

Package ‘expectreg’

July 2, 2014

Type Package

Title Expectile and Quantile Regression

Version 0.39

Date 2014-03-05

Author Fabian Sobotka <fabian.sobotka@wiwi.uni-goettingen.de>, Sabine Schnabel <sabine.schnabel@wur.nl> and Linda Schulze Waltrup <lschulze_waltrup@stat.uni-muenchen.de> with contributions from Paul Eilers <p.eilers@erasmusmc.nl>, Thomas Kneib <tkneib@uni-goettingen.de> and Goeran Kauermann <goeran.kauermann@stat.uni-muenchen.de>.

Maintainer Fabian Sobotka <fabian.sobotka@wiwi.uni-goettingen.de>

Depends R (>= 2.14.0), stats, parallel, mboost (>= 2.1.0), BayesX (>= 0.2-4)

Imports splines, quadprog

Suggests SemiPar, fields

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

LazyLoad yes

LazyData yes

NeedsCompilation no

Repository CRAN

Date/Publication 2014-03-05 14:39:34

R topics documented:

expectreg-package	2
cdf.qp	4
dutchboys	6
enorm	7
expectile	9
expectreg.boost	10
expectreg.ls	12
india	15
india.bnd	16
methods	17
northger.bnd	19
pemq	20
quant.bundle	21
rb	24
Index	27

expectreg-package	<i>Expectile and Quantile Regression</i>
-------------------	--

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:	0.39
Date:	2014-03-05
License:	GPL-2
LazyLoad:	yes
LazyData:	yes

- This package requires the packages [BayesX](#), [mboost](#), [splines](#) and [quadprog](#).

Author(s)

Fabian Sobotka
 Georg August University Goettingen
<http://www.uni-goettingen.de>

Sabine Schnabel
Wageningen University and Research Centre
<http://www.wur.nl>

Linda Schulze Waltrup
Ludwig Maximilian University Munich
<http://www.lmu.de>

with contributions from

Paul Eilers
Erasmus Medical Center Rotterdam
<http://www.erasmusmc.nl>

Thomas Kneib
Georg August University Goettingen
<http://www.uni-goettingen.de>

Goeran Kauermann
Ludwig Maximilian University Munich
<http://www.lmu.de>

Maintainer: Fabian Sobotka <fabian.sobotka@wiwi.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
- 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")
## The calculation of expectiles for given distributions
enorm(0.1)
## 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)
```

```
cdf.bundle(bundle, qout = NA, extrap = 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.
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.

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
<http://www.lmu.de>

Fabian Sobotka
Georg August University Goettingen
<http://www.uni-goettingen.de>

Sabine Schnabel
Wageningen University and Research Centre
<http://www.wur.nl>

Paul Eilers
Erasmus Medical Center Rotterdam
<http://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, Kauermann G and Kneib T (2011) *Comparing Expectiles and Quantiles Regarding Efficiency* Working Paper.

See Also

[expectreg.ls](#), [expectreg.qp](#)

Examples

```
d = expectreg.ls(dist ~ rb(speed), data=cars, smooth="f", lambda=5, estimate="restricted")
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 explicitly, 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 $G(z) = \int_{-\infty}^z u f(u) du$.

Value

Vector of the expectiles for the desired distribution.

Author(s)

Fabian Sobotka, Thomas Kneib
 Georg August University Goettingen
<http://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)
```

```
qnorm(x)
e = enorm(x)
```

```
penorm(e)
```

expectile	<i>Sample Expectiles</i>
-----------	--------------------------

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 y and x 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 Sobotka
 Georg August University Goettingen
<http://www.uni-goettingen.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)

quant.boost(formula, data, mstop = NA, quantiles = NA, cv = TRUE)
```

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 <code>cvrisk</code> . If set to FALSE, the results from mstop iterations are used.

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 <code>expectiles</code> .
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 Sobotka, Thomas Kneib
Georg August University Goettingen
<http://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")
expreg <- expectreg.boost(logratio ~ bbs(range), lidar, mstop=300, expectiles=c(0.05,0.5,0.95))
plot(expreg)
```

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", "gcv", "cvgrid", "aic", "bic", "lcurve", "fixed"),
             lambda = 1, expectiles = NA, ci = FALSE)
```

```
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 explicitly 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' minimizes a score-function using nlm or with a grid search by 'cvgrid' or the function uses a fixed penalty. The numerical minimisation is also possible with AIC or BIC as score. The L-curve is a new experimental grid search.
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 <code>cdf.qp</code> and <code>cdf.bundle</code> afterwards.
ci	Whether a covariance matrix for confidence intervals and a <code>summary</code> is calculated.

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 resemblance 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. "schall" smoothing does not yet work for sheets.

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 <code>expectiles</code> .
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)

Fabian Sobotka, Thomas Kneib
Georg August University Goettingen
<http://www.uni-goettingen.de>

Sabine Schnabel
Wageningen University and Research Centre
<http://www.wur.nl>

Paul Eilers
Erasmus Medical Center Rotterdam
<http://www.erasmusmc.nl>

Linda Schulze Waltrup, Goeran Kauermann
Ludwig Maximilians University Muenchen
<http://www.uni-muenchen.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

```

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=0.000001,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.

```

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 district 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 <http://www.uni-goettingen.de/de/64861.html>

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'  
plot(x,rug = TRUE, xlab = NULL, ylab = NULL, ylim = NULL, legend = TRUE, ci = FALSE, ...)
```

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

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

```
## S3 method for class 'expectreg'  
predict(object, newdata = NULL, ...)
```

```
## 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`.

<code>rug</code>	Boolean. Whether nonlinear effects are displayed in a rug plot.
<code>xlab, ylab, ylim</code>	Graphic parameters. <code>xlab</code> should match the number of covariates.
<code>legend</code>	Boolean. Decides whether a legend is added to the plots.
<code>ci</code>	Boolean. Whether confidence intervals and significances should be plotted.
<code>newdata</code>	Optionally, a data frame in which to look for variables with which to predict.
<code>i</code>	Covariate numbers to be kept in subset.
<code>level</code>	Coverage probability of the generated confidence intervals.
<code>parm</code>	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.
<code>...</code>	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.

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

`[` can be used to define a new object 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.

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

Author(s)

Fabian Sobotka
Georg August University Goettingen
<http://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)

print(expreg)

coef(expreg)

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

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

`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
<http://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: $F(x; ncp, s) = \frac{1}{2}(1 + \text{sgn}(\frac{x-ncp}{s}))\sqrt{1 - \frac{2}{2+(\frac{x-ncp}{s})^2}}$

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 Sobotka, Thomas Kneib
Georg August University Goettingen
<http://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))

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

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

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 explicitly 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 <code>nlm</code> 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 resemblance 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 <code>expectiles</code> .
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 Sobotka, Thomas Kneib
Georg August University Goettingen
<http://www.uni-goettingen.de>

Sabine Schnabel
Wageningen University and Research Centre
<http://www.wur.nl>

Paul Eilers
Erasmus Medical Center Rotterdam
<http://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", "radial", "krig",
              "random", "ridge", "special", "parametric"),
    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', 'radial' 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	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.

Details

Possible types of bases:

pspline	Penalized splines made upon 20 equidistant knots and with degree 2. The penalization matrix consists of different
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.
radial	2-dimensional base, knots are subset of observations, base is calculated as $r^2 \cdot \log(r)$ with r equalling the euclidean distance.
krig	'kriging' produces basically the same base as 'radial', but the base is calculated as $\exp(-r/\phi) \cdot (1+r/\phi)$ where ϕ is the range.
random	A 'random' effect is like the 'markov' random field based on a categorical variable, and since there is no neighbourhood structure, the base is calculated as $r^2 \cdot \log(r)$.
ridge	In a 'ridge' regression, the base is made from the independent variables while the goal is to determine significant variables.
special	In the 'special' case, B and P are user defined.
parametric	A parametric effect.

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.

Author(s)

Fabian Sobotka, Thomas Kneib
 Georg August University Goettingen
<http://www.uni-goettingen.de>

Sabine Schnabel
 Wageningen University and Research Centre
<http://www.wur.nl>

Paul Eilers
 Erasmus Medical Center Rotterdam
<http://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")
```

Index

- *Topic **datasets**
 - dutchboys, 6
 - india, 15
 - india.bnd, 16
 - northger.bnd, 19
 - *Topic **distribution**
 - cdf.qp, 4
 - enorm, 7
 - pemq, 20
 - *Topic **methods**
 - methods, 17
 - *Topic **models**
 - expectreg-package, 2
 - expectreg.boost, 10
 - expectreg.ls, 12
 - quant.bundle, 21
 - rb, 24
 - *Topic **multivariate**
 - cdf.qp, 4
 - expectreg-package, 2
 - expectreg.boost, 10
 - expectreg.ls, 12
 - quant.bundle, 21
 - rb, 24
 - *Topic **nonlinear**
 - cdf.qp, 4
 - expectreg-package, 2
 - expectreg.boost, 10
 - expectreg.ls, 12
 - quant.bundle, 21
 - rb, 24
 - *Topic **nonparametric**
 - cdf.qp, 4
 - expectile, 9
 - expectreg-package, 2
 - expectreg.boost, 10
 - expectreg.ls, 12
 - quant.bundle, 21
 - rb, 24
 - *Topic **package**
 - expectreg-package, 2
 - *Topic **regression**
 - expectreg-package, 2
 - expectreg.boost, 10
 - expectreg.ls, 12
 - quant.bundle, 21
 - rb, 24
 - *Topic **smooth**
 - expectreg-package, 2
 - expectreg.boost, 10
 - expectreg.ls, 12
 - quant.bundle, 21
 - *Topic **spatial**
 - rb, 24
 - *Topic **univariate**
 - expectile, 9
- [.expectreg (methods), 17
- BayesX, 2, 3, 16, 19
- bbs, 10, 12
- cdf.bundle, 13, 22
- cdf.bundle (cdf.qp), 4
- cdf.qp, 4, 13
- coef.expectreg (methods), 17
- coefficients.expectreg (methods), 17
- confint.expectreg (methods), 17
- cvrisk, 11, 12
- demq (pemq), 20
- diff, 24
- dutchboys, 6
- ebeta (enorm), 7
- echisq (enorm), 7
- eemq, 8
- eemq (pemq), 20
- eenorm (expectile), 9
- eeplot (expectile), 9

eexp (enorm), 7
effects, 11, 14, 23
effects.expectreg (methods), 17
egamma (enorm), 7
elnorm (enorm), 7
enorm, 7, 21
et (enorm), 7
eunif (enorm), 7
expectile, 9
expectreg (expectreg-package), 2
expectreg-package, 2
expectreg.boost, 10, 14, 18, 23
expectreg.ls, 5, 10, 12, 12, 17, 18, 25
expectreg.qp, 5, 18
expectreg.qp (expectreg.ls), 12

fitted, 11, 14, 23
fitted.expectreg (methods), 17
fitted.values.expectreg (methods), 17

gamboost, 10, 12

india, 15
india.bnd, 15, 16

mboost, 2, 3
methods, 17
mono (rb), 24

nlm, 12, 22
northger.bnd, 19

pebeta (enorm), 7
pechisq (enorm), 7
peemq (pemq), 20
peexp (enorm), 7
pegamma (enorm), 7
peInorm (enorm), 7
pemq, 20
penorm (enorm), 7
pet (enorm), 7
peunif (enorm), 7
plot, 11, 14, 23
plot.expectreg (methods), 17
predict, 11, 14, 23
predict.expectreg (methods), 17
print.expectreg (methods), 17

qemq (pemq), 20
qqnorm, 9
qqplot, 9
quant.boost (expectreg.boost), 10
quant.bundle, 21, 25
quantile, 9, 10

rb, 12, 14, 21, 23, 24
read.bnd, 16, 19, 24
remq (pemq), 20
resid, 11, 14, 23
resid.expectreg (methods), 17
residuals.expectreg (methods), 17

splines, 2
summary, 13
summary.expectreg (methods), 17