=cars,smooth="b",lambda=5,expectiles=c(0.01,0.2,0.8,0.99))
ex = expectreg.ls(dist ~ rb(speed),data=cars,smooth="f",lambda=5,estimate="restricted")
plot(ex)


data("lidar", package = "SemiPar")

explaws <- expectreg.ls(logratio~rb(range,"pspline"),data=lidar,smooth="gcv",
                        expectiles=c(0.05,0.5,0.95))
print(explaws)
plot(explaws)

###expectile regression using a fixed penalty
plot(expectreg.ls(logratio~rb(range,"pspline"),data=lidar,smooth="fixed",
     lambda=1,expectiles=c(0.05,0.25,0.75,0.95)))
plot(expectreg.ls(logratio~rb(range,"pspline"),data=lidar,smooth="fixed",
     lambda=0.0000001,expectiles=c(0.05,0.25,0.75,0.95)))
    #As can be seen in the plot, a too small penalty causes overfitting of the data.
plot(expectreg.ls(logratio~rb(range,"pspline"),data=lidar,smooth="fixed",
     lambda=50,expectiles=c(0.05,0.25,0.75,0.95)))
    #If the penalty parameter is chosen too large,
    #the expectile curves are smooth but don't represent the data anymore.
```

Gasoline *Gasoline Consumption*

Description

A panel of 18 observations from 1960 to 1978 in OECD countries.

Usage

```
data("Gasoline")
```

Format

A data frame with 342 observations on the following 6 variables.

`country` a factor with 18 levels `AUSTRIA` `BELGIUM` `CANADA` `DENMARK` `FRANCE` `GERMANY` `GREECE` `IRELAND` `ITALY` `JAPAN` `NETHERLA` `NORWAY` `SPAIN` `SWEDEN` `SWITZERL` `TURKEY` `U.K.` `U.S.A.`

`year` the year

`lgaspcar` logarithm of motor gasoline consumption per car

`lincomep` logarithm of real per-capita income

`lrpmg` logarithm of real motor gasoline price

`lcarpcap` logarithm of the stock of cars per capita

Source

Online complements to Baltagi (2001).

https://www.wiley.com/legacy/wileychi/baltagi/

References

Baltagi, Badi H. (2001) "Econometric Analysis of Panel Data", 2nd ed., John Wiley and Sons.

Gibraltar, B.H. and J.M. Griffin (1983) ???Gasoline demand in the OECD: An application of pooling and testing procedures???, European Economic Review, 22(2), 117???137.

Examples

```
data(Gasoline)

expreg<-expectreg.ls(lrpmg~rb(lcarpcap),smooth="fixed",data=Gasoline,
lambda=20,estimate="restricted",expectiles=c(0.01,0.05,0.2,0.8,0.95,0.99))

plot(expreg)
```

`india` *Malnutrition of Childen in India*

Description

Data sample from a 'Demographic and Health Survey' about malnutrition of children in india. Data set only contains 1/10 of the observations and some basic variables to enable first analyses.

Usage

```
data(india)
```

Format

A data frame with 4000 observations on the following 6 variables.

`stunting` A numeric malnutrition score with range (-600;600).

`cbmi` BMI of the child.

`cage` Age of the child in months.

`mbmi` BMI of the mother.

`mage` Age of the mother in years.

`distH` The distict in India, where the child lives. Encoded in the region naming of the map `india.bnd`.

Source

http://www.measuredhs.com

References

Fenske N and Kneib T and Hothorn T (2009) *Identifying Risk Factors for Severe Childhood Malnutrition by Boosting Additive Quantile Regression* Technical Report 052, University of Munich

Examples

```
data(india)

expreg <- expectreg.ls(stunting ~ rb(cbmi),smooth="fixed",data=india,
lambda=30,estimate="restricted",expectiles=c(0.01,0.05,0.2,0.8,0.95,0.99))
plot(expreg)
```

`india.bnd` *Regions of India - boundary format*

Description

Map of the country india, represented in the boundary format (bnd) as defined in the package `BayesX`.

Usage

```
data(india.bnd)
```

Format

The format is: List of 449 - attr(*, "class")= chr "bnd" - attr(*, "height2width")= num 0.96 - attr(*, "surrounding")=List of 449 - attr(*, "regions")= chr [1:440] "84" "108" "136" "277" ...

Details

For details about the format see read.bnd.

Source

Jan Priebe University of Goettingen https://www.giga-hamburg.de/de/team/11564856-priebe-jan/

Examples

```
data(india)
data(india.bnd)

drawmap(data=india,map=india.bnd,regionvar=6,plotvar=1)
```

---

methods *Methods for expectile regression objects*

Description

Methods for objects returned by expectile regression functions.

Usage

```
## S3 method for class 'expectreg'
print(x, ...)

## S3 method for class 'expectreg'
summary(object,...)

## S3 method for class 'expectreg'
predict(object, newdata = NULL, with_intercept = T, ...)

## S3 method for class 'expectreg'
x[i]

## S3 method for class 'expectreg'
residuals(object, ...)
## S3 method for class 'expectreg'
resid(object, ...)

## S3 method for class 'expectreg'
fitted(object, ...)
## S3 method for class 'expectreg'
fitted.values(object, ...)
```

```
## S3 method for class 'expectreg'
effects(object, ...)

## S3 method for class 'expectreg'
coef(object, ...)
## S3 method for class 'expectreg'
coefficients(object, ...)

## S3 method for class 'expectreg'
confint(object, parm = NULL, level = 0.95, ...)
```

Arguments

x,object	An object of class expectreg as returned e.g. by the function expectreg.ls.
newdata	Optionally, a data frame in which to look for variables with which to predict.
with_intercept	Should the intercept be added to the prediction of splines?
i	Covariate numbers to be kept in subset.
level	Coverage probability of the generated confidence intervals.
parm	Optionally the confidence intervals may be restricted to certain covariates, to be named in a vector. Otherwise the confidence intervals for the fit are returned.
...	additional arguments passed over.

Details

These functions can be used to extract details from fitted models. `print` shows a dense representation of the model fit.

`[` can be used to define a new object with a subset of covariates from the original fit.

The function `coef` extracts the regression coefficients for each covariate listed separately. For the function `expectreg.boost` this is not possible.

Value

`[` returns a new object of class expectreg with a subset of covariates from the original fit.

`resid` returns the residuals in order of the response.

`fitted` returns the overall fitted values $\hat{y}$ while `effects` returns the values for each covariate in a list.

`coef` returns a list of all regression coefficients separately for each covariate.

Author(s)

Fabian Otto- Sobotka
Carl von Ossietzky University Oldenburg
https://uol.de

Elmar Spiegel
Georg August University Goettingen https://www.uni-goettingen.de

References

Schnabel S and Eilers P (2009) *Optimal expectile smoothing* Computational Statistics and Data Analysis, 53:4168-4177

Sobotka F and Kneib T (2010) *Geoadditive Expectile Regression* Computational Statistics and Data Analysis, doi: 10.1016/j.csda.2010.11.015.

See Also

expectreg.ls, expectreg.boost, expectreg.qp

Examples

```
data(dutchboys)

expreg <- expectreg.ls(hgt ~ rb(age,"pspline"),data=dutchboys,smooth="f",
                       expectiles=c(0.05,0.2,0.8,0.95))

print(expreg)

coef(expreg)

new.d = dutchboys[1:10,]
new.d[,2] = 1:10

predict(expreg,newdata=new.d)
```

Mqreg *Semiparametric M-Quantile Regression*

Description

Robust M-quantiles are estimated using an iterative penalised reweighted least squares approach. Effects using quadratic penalties can be included, such as P-splines, Markov random fields or Kriging.

Usage

```
Mqreg(formula, data = NULL, smooth = c("schall", "acv", "fixed"),
      estimate = c("iprls", "restricted"),lambda = 1, tau = NA, robust = 1.345,
      adaptive = FALSE, ci = FALSE, LSMaxCores = 1)
```

Arguments

formula	An R formula object consisting of the response variable, '~' and the sum of all effects that should be taken into consideration. Each effect has to be given through the function rb.
data	Optional data frame containing the variables used in the model, if the data is not explicitely given in the formula.

`estimate`	Character string defining the estimation method that is used to fit the expectiles. Further detail on all available methods is given below.
`smooth`	There are different smoothing algorithms that should prevent overfitting. The 'schall' algorithm iterates the smoothing penalty `lambda` until it converges, the asymmetric cross-validation 'acv' minimizes a score-function using `nlm` or the function uses a fixed penalty.
`lambda`	The fixed penalty can be adjusted. Also serves as starting value for the smoothing algorithms.
`tau`	In default setting, the expectiles (0.01,0.02,0.05,0.1,0.2,0.5,0.8,0.9,0.95,0.98,0.99) are calculated. You may specify your own set of expectiles in a vector. The option may be set to 'density' for the calculation of a dense set of expectiles that enhances the use of `cdf.qp` and `cdf.bundle` afterwards.
`robust`	Robustness constant in M-estimation. See `Details` for definition.
`adaptive`	Logical. Whether the robustness constant is adapted along the covariates.
`ci`	Whether a covariance matrix for confidence intervals and the summary function is calculated.
`LSMaxCores`	How many cores should maximal be used by parallelization

Details

In the least squares approach the following loss function is minimised:

$S = \sum_{i=1}^{n} w_p(y_i - m_i(p))^2$

with weights

$w_p(u) = (-(1-p)*c*(u_i < -c) + (1-p)*u_i*(u_i < 0\&u_i >= -c) + p*u_i*(u_i >= 0\&u_i < c) + p*c*(u_i >= c))/u_i$

for quantiles and

$w_p(u) = -(1-p)*c*(u_i < -c) + (1-p)*u_i*(u_i < 0\&u_i >= -c) + p*u_i*(u_i >= 0\&u_i < c) + p*c*(u_i >= c)$

for expectiles, with standardised residuals $u_i = 0.6745*(y_i - m_i(p))/median(y - m(p))$ and robustness constant c.

Value

An object of class 'expectreg', which is basically a list consisting of:

`lambda`	The final smoothing parameters for all expectiles and for all effects in a list. For the restricted and the bundle regression there are only the mean and the residual lambda.
`intercepts`	The intercept for each expectile.
`coefficients`	A matrix of all the coefficients, for each base element a row and for each expectile a column.
`values`	The fitted values for each observation and all expectiles, separately in a list for each effect in the model, sorted in order of ascending covariate values.
`response`	Vector of the response variable.

covariates	List with the values of the covariates.
formula	The formula object that was given to the function.
asymmetries	Vector of fitted expectile asymmetries as given by argument expectiles.
effects	List of characters giving the types of covariates.
helper	List of additional parameters like neighbourhood structure for spatial effects or 'phi' for kriging.
design	Complete design matrix.
fitted	Fitted values $\hat{y}$.

plot, predict, resid, fitted, effects and further convenient methods are available for class 'expectreg'.

Author(s)

Monica Pratesi
University Pisa
https://www.unipi.it

M. Giovanna Ranalli
University Perugia
https://www.unipg.it

Nicola Salvati
University Perugia
https://www.unipg.it

Fabian Otto-Sobotka
University Oldenburg
https://uol.de

References

Pratesi M, Ranalli G and Salvati N (2009) *Nonparametric M-quantile regression using penalised splines* Journal of Nonparametric Statistics, 21:3, 287-304.

Otto-Sobotka F, Ranalli G, Salvati N, Kneib T (2019) *Adaptive Semiparametric M-quantile Regression* Econometrics and Statistics 11, 116-129.

See Also

expectreg.ls, rqss

Examples

```
data("lidar", package = "SemiPar")

m <- Mqreg(logratio~rb(range,"pspline"),data=lidar,smooth="f",
                        tau=c(0.05,0.5,0.95),lambda=10)
plot(m,rug=FALSE)
```

---

northger.bnd   *Regions of northern Germany - boundary format*

Description

Map of northern Germany, represented in the boundary format (bnd) as defined in the package BayesX.

Usage

```
data(northger.bnd)
```

Format

The format is: List of 145 - attr(*, "class")= chr "bnd" - attr(*, "height2width")= num 1.54 - attr(*, "surrounding")=List of 145 - attr(*, "regions")= chr [1:145] "1001" "1002" "1003" "1004" ...

Details

For details about the format see read.bnd.

Source

Thomas Kneib
Georg August University Goettingen
https://www.uni-goettingen.de

Examples

```
data(northger.bnd)

drawmap(map=northger.bnd,mar.min=NULL)
```

pemq *The "expectiles-meet-quantiles" distribution family.*

Description

Density, distribution function, quantile function, random generation, expectile function and expectile distribution function for a family of distributions for which expectiles and quantiles coincide.

Usage

```
pemq(z,ncp=0,s=1)
demq(z,ncp=0,s=1)
qemq(q,ncp=0,s=1)
remq(n,ncp=0,s=1)
eemq(asy,ncp=0,s=1)
peemq(e,ncp=0,s=1)
```

Arguments

ncp	non centrality parameter and mean of the distribution.
s	scaling parameter, has to be positive.
z, e	vector of quantiles / expectiles.
q, asy	vector of asymmetries / probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.

Details

This distribution has the cumulative distribution function: $F(x; ncp, s) = \frac{1}{2}(1 + sgn(\frac{x-ncp}{s})\sqrt{1 - \frac{2}{2+(\frac{x-ncp}{s})^2}})$

and the density: $f(x; ncp, s) = \frac{1}{s}(\frac{1}{2+(\frac{x-ncp}{s})^2})^{\frac{3}{2}}$

It has infinite variance, still can be scaled by the parameter `s`. It has mean `ncp`. In the canonical parameters it is equal to a students-t distribution with 2 degrees of freedom. For $s = \sqrt{2}$ it is equal to a distribution introduced by Koenker(2005).

Value

`demq` gives the density, `pemq` and `peemq` give the distribution function, `qemq` gives the quantile function, `eemq` computes the expectiles numerically and is only provided for completeness, since the quantiles = expectiles can be determined analytically using `qemq`, and `remq` generates random deviates.

Author(s)

Fabian Otto- Sobotka
Carl von Ossietzky University Oldenburg
https://uol.de

Thomas Kneib
Georg August University Goettingen
https://www.uni-goettingen.de

References

Koenker R (2005) *Quantile Regression* Cambridge University Press, New York

See Also

enorm

Examples

```
x <- seq(-5,5,length=100)
plot(x,demq(x))
plot(x,pemq(x,ncp=1))

z <- remq(100,s=sqrt(2))
plot(z)

y <- seq(0.02,0.98,0.2)
qemq(y)
eemq(y)

pemq(x) - peemq(x)
```

---

plot.expectreg *Default expectreg plotting*

Description

Takes a expectreg object and plots the estimated effects.

Usage

```
## S3 method for class 'expectreg'
plot(x, rug = TRUE, xlab = NULL, ylab = NULL, ylim = NULL,
legend = TRUE, ci = FALSE, ask = NULL, cex.main = 2, mar.min = 5, main = NULL,
cols = "rainbow", hcl.par = list(h = c(260, 0), c = 185, l = c(30, 85)),
ylim_spat = NULL, ylim_factor = NULL, range_warning = TRUE, add_intercept = TRUE, ...)
```

Arguments

`x`	An object of class `expectreg` as returned e.g. by the function `expectreg.ls`.
`rug`	Boolean. Whether nonlinear effects are displayed in a rug plot.
`xlab, ylab, ylim`	Graphic parameters. `xlab` should match the number of covariates.
`legend`	Boolean. Decides whether a legend is added to the plots.
`ci`	Boolean. Whether confidence intervals and significances should be plotted.
`ask`	Should always be asked before a new plot is printed.
`cex.main`	Font size of main
`mar.min`	Minimal margins, important when markov fields are plotted
`main`	Vector of main per plot
`cols`	Colours sheme of plots. Default is rainbow. Alternatively `hcl` can be used.
`hcl.par`	Parameters to specify the hcl coulour sheme.
`ylim_spat`	y_limits of the markov random field and all other spatial methods.
`ylim_factor`	y_limits of the plots of factor covariates.
`range_warning`	Should a warning be printed in the graphic if the range of the markov random field/factor plot is larger than the specified limits in `markov_ylim`/`factors_ylim`.
`add_intercept`	Should the intercept be added to the plots of splines?
`...`	

Details

The `plot` function gives a visual representation of the fitted expectiles separately for each covariate.

Value

No return value, only graphical output.

Author(s)

Fabian Otto- Sobotka
Carl von Ossietzky University Oldenburg
https://uol.de

Elmar Spiegel
Georg August University Goettingen
https://www.uni-goettingen.de

References

Schnabel S and Eilers P (2009) *Optimal expectile smoothing* Computational Statistics and Data Analysis, 53:4168-4177

Sobotka F and Kneib T (2010) *Geoadditive Expectile Regression* Computational Statistics and Data Analysis, doi: 10.1016/j.csda.2010.11.015.

See Also

expectreg.ls, expectreg.boost, expectreg.qp

Examples

```
data(dutchboys)

expreg <- expectreg.ls(hgt ~ rb(age,"pspline"),data=dutchboys,smooth="f",
                       expectiles=c(0.05,0.2,0.8,0.95))
plot(expreg)
```

---

quant.bundle *Restricted expectile regression of additive models*

Description

A location-scale model to fit generalized additive models with least asymmetrically weighted squares to obtain the graphs of different expectiles or quantiles for continuous, spatial or random effects.

Usage

```
quant.bundle(formula, data = NULL, smooth = c("schall", "acv", "fixed"),
             lambda = 1, quantiles = NA, simple = TRUE)
```

Arguments

formula	An R formula object consisting of the response variable, '~' and the sum of all effects that should be taken into consideration. Each effect has to be given through the function rb.
data	Optional data frame containing the variables used in the model, if the data is not explicitely given in the formula.
smooth	There are different smoothing algorithms that should prevent overfitting. The 'schall' algorithm iterates the smoothing penalty lambda until it converges, the asymmetric cross-validation 'acv' minimizes a score-function using nlm or the function uses a fixed penalty.
lambda	The fixed penalty can be adjusted. Also serves as starting value for the smoothing algorithms.
quantiles	In default setting, the quantiles (0.01,0.02,0.05,0.1,0.2,0.5,0.8,0.9,0.95,0.98,0.99) are calculated. You may specify your own set of expectiles in a vector.
simple	A binary variable depicting if the restricted expectiles (TRUE) or the bundle is used as basis for the quantile bundle.

Details

In least asymmetrically weighted squares (LAWS) each expectile is fitted by minimizing:
$S = \sum_{i=1}^{n} w_i(p)(y_i - \mu_i(p))^2$
with
$w_i(p) = p1_{(y_i > \mu_i(p))} + (1-p)1_{(y_i < \mu_i(p))}$.
The restricted version fits the 0.5 expectile at first and then the residuals. Afterwards the other expectiles are fitted as deviation by a factor of the residuals from the mean expectile. This algorithm is based on He(1997). The advantage is that expectile crossing cannot occur, the disadvantage is a suboptimal fit in certain heteroscedastic settings. Also, since the number of fits is significantly decreased, the restricted version is much faster.

The expectile bundle has a resemblence to the restricted regression. At first, a trend curve is fitted and then an iteration is performed between fitting the residuals and calculating the deviation factors for all the expectiles until the results are stable. Therefore this function shares the (dis)advantages of the restricted.

The quantile bundle uses either the restricted expectiles or the bundle to estimate a dense set of expectiles. Next this set is used to estimate a density with the function `cdf.bundle`. From this density quantiles are determined and inserted to the calculated bundle model. This results in an estimated location-scale model for quantile regression.

Value

An object of class 'expectreg', which is basically a list consisting of:

`lambda`	The final smoothing parameters for all expectiles and for all effects in a list. For the restricted and the bundle regression there are only the mean and the residual lambda.
`intercepts`	The intercept for each expectile.
`coefficients`	A matrix of all the coefficients, for each base element a row and for each expectile a column.
`values`	The fitted values for each observation and all expectiles, separately in a list for each effect in the model, sorted in order of ascending covariate values.
`response`	Vector of the response variable.
`covariates`	List with the values of the covariates.
`formula`	The formula object that was given to the function.
`asymmetries`	Vector of fitted expectile asymmetries as given by argument `expectiles`.
`effects`	List of characters giving the types of covariates.
`helper`	List of additional parameters like neighbourhood structure for spatial effects or 'phi' for kriging.
`trend.coef`	Coefficients of the trend function.
`residual.coef`	Vector of the coefficients the residual curve was fitted with.
`asymmetry`	Vector of the asymmetry factors for all expectiles.
`design`	Complete design matrix.
`fitted`	Fitted values $\hat{y}$.

`plot`, `predict`, `resid`, `fitted` and `effects` methods are available for class 'expectreg'.

Author(s)

Fabian Otto- Sobotka
Carl von Ossietzky University Oldenburg
https://uol.de

Thomas Kneib
Georg August University Goettingen
https://www.uni-goettingen.de

Sabine Schnabel
Wageningen University and Research Centre
https://www.wur.nl

Paul Eilers
Erasmus Medical Center Rotterdam
https://www.erasmusmc.nl

References

Schnabel S and Eilers P (2009) *Optimal expectile smoothing* Computational Statistics and Data Analysis, 53:4168-4177

He X (1997) *Quantile Curves without Crossing* The American Statistician, 51(2):186-192

Schnabel S and Eilers P (2011) *A location scale model for non-crossing expectile curves* (working paper)

Sobotka F and Kneib T (2010) *Geoadditive Expectile Regression* Computational Statistics and Data Analysis, doi: 10.1016/j.csda.2010.11.015.

See Also

rb, expectreg.boost

Examples

```
qb = quant.bundle(dist ~ rb(speed),data=cars,smooth="f",lambda=5)
plot(qb)

qbund <- quant.bundle(dist ~ rb(speed),data=cars,smooth="f",lambda=50000,simple=FALSE)
```

---

rb *Creates base for a regression based on covariates*

Description

Based on given observations a matrix is created that creates a basis e.g. of splines or a markov random field that is evaluated for each observation. Additionally a penalty matrix is generated. Shape constraint p-spline bases can also be specified.

Usage

```
rb(x, type = c("pspline", "2dspline", "markov", "krig", "random",
"ridge", "special", "parametric", "penalizedpart_pspline"), B_size = 20,
B = NA, P = NA, bnd = NA, center = TRUE, by = NA, ...)

mono(x, constraint = c("increase", "decrease", "convex", "concave", "flatend"),
by = NA)
```

Arguments

`x`	Data vector, matrix or data frame. In case of '2dspline', or 'krig' `type` number of variables of `x` has to be 2. More dimensions are allowed in 'ridge' and 'special' `type`. 'markov' and 'random' `type` require a vector of a factor.
`type`	Character string defining the type of base that is generated for the given variable(s) `x`. Further description of the possible options is given below in `details`.
`B_size`	Number of basis functions of psplines. Default is 20.
`B`	For the 'special' `type` the base `B` and penalization matrix `P` are entered manually. The data frame or matrix needs as many rows as observations in `x` and as many columns as `P`.
`P`	Square matrix that has to be provided in 'special' case and with 'markov' `type` if no `bnd` is given.
`bnd`	Object of class `bnd`, required with 'markov' type if `P` is not given. See `read.bnd`.
`center`	Logical to state whether the basis shall be centered in order to fit additive models with one central intercept.
`by`	An optional variable defining varying coefficients, either a factor or numeric variable. Per default treatment coding is used. Note that the main effect needs to be specified in a separate basis.
`constraint`	Character string defining the type of shape constraint that is imposed on the spline curve. The last option 'flatend' results in constant functions at the covariate edges.
`...`	Currently not used.

Details

Possible `type`s of bases:

pspline Penalized splines made upon `B_size` equidistant knots with degree 3. The penalization matrix consists of differences of the second order, see `diff`.

2dspline Tensor product of 2 p-spline bases with the same properties as above.

markov Gaussian markov random field with a neighbourhood structure given by `P` or `bnd`.

krig 'kriging' produces a 2-dimensional base, which is calculated as exp(-r/phi)*(1+r/phi) where `phi` is the maximum euclidean distance between two knots divided by a constant.

random A 'random' effect is like the 'markov' random field based on a categorial variable, and since there is no neighbourhood structure, P = I.

ridge In a 'ridge' regression, the base is made from the independent variables while the goal is to determine significant variables from the coefficients. Therefore no penalization is used (P = I).

special In the 'special' case, B and P are user defined.

parametric A parametric effect.

penalizedpart_pspline Penalized splines made upon B_size equidistant knots with degree 3. The penalization matrix consists of differences of the second order, see diff. Generally a P-spline of degree 3 with 2 order penalty can be splited in a linear trend and the deviation of the linear trend. Here only the wiggly deviation of the linear trend is kept. It is possible to combine it with the same covariate of type parametric

Value

List consisting of:

B	Matrix of the evaluated base, one row for each observation, one column for each base element.
P	Penalty square matrix, needed for the smoothing in the regression.
x	The observations x given to the function.
type	The type as given to the function.
bnd	The bnd as given to the function, only needed with 'markov' type.
Zspathelp	Matrix that is also only needed with 'markov' type for calculation of the fitted values.
phi	Constant only needed with 'kriging' type, otherwise 'NA'.
center	The boolean value of the argument center.
by	The variable included in the by argument if available.
xname	Name of the variable x given to the function. Modified by its type.
constraint	Part of the penalty matrix.
B_size	Same as input
P_orig	Original penalty P before restructuring. Used for model selection.
B_mean	Original mean of design matrix B before centering.
param_center	Parameters of centering the covariate.
nbp	Number of penalized parameters in this covariate.
nbunp	Number of unpenalized parameters in this covariate.

Warning

The pspline is now centered around its mean. Thus different results compared to old versions of expectreg occure.

Author(s)

Fabian Otto- Sobotka
Carl von Ossietzky University Oldenburg
https://uol.de

Thomas Kneib, Elmar Spiegel
Georg August University Goettingen
https://www.uni-goettingen.de

Sabine Schnabel
Wageningen University and Research Centre
https://www.wur.nl

Paul Eilers
Erasmus Medical Center Rotterdam
https://www.erasmusmc.nl

References

Fahrmeir L and Kneib T and Lang S (2009) *Regression* Springer, New York

See Also

quant.bundle, expectreg.ls

Examples

```
x <- rnorm(100)

bx <- rb(x,"pspline")

y <- sample(10,100,replace=TRUE)

by <- rb(y,"random")
```

---

update.expectreg — *Update given expectreg model*

Description

Updates a given expectreg model with the specified changes

Usage

```
## S3 method for class 'expectreg'
update(object, add_formula, data = NULL, estimate = NULL,
smooth = NULL, lambda = NULL, expectiles = NULL, delta_garrote = NULL, ci = NULL,
...)
```

Arguments

object	of class expectreg
add_formula	update for formula
data	Should other data be used
estimate	Change estimate
smooth	Change smooth
lambda	Change lambda
expectiles	Change asymmetries
delta_garrote	Change delta_garrote
ci	Change ci
...	

Details

Re-estimates the given model, with the specified changes. If nothing is specified the characteristics of the original model are used. Except lambda here the default 1 is used as initial value.

Value

object of class expectreg

Author(s)

Elmar Spiegel
Helmholtz Zentrum Muenchen
https://www.helmholtz-munich.de

See Also

update, update.formula

Examples

```
data(india)

model1<-expectreg.ls(stunting~rb(cbmi),smooth="fixed",data=india,lambda=30,
                    estimate="restricted",expectiles=c(0.01,0.05,0.2,0.8,0.95,0.99))
plot(model1)

# Change formula and update model
add_formula<-.~.+rb(cage)
update_model1<-update(model1,add_formula)
plot(update_model1)

# Use different asymmetries and update model
update_model2<-update(model1,expectiles=c(0.1,0.5,0.9))
plot(update_model2)
```

Index