

Package ‘polspline’

August 21, 2009

Version 1.1.4

Date 2009-08-21

Title Polynomial spline routines

Author Charles Kooperberg <clk@fhcrc.org>

Maintainer Charles Kooperberg <clk@fhcrc.org>

Description Routines for the polynomial spline fitting routines hazard regression, hazard estimation with flexible tails, logspline, lspec, polyclass, and polymars, by C. Kooperberg and co-authors

License GPL (>= 2)

Repository CRAN

Date/Publication 2009-08-21 17:23:18

R topics documented:

beta.polyclass	2
clspec	3
cpolyclass	5
design.polymars	6
dhare	7
dheft	8
dlogspline	10
doldlogspline	11
hare	12
heft	14
logspline	16
lspec	19
oldlogspline	21
oldlogspline.to.logspline	23
persp.polymars	25
plot.hare	26
plot.heft	27

plot.logsplines	28
plot.lspec	30
plot.oldsplines	31
plot.polyclass	32
plot.polymars	34
polyclass	35
polymars	39
predict.polymars	43
summary.hare	44
summary.heft	45
summary.logsplines	47
summary.lspec	48
summary.oldsplines	49
summary.polyclass	50
summary.polymars	51
testhare	52
unstrip	53
xhare	54

Index 56

beta.polyclass	<i>Polychlass: polychotomous regression and multiple classification</i>
----------------	-------------------------------------------------------------------------

Description

Produces a beta-plot for a `polyclass` object.

Usage

```
beta.polyclass(fit, which, xsp = 0.4, cex)
```

Arguments

<code>fit</code>	<code>polyclass</code> object, typically the result of <code>polyclass</code> .
<code>which</code>	which classes should be compared? Default is to compare all classes.
<code>xsp</code>	location of the vertical line to the left of the axis. Useful for making high quality, device dependent, graphics.
<code>cex</code>	character size. Default is whatever the present character size is. Useful for making high quality, device dependent, graphics.

Value

A beta plot. One line for each basis function. The left part of the plot indicates the basis function, the right half the relative location of the betas (coefficients) of that basis function, normalized with respect to parent basis functions, for all classes. The scaling is supposed to suggest a relative importance of the basis functions. This may suggest which basis functions are important for separating particular classes.

Note

This is not a generic function, and the complete name, `beta.polyclass`, has to be specified.

Author(s)

Charles Kooperberg [⟨clk@fhcrc.org⟩](mailto:clk@fhcrc.org).

References

Charles Kooperberg, Smarajit Bose, and Charles J. Stone (1997). Polychotomous regression. *Journal of the American Statistical Association*, **92**, 117–127.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[polyclass](#), [plot.polyclass](#), [summary.polyclass](#), [cpolyclass](#), [ppolyclass](#), [rpolyclass](#).

Examples

```
data(iris)
fit.iris <- polyclass(iris[,5], iris[,1:4])
beta.polyclass(fit.iris)
```

clspec

Lspec: logspline estimation of a spectral distribution

Description

Autocorrelations, autocovariances (`clspec`), spectral densities and line spectrum (`dlspec`), spectral distributions (`plspec`) or a random time series (`rlspec`) from a model fitted with `lspec`.

Usage

```
clspec(lag, fit, cov = TRUE, mm)
dlspec(freq, fit)
plspec(freq, fit, mm)
rlspec(n, fit, mean = 0, cosmodel = FALSE, mm)
```

Arguments

lag	vector of integer-valued lags for which the autocorrelations or autocovariances are to be computed.
fit	<code>lspec</code> object, typically the result of <code>lspec</code> .
cov	compute autocovariances (TRUE) or autocorrelations (FALSE).

mm	number of points used in integration and the fft. Default is the smallest power of two larger than $\max(\text{fit}\$sample, \max(\text{lag}), 1024)$ for <code>clspec</code> and <code>plspect</code> or the smallest power of two larger than $\max(\text{fit}\$sample, n, \max(\text{lag}), 1024)$ for <code>rlspect</code> .
freq	vector of frequencies. For <code>plspect</code> frequencies should be between $-\pi$ and π .
n	length of the random time series to be generated.
mean	mean level of the time series to be generated.
cosmodel	indicate that the data should be generated from a model with constant harmonic terms rather than a true Gaussian time series.

Value

Autocovariances or autocorrelations (`clspect`); values of the spectral distribution at the requested frequencies. (`plspect`); random time series of length `n` (`rlspect`); or a list with three components (`dlspect`):

d	the spectral density evaluated at the vector of frequencies,
modfreq	modified frequencies of the form $\frac{2\pi j}{T}$ that are close to the frequencies that were requested,
m	mass of the line spectrum at the modified frequencies.

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg, Charles J. Stone, and Young K. Truong (1995). Log spline Estimation of a Possibly Mixed Spectral Distribution. *Journal of Time Series Analysis*, **16**, 359-388.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371-1470.

See Also

[lspec](#), [plot.lspec](#), [summary.lspec](#).

Examples

```
data(co2)
co2.detrend <- lm(co2~c(1:length(co2)))$residuals
fit <- lspect(co2.detrend)
clspect(0:12, fit)
plspect((0:314)/100, fit)
dlspect((0:314)/100, fit)
rlspect(length(co2), fit)
```

`cpolyclass`*Polychotomous regression and multiple classification*

Description

Classify new cases (`cpolyclass`), compute class probabilities for new cases (`ppolyclass`), and generate random multinomials for new cases (`rpolyclass`) for a `polyclass` model.

Usage

```
cpolyclass(cov, fit)
ppolyclass(data, cov, fit)
rpolyclass(n, cov, fit)
```

Arguments

<code>cov</code>	covariates. Should be a matrix with <code>fit\$ncov</code> columns. For <code>rpolyclass</code> <code>cov</code> should either have one row, in which case all random numbers are based on the same covariates, or <code>n</code> rows in which case each random number has its own covariates.
<code>fit</code>	<code>polyclass</code> object, typically the result of <code>polyclass</code> .
<code>data</code>	there are several possibilities. If <code>data</code> is a vector with as many elements as <code>cov</code> has rows, each element of <code>data</code> corresponds to a row of <code>cov</code> ; if only one value is given, the probability of being in that class is computed for all sets of covariates. If <code>data</code> is omitted, all class probabilities are provided.
<code>n</code>	number of pseudo random numbers to be generated.

Value

Most likely classes (`cpolyclass`), probabilities (`cpolyclass`), or random classes according to the estimated probabilities (`rpolyclass`).

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg, Smarajit Bose, and Charles J. Stone (1997). Polychotomous regression. *Journal of the American Statistical Association*, **92**, 117–127.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

`polyclass`, `plot.polyclass`, `summary.polyclass`, `beta.polyclass`.

Examples

```

data(iris)
fit.iris <- polyclass(iris[,5], iris[,1:4])
class.iris <- cpolyclass(iris[,1:4], fit.iris)
table(class.iris, iris[,5])
prob.setosa <- ppolyclass(1, iris[,1:4], fit.iris)
prob.correct <- ppolyclass(iris[,5], iris[,1:4], fit.iris)
rpolyclass(100, iris[64,1:4], fit.iris)

```

design.polymars *Polymars: multivariate adaptive polynomial spline regression*

Description

Produces a design matrix for a model of class `polymars`.

Usage

```
design.polymars(object, x)
```

Arguments

<code>object</code>	object of the class <code>polymars</code> , typically the result of <code>polymars</code> .
<code>x</code>	the predictor values at which the design matrix will be computed. The predictor values can be in a number of formats. It can take the form of a vector of length equal to the number of predictors in the original data set or it can be shortened to the length of only those predictors that occur in the model, in the same order as they appear in the original data set. Similarly, <code>x</code> can take the form of a matrix with the number of columns equal to the number of predictors in the original data set, or shortened to the number of predictors in the model.

Value

The design matrix corresponding to the fitted `polymars` model.

Author(s)

Charles Kooperberg

References

Charles Kooperberg, Smarajit Bose, and Charles J. Stone (1997). Polychotomous regression. *Journal of the American Statistical Association*, **92**, 117–127.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[polymars](#), [plot.polymars](#), [predict.polymars](#), [summary.polymars](#).

Examples

```
data(state)
state.pm <- polymars(state.region, state.x77, knots = 15, classify = TRUE, gcv = 1)
desmat <- design.polymars(state.pm, state.x77)
# compute traditional summary of the fit for the first class
summary(lm(((state.region=="Northeast")*1) ~ desmat -1))
```

 dhare

Hare: hazard regression

Description

Density (`dhare`), cumulative probability (`phare`), hazard rate (`hhare`), quantiles (`qhare`), and random samples (`rhare`) from a [hare](#) object.

Usage

```
dhare(q, cov, fit)
hhare(q, cov, fit)
phare(q, cov, fit)
qhare(p, cov, fit)
rhare(n, cov, fit)
```

Arguments

<code>q</code>	vector of quantiles. Missing values (NAs) are allowed.
<code>p</code>	vector of probabilities. Missing values (NAs) are allowed.
<code>n</code>	sample size. If <code>length(n)</code> is larger than 1, then <code>length(n)</code> random values are returned.
<code>cov</code>	covariates. There are several possibilities. If a vector of length <code>fit\$ncov</code> is provided, these covariates are used for all elements of <code>p</code> or <code>q</code> or for all random numbers. If a matrix of dimension <code>length(p)</code> , <code>length(q)</code> , or <code>n</code> by <code>fit\$ncov</code> is provided, the rows of <code>cov</code> are matched with the elements of <code>p</code> or <code>q</code> or every row of <code>cov</code> has its own random number. If a matrix of dimension <code>m</code> times <code>fit\$ncov</code> is provided, while <code>length(p) = 1</code> or <code>length(q) = 1</code> or <code>n = 1</code> , the single element of <code>p</code> or <code>q</code> is used <code>m</code> times, or <code>m</code> random numbers with different sets of covariates are generated.
<code>fit</code>	<code>hare</code> object, typically obtained from hare .

Details

Elements of `q` or `p` that are missing will cause the corresponding elements of the result to be missing.

Value

Densities (dhare), hazard rates (hhare), probabilities (phare), quantiles (qhare), or a random sample (rhare) from a [hare](#) object.

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg, Charles J. Stone and Young K. Truong (1995). Hazard regression. *Journal of the American Statistical Association*, **90**, 78-94.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[hare](#), [plot.hare](#), [summary.hare](#).

Examples

```
fit <- hare(testhare[,1], testhare[,2], testhare[,3:8])
dhare(0:10, testhare[117,3:8], fit)
hhare(0:10, testhare[1:11,3:8], fit)
phare(10, testhare[1:25,3:8], fit)
qhare((1:19)/20, testhare[117,3:8], fit)
rhare(10, testhare[117,3:8], fit)
```

dheft

Heft: hazard estimation with flexible tails

Description

Density (dheft), cumulative probability (pheft), hazard rate (hheft), quantiles (qheft), and random samples (rheft) from a [heft](#) object

Usage

```
dheft(q, fit)
hheft(q, fit)
pheft(q, fit)
qheft(p, fit)
rheft(n, fit)
```

Arguments

<code>q</code>	vector of quantiles. Missing values (NAs) are allowed.
<code>p</code>	vector of probabilities. Missing values (NAs) are allowed.
<code>n</code>	sample size. If <code>length(n)</code> is larger than 1, then <code>length(n)</code> random values are returned.
<code>fit</code>	heft object, typically obtained from heft .

Details

Elements of `q` or `p` that are missing will cause the corresponding elements of the result to be missing.

Value

Densities (`dheft`), hazard rates (`hheft`), probabilities (`pheft`), quantiles (`qheft`), or a random sample (`rheft`) from a [heft](#) object.

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg, Charles J. Stone and Young K. Truong (1995). Hazard regression. *Journal of the American Statistical Association*, **90**, 78-94.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[heft](#), [plot.heft](#), [summary.heft](#).

Examples

```
fit <- heft(testhare[,1],testhare[,2])
dheft(0:10,fit)
hheft(0:10,fit)
pheft(0:10,fit)
qheft((1:19)/20,fit)
rheft(10,fit)
```

`dlogspline`*Log spline Density Estimation*

Description

Density (`dlogspline`), cumulative probability (`plogspline`), quantiles (`qlogspline`), and random samples (`rlogspline`) from a log spline density that was fitted using the 1997 knot addition and deletion algorithm (`logspline`). The 1992 algorithm is available using the `oldlogspline` function.

Usage

```
dlogspline(q, fit)
plogspline(q, fit)
qlogspline(p, fit)
rlogspline(n, fit)
```

Arguments

<code>q</code>	vector of quantiles. Missing values (NAs) are allowed.
<code>p</code>	vector of probabilities. Missing values (NAs) are allowed.
<code>n</code>	sample size. If <code>length(n)</code> is larger than 1, then <code>length(n)</code> random values are returned.
<code>fit</code>	<code>logspline</code> object, typically the result of <code>logspline</code> .

Details

Elements of `q` or `p` that are missing will cause the corresponding elements of the result to be missing.

Value

Densities (`dlogspline`), probabilities (`plogspline`), quantiles (`qlogspline`), or a random sample (`rlogspline`) from a log spline density that was fitted using knot addition and deletion.

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg and Charles J. Stone. Log spline density estimation for censored data (1992). *Journal of Computational and Graphical Statistics*, **1**, 301–328.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[logspline](#), [plot.logspline](#), [summary.logspline](#), [oldlogspline](#).

Examples

```
x <- rnorm(100)
fit <- logspline(x)
qq <- qlogspline((1:99)/100, fit)
plot(qnorm((1:99)/100), qq) # qq plot of the fitted density
pp <- plogspline((-250:250)/100, fit)
plot((-250:250)/100, pp, type = "l")
lines((-250:250)/100, pnorm((-250:250)/100)) # asses the fit of the distribution
dd <- dlogspline((-250:250)/100, fit)
plot((-250:250)/100, dd, type = "l")
lines((-250:250)/100, dnorm((-250:250)/100)) # asses the fit of the density
rr <- rlogspline(100, fit) # random sample from fit
```

doldlogspline

Logspline Density Estimation - 1992 version

Description

Probability density function ([doldlogspline](#)), distribution function ([poldlogspline](#)), quantiles ([qoldlogspline](#)), and random samples ([roldlogspline](#)) from a logspline density that was fitted using the 1992 knot deletion algorithm ([oldlogspline](#)). The 1997 algorithm using knot deletion and addition is available using the [logspline](#) function.

Usage

```
doldlogspline(q, fit)
poldlogspline(q, fit)
qoldlogspline(p, fit)
roldlogspline(n, fit)
```

Arguments

q	vector of quantiles. Missing values (NAs) are allowed.
p	vector of probabilities. Missing values (NAs) are allowed.
n	sample size. If <code>length(n)</code> is larger than 1, then <code>length(n)</code> random values are returned.
fit	oldlogspline object, typically the result of oldlogspline .

Details

Elements of `q` or `p` that are missing will cause the corresponding elements of the result to be missing.

Value

Densities (`doldlogspline`), probabilities (`poldlogspline`), quantiles (`qoldlogspline`), or a random sample (`roldlogspline`) from an `oldlogspline` density that was fitted using knot deletion.

Author(s)

Charles Kooperberg [⟨clk@fhcrc.org⟩](mailto:clk@fhcrc.org).

References

Charles Kooperberg and Charles J. Stone. Log-spline density estimation for censored data (1992). *Journal of Computational and Graphical Statistics*, **1**, 301–328.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[logspline](#), [oldlogspline](#), [plot.oldlogspline](#), [summary.oldlogspline](#)

Examples

```
x <- rnorm(100)
fit <- oldlogspline(x)
qq <- qoldlogspline((1:99)/100, fit)
plot(qnorm((1:99)/100), qq) # qq plot of the fitted density
pp <- poldlogspline((-250:250)/100, fit)
plot((-250:250)/100, pp, type = "l")
lines((-250:250)/100, pnorm((-250:250)/100)) # asses the fit of the distribution
dd <- doldlogspline((-250:250)/100, fit)
plot((-250:250)/100, dd, type = "l")
lines((-250:250)/100, dnorm((-250:250)/100)) # asses the fit of the density
rr <- roldlogspline(100, fit) # random sample from fit
```

hare

Hare: hazard regression

Description

Fit a hazard regression model: linear splines are used to model the baseline hazard, covariates, and interactions. Fitted models can be, but do not need to be, proportional hazards models.

Usage

```
hare(data, delta, cov, penalty, maxdim, exclude, include, prophaz = FALSE,
      additive = FALSE, linear, fit, silent = TRUE)
```

Arguments

<code>data</code>	vector of observations. Observations may or may not be right censored. All observations should be nonnegative.
<code>delta</code>	binary vector with the same length as <code>data</code> . Elements of <code>data</code> for which the corresponding element of <code>delta</code> is 0 are assumed to be right censored, elements of <code>data</code> for which the corresponding element of <code>delta</code> is 1 are assumed to be uncensored. If <code>delta</code> is missing, all observations are assumed to be uncensored.
<code>cov</code>	covariates: matrix with as many rows as the length of <code>data</code> . May be omitted if there are no covariates. (If there are no covariates, however, <code>heft</code> will provide a more flexible model using cubic splines.)
<code>penalty</code>	the parameter to be used in the AIC criterion. The method chooses the number of knots that minimizes $-2 * \text{loglikelihood} + \text{penalty} * (\text{dimension})$. The default is to use <code>penalty = log(samplesize)</code> as in BIC. The effect of this parameter is summarized in <code>summary.hare</code> .
<code>maxdim</code>	maximum dimension (default is $6 * \text{length}(\text{data})^0.2$).
<code>exclude</code>	combinations to be excluded - this should be a matrix with 2 columns - if for example <code>exclude[1, 1] = 2</code> and <code>exclude[1, 2] = 3</code> no interaction between covariate 2 and 3 is included. 0 represents time.
<code>include</code>	those combinations that can be included. Should have the same format as <code>exclude</code> . Only one of <code>exclude</code> and <code>include</code> can be specified.
<code>prophaz</code>	should the model selection be restricted to proportional hazards models?
<code>additive</code>	should the model selection be restricted to additive models?
<code>linear</code>	vector indicating for which of the variables no knots should be entered. For example, if <code>linear = c(2, 3)</code> no knots for either covariate 2 or 3 are entered. 0 represents time. The default is none.
<code>fit</code>	<code>hare</code> object. If <code>fit</code> is specified, <code>hare</code> adds basis functions starting with those in <code>fit</code> .
<code>silent</code>	suppresses the printing of diagnostic output about basis functions added or deleted, Rao-statistics, Wald-statistics and log-likelihoods.

Value

An object of class `hare`, which is organized to serve as input for `plot.hare`, `summary.hare`, `dhare` (conditional density), `hhare` (conditional hazard rate), `phare` (conditional probabilities), `qhare` (conditional quantiles), and `rhare` (random numbers). The object is a list with the following members:

<code>ncov</code>	number of covariates.
<code>ndim</code>	number of dimensions of the fitted model.
<code>fcts</code>	matrix of size <code>ndim</code> × 6. each row is a basis function. First element: first covariate involved (0 means time); second element: which knot (0 means: constant (time) or linear (covariate)); third element: second covariate involved (NA means: this is a function of one variable);

	fourth element: knot involved (if the third element is NA, of no relevance); fifth element: beta; sixth element: standard error of beta.
knots	a matrix with <code>ncov</code> rows. Covariate <code>i</code> has row <code>i+1</code> , time has row 1. First column: number of knots in this dimension; other columns: the knots, appended with NAs to make it a matrix.
penalty	the parameter used in the AIC criterion.
max	maximum element of survival data.
ranges	column <code>i</code> gives the range of the <code>i</code> -th covariate.
logl	matrix with two columns. The <code>i</code> -th element of the first column is the loglikelihood of the model of dimension <code>i</code> . The second column indicates whether this model was fitted during the addition stage (1) or during the deletion stage (0).
sample	sample size.

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg, Charles J. Stone and Young K. Truong (1995). Hazard regression. *Journal of the American Statistical Association*, **90**, 78-94.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[heft](#), [plot.hare](#), [summary.hare](#), [dhare](#), [hhare](#), [phare](#), [qhare](#), [rhare](#).

Examples

```
fit <- hare(testhare[,1], testhare[,2], testhare[,3:8])
```

heft

Heft: hazard estimation with flexible tails

Description

Hazard estimation using cubic splines to approximate the log-hazard function and special functions to allow non-polynomial shapes in both tails.

Usage

```
heft(data, delta, penalty, knots, leftlin, shift, leftlog,
      rightlog, maxknots, mindist, silent = TRUE)
```

Arguments

<code>data</code>	vector of observations. Observations may or may not be right censored. All observations should be nonnegative.
<code>delta</code>	binary vector with the same length as <code>data</code> . Elements of <code>data</code> for which the corresponding element of <code>delta</code> is 0 are assumed to be right censored, elements of <code>data</code> for which the corresponding element of <code>delta</code> is 1 are assumed to be uncensored. If <code>delta</code> is missing, all observations are assumed to be uncensored.
<code>penalty</code>	the parameter to be used in the AIC criterion. The method chooses the number of knots that minimizes $-2 * \text{loglikelihood} + \text{penalty} * (\text{dimension})$. The default is to use <code>penalty = log(samplesize)</code> as in BIC. The effect of this parameter is summarized in summary.heft .
<code>knots</code>	ordered vector of values, which forces the method to start with these knots. If <code>knots</code> is not specified, a default knot-placement rule is employed.
<code>leftlin</code>	if <code>leftlin</code> is TRUE an extra basis-function, which is linear to the left of the first knot, is included in the basis. If any of <code>data</code> is exactly 0, the default of <code>leftlin</code> is TRUE, otherwise it is FALSE.
<code>shift</code>	parameter for the log terms. Default is <code>quantile(data[delta == 1], .75)</code> .
<code>leftlog</code>	coefficient of $\log \frac{x}{x+\text{shift}}$, which must be greater than -1. (In particular, if <code>leftlog</code> equals zero no $\log \frac{x}{x+\text{shift}}$ term is included.) If <code>leftlog</code> is missing its maximum likelihood estimate is used. If any of <code>data</code> is exactly zero, <code>leftlog</code> is set to zero.
<code>rightlog</code>	coefficient of $\log(x + \text{shift})$, which must be greater than -1. (In particular, if <code>rightlog</code> equals zero no $\log(x + \text{shift})$ term is included.) If <code>rightlog</code> is missing its maximum likelihood estimate is used.
<code>maxknots</code>	maximum number of knots allowed in the model (default is $4 * n^{0.2}$), where n is the length of <code>data</code> .
<code>mindist</code>	minimum distance in order statistics between knots. The default is 5.
<code>silent</code>	suppresses the printing of diagnostic output about knots added or deleted, Rao-statistics, Wald-statistics and log-likelihoods.

Value

An object of class `heft`, which is organized to serve as input for [plot.heft](#), [summary.heft](#), [dheft](#) (density), [hheft](#) (hazard rate), [pheft](#) (probabilities), [qheft](#) (quantiles), and [rheft](#) (random numbers). The object is a list with the following members:

<code>knots</code>	vector of the locations of the knots in the <code>heft</code> model.
<code>logl</code>	the k -th element is the log-likelihood of the fit with k knots.
<code>thetak</code>	coefficients of the knot part of the spline. The k -th coefficient is the coefficient of $(x - t(k))_+^3$. If a coefficient is zero the corresponding knot was considered and then deleted from the model.
<code>thetap</code>	coefficients of the polynomial part of the spline. The first element is the constant term and the second element is the linear term.

thetal	coefficients of the logarithmic terms. The first element equals <code>leftlog</code> and the second element equals <code>rightlog</code> .
penalty	the penalty that was used.
shift	parameter used in the definition of the log terms.
sample	the sample size.
logse	the standard errors of <code>thetal</code> .
max	the largest element of data.
ad	vector indicating whether a model of this dimension was not fit (2), fit during the addition stage (0) or during the deletion stage (1).

Author(s)

Charles Kooperberg [⟨clk@fhcrc.org⟩](mailto:clk@fhcrc.org).

References

Charles Kooperberg, Charles J. Stone and Young K. Truong (1995). Hazard regression. *Journal of the American Statistical Association*, **90**, 78-94.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[hare](#), [plot.heft](#), [summary.heft](#), [dheft](#), [hheft](#), [pheft](#), [qheft](#), [rheft](#).

Examples

```
fit1 <- heft(testhare[,1], testhare[,2])
# modify tail behavior
fit2 <- heft(testhare[,1], testhare[,2], leftlog = FALSE, rightlog = FALSE,
            leftlin = TRUE)
fit3 <- heft(testhare[,1], testhare[,2], penalty = 0) # select largest model
```

logspline

Logspline Density Estimation

Description

Fits a logspline density using splines to approximate the log-density using the 1997 knot addition and deletion algorithm ([logspline](#)). The 1992 algorithm is available using the [oldlogspline](#) function.

Usage

```
logspline(x, lbound, ubound, maxknots = 0, knots, nknots = 0, penalty,
          silent = TRUE, mind = -1, error.action = 2)
```

Arguments

<code>x</code>	data vector. The data needs to be uncensored. <code>oldlogspline</code> can deal with right- left- and interval-censored data.
<code>lbound, ubound</code>	lower/upper bound for the support of the density. For example, if there is a priori knowledge that the density equals zero to the left of 0, and has a discontinuity at 0, the user could specify <code>lbound = 0</code> . However, if the density is essentially zero near 0, one does not need to specify <code>lbound</code> .
<code>maxknots</code>	the maximum number of knots. The routine stops adding knots when this number of knots is reached. The method has an automatic rule for selecting <code>maxknots</code> if this parameter is not specified.
<code>knots</code>	ordered vector of values (that should cover the complete range of the observations), which forces the method to start with these knots. Overrides <code>knots</code> . If <code>knots</code> is not specified, a default knot-placement rule is employed.
<code>nknots</code>	forces the method to start with <code>nknots</code> knots. The method has an automatic rule for selecting <code>nknots</code> if this parameter is not specified.
<code>penalty</code>	the parameter to be used in the AIC criterion. The method chooses the number of knots that minimizes $-2 * \text{loglikelihood} + \text{penalty} * (\text{number of knots} - 1)$. The default is to use a penalty parameter of <code>penalty = log(samplesize)</code> as in BIC. The effect of this parameter is summarized in summary.logspline .
<code>silent</code>	should diagnostic output be printed?
<code>mind</code>	minimum distance, in order statistics, between knots.
<code>error.action</code>	how should <code>logspline</code> deal with non-convergence problems? Very-very rarely in some extreme situations <code>logspline</code> has convergence problems. The only two situations that I am aware of are when there is effectively a sharp bound, but this bound was not specified, or when the data is severely rounded. <code>logspline</code> can deal with this in three ways. If <code>error.action</code> is 2, the same data is rerun with the slightly more stable, but less flexible <code>oldlogspline</code> . The object is translated in a <code>logspline</code> object using <code>oldlogspline.to.logspline</code> , so this is almost invisible to the user. It is particularly useful when you run simulation studies, as the code can seamlessly continue. Only the <code>lbound</code> and <code>ubound</code> options are passed on to <code>oldlogspline</code> , other options revert to the default. If <code>error.action</code> is 1, a warning is printed, and <code>logspline</code> returns nothing (but does not crash). This is useful if you run a simulation, but do not like to revert to <code>oldlogspline</code> . If <code>error.action</code> is 0, the code crashes using the <code>stop</code> function.

Value

Object of the class `logspline`, that is intended as input for [plot.logspline](#) (summary plots), [summary.logspline](#) (fitting summary), [dlogspline](#) (densities), [plogspline](#) (probabilities), [qlogspline](#) (quantiles), [rlogspline](#) (random numbers from the fitted distribution).

The object has the following members:

<code>call</code>	the command that was executed.
-------------------	--------------------------------

<code>nknots</code>	the number of knots in the model that was selected.
<code>coef.pol</code>	coefficients of the polynomial part of the spline. The first coefficient is the constant term and the second is the linear term.
<code>coef.kts</code>	coefficients of the knots part of the spline. The k -th element is the coefficient of $(x - t(k))_+^3$ (where x_+^3 means the positive part of the third power of x , and $t(k)$ means knot k).
<code>knots</code>	vector of the locations of the knots in the <code>logspline</code> model.
<code>maxknots</code>	the largest number of knots minus one considered during fitting (i.e. with <code>maxknots = 6</code> the maximum number of knots is 5).
<code>penalty</code>	the penalty that was used.
<code>bound</code>	first element: 0 - <code>lbound</code> was $-\infty$ 1 it was something else; second element: <code>lbound</code> , if specified; third element: 0 - <code>ubound</code> was ∞ 1 it was something else; fourth element: <code>ubound</code> , if specified.
<code>samples</code>	the sample size.
<code>logl</code>	matrix with 3 columns. Column one: number of knots; column two: model fitted during addition (1) or deletion (2); column 3: log-likelihood.
<code>range</code>	range of the input data.
<code>mind</code>	minimum distance in order statistics between knots required during fitting (the actual minimum distance may be much larger).

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg and Charles J. Stone. Log-spline density estimation for censored data (1992). *Journal of Computational and Graphical Statistics*, **1**, 301–328.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[plot.logspline](#), [summary.logspline](#), [dlogspline](#), [plogspline](#), [qlogspline](#), [rlogspline](#), [oldlogspline](#), [oldlogspline.to.logspline](#).

Examples

```
y <- rnorm(100)
fit <- logspline(y)
plot(fit)
#
# as (4 == length(-2, -1, 0, 1, 2) - 1), this forces these initial knots,
# and does no knot selection
fit <- logspline(y, knots = c(-2, -1, 0, 1, 2), maxknots = 4, penalty = 0)
#
```

```

# the following example give one of the rare examples where logspline
# crashes, and this shows the use of error.action = 2.
#
set.seed(118)
zz <- rnorm(300)
zz[151:300] <- zz[151:300]+5
zz <- round(zz)
fit <- logspline(zz)
#
# you could rerun this with
# fit <- logspline(zz, error.action=0)
# or
# fit <- logspline(zz, error.action=1)

```

lspec

Lspec: logspline estimation of a spectral distribution

Description

Fit an `lspec` model to a time-series or a periodogram.

Usage

```
lspec(data, period, penalty, minmass, knots, maxknots, atoms, maxatoms,
maxdim, odd = FALSE, updown = 3, silent = TRUE)
```

Arguments

<code>data</code>	time series (exactly one of <code>data</code> and <code>period</code> should be specified). If <code>data</code> is specified, <code>lspec</code> first computes the modulus of the fast Fourier transform of the series using the function <code>fft</code> , resulting in a periodogram of length <code>floor(length(data)/2)</code> .
<code>period</code>	value of the periodogram for a time series at frequencies $\frac{2\pi j}{T}$, for $1 \leq j \leq T/2$. If <code>period</code> is specified, <code>odd</code> should indicate whether the length of the series <code>T</code> is odd (<code>odd = TRUE</code>) or even (<code>odd = FALSE</code>). Exactly one of <code>data</code> and <code>period</code> should be specified.
<code>penalty</code>	the parameter to be used in the AIC criterion. The method chooses the number of basis functions that minimizes $-2 * \text{loglikelihood} + \text{penalty} * (\text{number of basis functions})$. Default is to use a penalty parameter of <code>penalty = log(length(period))</code> as in BIC.
<code>minmass</code>	threshold value for atoms. No atoms having smaller mass than <code>minmass</code> are included in the model. If <code>minmass</code> takes its default value, in 95% of the samples, when data is Gaussian white noise, the model will not contain atoms.
<code>knots</code>	ordered vector of values, which forces the method to start with these knots. If <code>knots</code> is not specified, the program starts with one knot at zero and then employs stepwise addition of knots and atoms.

<code>maxknots</code>	maximum number of knots allowed in the model. Does not need to be specified, since the program has a default for <code>maxdim</code> and the number of dimensions equals the number of knots plus the number of atoms. If <code>maxknots = 1</code> the fitted spectral density function is constant.
<code>atoms</code>	ordered vector of values, which forces the method to start with discrete components at these frequencies. The values of <code>atoms</code> are rounded to the nearest multiple of $\frac{2\pi}{T}$. If <code>atoms</code> is not specified, the program starts with no atoms and then performs stepwise addition of knots and atoms.
<code>maxatoms</code>	maximum number of discrete components allowed in the model. Does not need to be specified, since the program has a default for <code>maxdim</code> and the number of dimensions equals the number of knots plus the number of atoms. If <code>maxatoms = 0</code> a continuous spectral distribution is fit.
<code>maxdim</code>	maximum number of basis functions allowed in the model (default is $\max(15, 4 \times \text{length}(\text{period})^{0.2})$).
<code>odd</code>	see <code>period</code> . If <code>period</code> is not specified, <code>odd</code> is not relevant.
<code>updown</code>	the maximal number of times that <code>lspec</code> should go through a cycle of stepwise addition and stepwise deletion until a stable solution is reached.
<code>silent</code>	should printing of information be suppressed?

Value

Object of class `lspec`. The output is organized to serve as input for `plot.lspec` (summary plots), `summary.lspec` (summarizes fitting), `clspec` (for autocorrelations and autocovariances), `dlspec` (for spectral density and line-spectrum,) `plspec` (for the spectral distribution), and `rlspec` (for random time series with the same spectrum).

<code>call</code>	the command that was executed.
<code>thetap</code>	coefficients of the polynomial part of the spline.
<code>nknots</code>	the number of knots that were retained.
<code>knots</code>	vector of the locations of the knots in the logspline model. Only the knots that were retained are in this vector.
<code>thetak</code>	coefficients of the knot part of the spline. The k -th coefficient is the coefficient of $(x - t(k))_+^3$.
<code>natoms</code>	the number of atoms that were retained.
<code>atoms</code>	vector of the locations of the atoms in the model. Only the atoms that were retained are in this vector.
<code>mass</code>	The k -th coefficient is the mass at <code>atom[k]</code> .
<code>logl</code>	the log-likelihood of the model.
<code>penalty</code>	the penalty that was used.
<code>minmass</code>	the minimum mass for an atom that was allowed.
<code>sample</code>	the sample size that was used, either computed as <code>length(data)</code> or as $(2 * \text{length}(\text{period}))$ when <code>odd = FALSE</code> or as $(2 * \text{length}(\text{period}) + 1)$ when <code>odd = TRUE</code> .

updown the actual number of times that `lspec` went through a cycle of stepwise addition and stepwise deletion until a stable solution was reached, or minus the number of times that `lspec` went through a cycle of stepwise addition and stepwise deletion until it decided to quit.

Author(s)

Charles Kooperberg [⟨clk@fhcrc.org⟩](mailto:clk@fhcrc.org).

References

Charles Kooperberg, Charles J. Stone, and Young K. Truong (1995). Log spline Estimation of a Possibly Mixed Spectral Distribution. *Journal of Time Series Analysis*, **16**, 359-388.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[plot.lspec](#), [summary.lspec](#), [clspec](#), [dlspec](#), [plspec](#), [rlspec](#).

Examples

```
data(co2)
co2.detrend <- unstrip(lm(co2~c(1:length(co2)))$residuals)
fit <- lspec(co2.detrend)
```

oldlogspline

Log spline Density Estimation - 1992 version

Description

Fits a `logspline` density using splines to approximate the log-density using the 1992 knot deletion algorithm ([oldlogspline](#)). The 1997 algorithm using knot deletion and addition is available using the [logspline](#) function.

Usage

```
oldlogspline(uncensored, right, left, interval, lbound,
             ubound, nknots, knots, penalty, delete = TRUE)
```

Arguments

`uncensored` vector of uncensored observations from the distribution whose density is to be estimated. If there are no uncensored observations, this argument can be omitted. However, either `uncensored` or `interval` must be specified.

<code>right</code>	vector of right censored observations from the distribution whose density is to be estimated. If there are no right censored observations, this argument can be omitted.
<code>left</code>	vector of left censored observations from the distribution whose density is to be estimated. If there are no left censored observations, this argument can be omitted.
<code>interval</code>	two column matrix of lower and upper bounds of observations that are interval censored from the distribution whose density is to be estimated. If there are no interval censored observations, this argument can be omitted.
<code>lbound, ubound</code>	lower/upper bound for the support of the density. For example, if there is a priori knowledge that the density equals zero to the left of 0, and has a discontinuity at 0, the user could specify <code>lbound = 0</code> . However, if the density is essentially zero near 0, one does not need to specify <code>lbound</code> . The default for <code>lbound</code> is <code>-inf</code> and the default for <code>ubound</code> is <code>inf</code> .
<code>nknots</code>	forces the method to start with <code>nknots</code> knots (<code>delete = TRUE</code>) or to fit a density with <code>nknots</code> knots (<code>delete = FALSE</code>). The method has an automatic rule for selecting <code>nknots</code> if this parameter is not specified.
<code>knots</code>	ordered vector of values (that should cover the complete range of the observations), which forces the method to start with these knots (<code>delete = TRUE</code>) or to fit a density with these knots <code>delete = FALSE</code>). Overrides <code>nknots</code> . If <code>knots</code> is not specified, a default knot-placement rule is employed.
<code>penalty</code>	the parameter to be used in the AIC criterion. The method chooses the number of knots that minimizes $-2 * \text{loglikelihood} + \text{penalty} * (\text{number of knots} - 1)$. The default is to use a penalty parameter of <code>penalty = log(samplesize)</code> as in BIC. The effect of this parameter is summarized in summary.oldlogspline .
<code>delete</code>	should stepwise knot deletion be employed?

Value

Object of the class `oldlogspline`, that is intended as input for [plot.oldlogspline](#), [summary.oldlogspline](#), [doldlogspline](#) (densities), [poldlogspline](#) (probabilities), [qoldlogspline](#) (quantiles), [roidlogspline](#) (random numbers from the fitted distribution). The function [oldlogspline.to.logspline](#) can translate an object of the class `oldlogspline` to an object of the class `logspline`.

The object has the following members:

<code>call</code>	the command that was executed.
<code>knots</code>	vector of the locations of the knots in the <code>oldlogspline</code> model. <code>old</code>
<code>coef</code>	coefficients of the spline. The first coefficient is the constant term, the second is the linear term and the k -th ($k > 2$) is the coefficient of $(x - t(k - 2))_+^3$ (where x_+^3 means the positive part of the third power of x , and $t(k - 2)$ means knot $k - 2$). If a coefficient is zero the corresponding knot was deleted from the model.

bound	first element: 0 - lbound was $-\infty$, 1 it was something else; second element: lbound, if specified; third element: 0 - ubound was ∞ , 1 it was something else; fourth element: ubound, if specified.
logl	the k-th element is the log-likelihood of the fit with k+2 knots.
penalty	the penalty that was used.
sample	the sample size that was used.
delete	was stepwise knot deletion employed?

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg and Charles J. Stone. Log spline density estimation for censored data (1992). *Journal of Computational and Graphical Statistics*, **1**, 301–328.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[logspline](#), [oldlogspline](#), [plot.oldlogspline](#), [summary.oldlogspline](#), [doldlogspline](#), [poldlogspline](#), [qoldlogspline](#), [roidlogspline](#), [oldlogspline.to.logspline](#).

Examples

```
# A simple example
y <- rnorm(100)
fit <- oldlogspline(y)
plot(fit)
# An example involving censoring and a lower bound
y <- rlnorm(1000)
censoring <- rexp(1000) * 4
delta <- 1 * (y <= censoring)
y[delta == 0] <- censoring[delta == 0]
fit <- oldlogspline(y[delta == 1], y[delta == 0], lbound = 0)
```

oldlogspline.to.logspline

Log spline Density Estimation - 1992 to 1997 version

Description

Translates an `oldlogspline` object in an `logspline` object. This routine is mostly used in `logspline`, as it allows the routine to use `oldlogspline` for some situations where `logspline` crashes. The other use is when you have censored data, and thus have to use `oldlogspline` to fit, but wish to use the auxiliary routines from `logspline`.

Usage

```
oldlogspline.to.logspline(obj, data)
```

Arguments

obj	object of class <code>logspline</code>
data	the original data. Used to compute the <code>range</code> component of the new object. if <code>data</code> is not available, the $1/(n+1)$ and $n/(n+1)$ quantiles of the fitted distribution are used for <code>range</code> .

Value

object of the class `logspline`. The `call` component of the new object is not useful. The `delete` component of the old object is ignored.

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg and Charles J. Stone. Log-spline density estimation for censored data (1992). *Journal of Computational and Graphical Statistics*, **1**, 301–328.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[logspline](#), [oldlogspline](#).

Examples

```
x <- rnorm(100)
fit.old <- oldlogspline(x)
fit.translate <- oldlogspline.to.logspline(fit.old, x)
fit.new <- logspline(x)
plot(fit.new)
plot(fit.old, add=TRUE, col=2)
#
# should look almost the same, the differences are the
# different fitting routines
#
```

persp.polymars *Polymars: multivariate adaptive polynomial spline regression*

Description

This function is not intended for direct use. It is called by `plot.polymars`.

Usage

```
## S3 method for class 'polymars':
persp(x, predictor1, predictor2, response, n = 33,
      xlim, ylim, xx, contour.polymars, main, intercept, ...)
```

Arguments

`x, predictor1, predictor2`
this function is not intended to be called directly.

`response, n, xlim, ylim`
this function is not intended to be called directly.

`xx, contour.polymars`
this function is not intended to be called directly.

`main, intercept, ...`
this function is not intended to be called directly.

Details

This function produces a 3-d contour or perspective plot. It is intended to be called by `plot.polymars`.

Author(s)

Martin O'Connor.

References

Charles Kooperberg, Smarajit Bose, and Charles J. Stone (1997). Polychotomous regression. *Journal of the American Statistical Association*, **92**, 117–127.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

`polymars`, `plot.polymars`.

plot.hare

Hare: hazard regression

Description

Plots a density, distribution function, hazard function or survival function for a `hare` object.

Usage

```
## S3 method for class 'hare':
plot(x, cov, n = 100, which = 0, what = "d", time, add = FALSE, xlim,
     xlab, ylab, type, ...)
```

Arguments

<code>x</code>	<code>hare</code> object, typically the result of <code>hare</code> .
<code>cov</code>	a vector of length <code>fit\$ncov</code> , indicating for which combination of covariates the plot should be made. Can be omitted only if <code>fit\$ncov</code> is 0.
<code>n</code>	the number of equally spaced points at which to plot the function.
<code>which</code>	for which coordinate should the plot be made. 0: time; positive value <code>i</code> : covariate <code>i</code> . Note that if <code>which</code> is the positive value <code>i</code> , then the element corresponding to this covariate must be given in <code>cov</code> even though its actual value is irrelevant.
<code>what</code>	what should be plotted: "d" (density), "p" (distribution function), "s" (survival function) or "h" (hazard function).
<code>time</code>	if <code>which</code> is not equal to 0, the value of time for which the plot should be made.
<code>add</code>	should the plot be added to an existing plot?
<code>xlim</code>	plotting limits; default is from the maximum of 0 and 10% before the 1st percentile to the minimum of 10% further than the 99th percentile and the largest observation.
<code>xlab, ylab</code>	labels for the axes. Per default no labels are printed.
<code>type</code>	plotting type. The default is lines.
<code>...</code>	all other plotting options are passed on.

Details

This function produces a plot of a `hare` fit at `n` equally spaced points roughly covering the support of the density. (Use `xlim=c(from, to)` to change the range of these points.)

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg, Charles J. Stone and Young K. Truong (1995). Hazard regression. *Journal of the American Statistical Association*, **90**, 78-94.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[hare](#), [summary.hare](#), [dhare](#), [hhare](#), [phare](#), [qhare](#), [rhare](#).

Examples

```
fit <- hare(testhare[,1], testhare[,2], testhare[,3:8])
# hazard curve for covariates like case 1
plot(fit, testhare[1,3:8], what = "h")
# survival function as a function of covariate 2, for covariates as case 1 at t=3
plot(fit, testhare[1,3:8], which = 2, what = "s", time = 3)
```

plot.heft

Heft: hazard estimation with flexible tails

Description

Plots a density, distribution function, hazard function or survival function for a `heft` object.

Usage

```
## S3 method for class 'heft':
plot(x, n = 100, what = "d", add = FALSE, xlim, xlab, ylab,
type, ...)
```

Arguments

<code>x</code>	<code>heft</code> object, typically the result of heft .
<code>n</code>	the number of equally spaced points at which to plot the function.
<code>what</code>	what should be plotted: "d" (density), "p" (distribution function), "s" (survival function) or "h" (hazard function).
<code>add</code>	should the plot be added to an existing plot?
<code>xlim</code>	plotting limits; default is from the maximum of 0 and 10% before the 1st percentile to the minimum of 10% further than the 99th percentile and the largest observation.
<code>xlab, ylab</code>	labels for the axes. The default is no labels.
<code>type</code>	plotting type. The default is lines.
<code>...</code>	all other plotting options are passed on.

Details

This function produces a plot of a [heft](#) fit at n equally spaced points roughly covering the support of the density. (Use `xlim=c(from,to)` to change the range of these points.)

Author(s)

Charles Kooperberg [⟨clk@fhcrc.org⟩](mailto:clk@fhcrc.org).

References

Charles Kooperberg, Charles J. Stone and Young K. Truong (1995). Hazard regression. *Journal of the American Statistical Association*, **90**, 78-94.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[heft](#), [summary.heft](#), [dheft](#), [hheft](#), [pheft](#), [qheft](#), [rheft](#).

Examples

```
fit1 <- heft(testhare[,1], testhare[,2])
plot(fit1, what = "h")
# modify tail behavior
fit2 <- heft(testhare[,1], testhare[,2], leftlog = FALSE, rightlog = FALSE,
  leftlin = TRUE)
plot(fit2, what = "h", add = TRUE, lty = 2)
fit3 <- heft(testhare[,1], testhare[,2], penalty = 0) # select largest model
plot(fit3, what = "h", add = TRUE, lty = 3)
```

plot.logspline *Logspline Density Estimation*

Description

Plots a `logspline` density, distribution function, hazard function or survival function from a `logspline` density that was fitted using the 1997 knot addition and deletion algorithm ([logspline](#)). The 1992 algorithm is available using the [oldlogspline](#) function.

Usage

```
## S3 method for class 'logspline':
plot(x, n = 100, what = "d", add = FALSE, xlim, xlab = "",
  ylab = "", type = "l", ...)
```

Arguments

<code>x</code>	logspline object, typically the result of <code>logspline</code> .
<code>n</code>	the number of equally spaced points at which to plot the density.
<code>what</code>	what should be plotted: "d" (density), "p" (distribution function), "s" (survival function) or "h" (hazard function).
<code>add</code>	should the plot be added to an existing plot.
<code>xlim</code>	range of data on which to plot. Default is from the 1th to the 99th percentile of the density, extended by 10% on each end.
<code>xlab, ylab</code>	labels plotted on the axes.
<code>type</code>	type of plot.
<code>...</code>	other plotting options, as desired

Details

This function produces a plot of a `logspline` fit at `n` equally spaced points roughly covering the support of the density. (Use `xlim = c(from, to)` to change the range of these points.)

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg and Charles J. Stone. Log-spline density estimation for censored data (1992). *Journal of Computational and Graphical Statistics*, **1**, 301–328.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[logspline](#), [summary.logspline](#), [dlogspline](#), [plogspline](#), [qlogspline](#), [rlogspline](#), [oldlogspline](#).

Examples

```
y <- rnorm(100)
fit <- logspline(y)
plot(fit)
```

plot.lspec

Lspec: logspline estimation of a spectral distribution

Description

Plots a spectral density function, line spectrum, or spectral distribution from a model fitted with `lspec`

Usage

```
## S3 method for class 'lspec':
plot(x, what = "b", n, add = FALSE, xlim, ylim, xlab = "", ylab = "",
     type, ...)
```

Arguments

<code>x</code>	<code>lspec</code> object, typically the result of <code>lspec</code> .
<code>what</code>	what should be plotted: <code>b</code> (spectral density and line spectrum superimposed), <code>d</code> (spectral density function), <code>l</code> (line spectrum) or <code>p</code> (spectral distribution function).
<code>n</code>	the number of equally spaced points at which to plot the fit; default is <code>max(100, fit\$sample)</code> .
<code>add</code>	indicate that the plot should be added to an existing plot.
<code>xlim</code>	X-axis plotting limits: default is $c(0, \pi)$, except when <code>what = "p"</code> , when the default is $c(-\pi, \pi)$.
<code>ylim</code>	Y-axis plotting limits.
<code>xlab, ylab</code>	axis labels.
<code>type</code>	plotting type; default is <code>"l"</code> when <code>what = "d"</code> and <code>what = "p"</code> , <code>"h"</code> when <code>what = "l"</code> , and a combination of <code>"h"</code> and <code>"l"</code> when <code>what = "b"</code>
<code>...</code>	all regular plotting options are passed on.

Note

If `what = "p"` the plotting range cannot extend beyond the interval $[-\pi, \pi]$.

Author(s)

Charles Kooperberg (`clk@fhcrc.org`).

References

Charles Kooperberg, Charles J. Stone, and Young K. Truong (1995). Logspline Estimation of a Possibly Mixed Spectral Distribution. *Journal of Time Series Analysis*, **16**, 359-388.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371-1470.

See Also

[lspec](#), [summary.lspec](#), [clspec](#), [dlspec](#), [plspec](#), [rlspec](#).

Examples

```
data(co2)
co2.detrend <- lm(co2~c(1:length(co2)))$residuals
fit <- lspect(co2.detrend)
plot(fit)
```

plot.oldlogspline *Log spline Density Estimation - 1992 version*

Description

Plots an `oldlogspline` density, distribution function, hazard function or survival function from a logspline density that was fitted using the 1992 knot deletion algorithm. The 1997 algorithm using knot deletion and addition is available using the [logspline](#) function.

Usage

```
## S3 method for class 'oldlogspline':
plot(x, n = 100, what = "d", xlim, xlab = "", ylab = "",
type = "l", ...)
```

Arguments

<code>x</code>	logspline object, typically the result of logspline .
<code>n</code>	the number of equally spaced points at which to plot the density.
<code>what</code>	what should be plotted: "d" (density), "p" (distribution function), "s" (survival function) or "h" (hazard function).
<code>xlim</code>	range of data on which to plot. Default is from the 1th to the 99th percentile of the density, extended by 10% on each end.
<code>xlab, ylab</code>	labels plotted on the axes.
<code>type</code>	type of plot.
<code>...</code>	other plotting options, as desired

Details

This function produces a plot of a `oldlogspline` fit at `n` equally spaced points roughly covering the support of the density. (Use `xlim=c(from, to)` to change the range of these points.)

Author(s)

Charles Kooperberg (clk@fhcrc.org).

References

Charles Kooperberg and Charles J. Stone. Log spline density estimation for censored data (1992). *Journal of Computational and Graphical Statistics*, **1**, 301–328.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[logspline](#), [oldlogspline](#), [summary.oldlogspline](#), [doldlogspline](#), [poldlogspline](#), [qoldlogspline](#), [roidlogspline](#).

Examples

```
y <- rnorm(100)
fit <- oldlogspline(y)
plot(fit)
```

plot.polyclass *Polyclass: polychotomous regression and multiple classification*

Description

Probability or classification plots for a [polyclass](#) model.

Usage

```
## S3 method for class 'polyclass':
plot(x, cov, which, lims, what, data, n, xlab="", ylab="",
     zlab="", ...)
```

Arguments

x	polyclass object, typically the result of polyclass .
cov	a vector of length <code>fit\$ncov</code> , indicating for which combination of covariates the plot should be made. Can never be omitted. Should always have length <code>fit\$ncov</code> , even if some values are irrelevant.
which	for which covariates should the plot be made. Number or a character string defining the name, if the same names were used with the call to polyclass . Which should have length one if <code>what</code> is 6 or larger and length two if <code>what</code> is 5 or smaller.
lims	plotting limits. If omitted, the plot is made over the same range of the covariate as in the original data. Otherwise a vector of length two of the form <code>c(min, max)</code> if <code>what</code> is 6 or larger and a vector of length four of the form <code>c(xmin, xmax, ymin, ymax)</code> if <code>what</code> is 5 or smaller.
what	an integer between 1 and 8, defining the type of plot to be made.

1. Plots the probability of one class as a contour plot of two variables.
 2. Plots the probability of one class as a perspective plot of two variables.
 3. Plots the probability of one class as an image plot of two variables.
 4. Classifies the area as a contour plot of two variables.
 5. Classifies the area as an image plot of two variables.
 6. Classifies the line as a plot of one variable.
 7. Plots the probabilities of all classes as a function of one variable.
 8. Plots the probability of one class as a function of one variable.
- data Class for which the plot is made. Should be provided if `what` is 1, 2, 3 or 8.
- n the number of equally spaced points at which to plot the fit. The default is 250 if `what` is 6 or larger or 50 (which results in 2500 plotting points) if `what` is 5 or smaller.
- xlab, ylab, zlab axis plotting labels.
- ... all other options are passed on.

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg, Smarajit Bose, and Charles J. Stone (1997). Polychotomous regression. *Journal of the American Statistical Association*, **92**, 117–127.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[polyclass](#), [summary.polyclass](#), [beta.polyclass](#), [cpolyclass](#), [ppolyclass](#), [rpolyclass](#).

Examples

```
data(iris)
fit.iris <- polyclass(iris[,5], iris[,1:4])
plot(fit.iris, iris[64,1:4], which=c(3,4), data=2, what=1)
plot(fit.iris, iris[64,1:4], which=c(3,4), what=5)
plot(fit.iris, iris[64,1:4], which=4, what=7)
```

plot.polymars *Polymars: multivariate adaptive polynomial spline regression*

Description

Produces two and three dimensional plots of the fitted values from a `polymars` object.

Usage

```
## S3 method for class 'polymars':
plot(x, predictor1, response, predictor2, xx, add = FALSE, n,
     xyz = FALSE, contour.polymars = FALSE, xlim, ylim, intercept, ...)
```

Arguments

<code>x</code>	<code>polymars</code> object, typically the result of <code>polymars</code> .
<code>predictor1</code>	the index of a predictor that was used when the <code>polymars</code> model was fit. For the two dimensional plots, this variable is plotted along the X-axis.
<code>response</code>	if the model was fitted to multiple response data the response index should be specified.
<code>predictor2</code>	the index of a predictor that was used when the <code>polymars</code> model was fit. For the three dimensional plots, this variable is plotted along the Y-axis. See <code>xyz</code> .
<code>xx</code>	should be a vector of length equal to the number of predictors in the original data set. The values should be in the same order as in the original dataset. By default the function uses the median values of the data that was used to fit the model. Although the values for <code>predictor</code> and <code>predictor2</code> are not used, they should still be provided as part of <code>xx</code> .
<code>add</code>	should the plot be added to a previously created plot? Works only for two dimensional plots.
<code>n</code>	number of plotting points (2 dimensional plot) or plotting points along each axis (3 dimensional plot). The default is <code>n = 100</code> for 2 dimensional plots and <code>n = 33</code> for 3 dimensional plots.
<code>xyz</code>	is the plot being made a 3 dimensional plot? If there is only one response it need not be set, if two numerical values accompany the model in the call they will be understood as two predictors for a 3-d plot. By default a 3-d plot uses the <code>persp</code> function. Categorical predictors cannot be used for 3 dimensional plots.
<code>contour.polymars</code>	if the plot being made a 3 dimensional plot should it be made as a contour plot (<code>TRUE</code>) or a perspective plot (<code>FALSE</code>). function <code>contour</code> is being made.
<code>intercept</code>	Setting <code>intercept</code> equal to <code>FALSE</code> evaluates the object without intercept. The intercept may also be given any numerical value which overrides the fitted coefficient from the object. The default is <code>TRUE</code> .
<code>xlim, ylim</code>	Plotting limits. The function tries to choose intelligent limits itself
<code>...</code>	other options are passed on.

Details

This function produces a 2-d plot of 1 predictor and response of a `polymars` object at `n` equally spaced points or a 3-d plot of two predictors and response of a `polymars` object. The range of the plot is by default equal to the range of the particular predictor(s) in the original data, but this can be changed by `xlim = c(from, to)` and `ylim = c(from, to)`.

Author(s)

Martin O'Connor.

References

Charles Kooperberg, Smarajit Bose, and Charles J. Stone (1997). Polychotomous regression. *Journal of the American Statistical Association*, **92**, 117–127.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[design.polymars](#), [polymars](#), [predict.polymars](#), [summary.polymars](#).

Examples

```
data(state)
state.pm <- polymars(state.region, state.x77, knots = 15, classify = TRUE, gcv = 1)
plot(state.pm, 3, 4)
```

polyclass

Polyclass: polychotomous regression and multiple classification

Description

Fit a polychotomous regression and multiple classification using linear splines and selected tensor products.

Usage

```
polyclass(data, cov, weight, penalty, maxdim, exclude, include,
additive = FALSE, linear, delete = 2, fit, silent = TRUE,
normweight = TRUE, tdata, tcov, tweight, cv, select, loss, seed)
```

Arguments

<code>data</code>	vector of classes: data should ranges over consecutive integers with 0 or 1 as the minimum value.
<code>cov</code>	covariates: matrix with as many rows as the length of <code>data</code> .
<code>weight</code>	optional vector of case-weights. Should have the same length as <code>data</code> .
<code>penalty</code>	the parameter to be used in the AIC criterion if the model selection is carried out by AIC. The program chooses the number of knots that minimizes $-2 * \text{loglikelihood} + \text{penalty} * (\text{dimension})$. The default is to use $\text{penalty} = \log(\text{length}(\text{data}))$ as in BIC. If the model selection is carried out by cross-validation or using a test set, the program uses the number of knots that minimizes $\text{loss} + \text{penalty} * \text{dimension} * (\text{loss for smallest model})$. In this case the default of <code>penalty</code> is 0.
<code>maxdim</code>	maximum dimension (default is $\min(n, 4*n^{1/3}*(cl-1))$, where n is <code>length(data)</code> and cl the number of classes.
<code>exclude</code>	combinations to be excluded - this should be a matrix with 2 columns - if for example <code>exclude[1, 1] = 2</code> and <code>exclude[1, 2] = 3</code> no interaction between covariate 2 and 3 is included. 0 represents time.
<code>include</code>	those combinations that can be included. Should have the same format as <code>exclude</code> . Only one of <code>exclude</code> and <code>include</code> can be specified .
<code>additive</code>	should the model selection be restricted to additive models?
<code>linear</code>	vector indicating for which of the variables no knots should be entered. For example, if <code>linear = c(2, 3)</code> no knots for either covariate 2 or 3 are entered. 0 represents time.
<code>delete</code>	should complete basis functions be deleted at once (2), should only individual dimensions be deleted (1) or should only the addition stage of the model selection be carried out (0)?
<code>fit</code>	<code>polyclass</code> object. If <code>fit</code> is specified, <code>polyclass</code> adds basis functions starting with those in <code>fit</code> .
<code>silent</code>	suppresses the printing of diagnostic output about basis functions added or deleted, Rao-statistics, Wald-statistics and log-likelihoods.
<code>normweight</code>	should the weights be normalized so that they average to one? This option has only an effect if the model is selected using AIC.
<code>tdata, tcov, tweight</code>	test set. Should satisfy the same requirements as <code>data</code> , <code>cov</code> and <code>weight</code> . If all test set weights are one, <code>tweight</code> can be omitted. If <code>tdata</code> and <code>tcov</code> are specified, the model selection is carried out using this test set, irrespective of the input for <code>penalty</code> or <code>cv</code> .
<code>cv</code>	in how many subsets should the data be divided for cross-validation? If <code>cv</code> is specified and <code>tdata</code> is omitted, the model selection is carried out by cross-validation.
<code>select</code>	if a test set is provided, or if the model is selected using cross validation, should the model be select that minimizes (misclassification) loss (0), that maximizes test set log-likelihood (1) or that minimizes test set squared error loss (2)?

loss	a rectangular matrix specifying the loss function, whose size is the number of classes times number of actions. Used for cross-validation and test set model selection. <code>loss[i, j]</code> contains the loss for assigning action <code>j</code> to an object whose true class is <code>i</code> . The default is 1 minus the identity matrix. <code>loss</code> does not need to be square.
seed	optional seed for the random number generator that determines the sequence of the cases for cross-validation. If the seed has length 12 or more, the first twelve elements are assumed to be <code>.Random.seed</code> , otherwise the function <code>set.seed</code> is used. If <code>seed</code> is 0 or <code>rep(0, 12)</code> , it is assumed that the user has already provided a (random) ordering. If <code>seed</code> is not provided, while a fit with an element <code>fit\$seed</code> is provided, <code>.Random.seed</code> is set using <code>set.seed(fit\$seed)</code> . Otherwise the present value of <code>.Random.seed</code> is used.

Value

The output is an object of class `polyclass`, organized to serve as input for `plot.polyclass`, `beta.polyclass`, `summary.polyclass`, `ppolyclass` (fitted probabilities), `cpolyclass` (fitted classes) and `rpolyclass` (random classes). The function returns a list with the following members:

call	the command that was executed.
ncov	number of covariates.
ndim	number of dimensions of the fitted model.
nclass	number of classes.
nbas	number of basis functions.
naction	number of possible actions that are considered.
fcts	matrix of size <code>nbas × (nclass + 4)</code> . each row is a basis function. First element: first covariate involved (NA = constant); second element: which knot (NA means: constant or linear); third element: second covariate involved (NA means: this is a function of one variable); fourth element: knot involved (if the third element is NA, of no relevance); fifth, sixth,... element: beta (coefficient) for class one, two, ...
knots	a matrix with <code>ncov</code> rows. Covariate <code>i</code> has row <code>i+1</code> , time has row 1. First column: number of knots in this dimension; other columns: the knots, appended with NAs to make it a matrix.
cv	in how many sets was the data divided for cross-validation. Only provided if <code>method = 2</code> .
loss	the loss matrix used in cross-validation and test set. Only provided if <code>method = 1</code> or <code>method = 2</code> .
penalty	the parameter used in the AIC criterion. Only provided if <code>method = 0</code> .
method	0 = AIC, 1 = test set, 2 = cross-validation.
ranges	column <code>i</code> gives the range of the <code>i</code> -th covariate.

logl	matrix with eight or eleven columns. Summarizes fits. Column one indicates the dimension, column two the AIC or loss value, whichever was used during the model selection appropriate, column three four and five give the training set log-likelihood, (misclassification) loss and squared error loss, columns six to eight give the same information for the test set, column nine (or column six if <code>method = 0</code> or <code>method = 2</code>) indicates whether the model was fitted during the addition stage (1) or during the deletion stage (0), column ten and eleven (or seven and eight) the minimum and maximum penalty parameter for which AIC would have selected this model.
sample	sample size.
tsample	the sample size of the test set. Only provided if <code>method = 1</code> .
wgtsum	sum of the case weights.
covnames	names of the covariates.
classnames	(numerical) names of the classes.
cv.aic	the penalty value that was determined optimal by by cross validation. Only provided if <code>method = 2</code> .
cv.tab	table with three columns. Column one and two indicate the penalty parameter range for which the cv-loss in column three would be realized. Only provided if <code>method = 2</code> .
seed	the random seed that was used to determine the order of the cases for cross-validation. Only provided if <code>method = 2</code> .
delete	were complete basis functions deleted at once (2), were only individual dimensions deleted (1) or was only the addition stage of the model selection carried out (0)?
beta	moments of basisfunctions. Needed for <code>beta.polyclass</code> .
select	if a test set is provided, or if the model is selected using cross validation, was the model selected that minimized (misclassification) loss (0), that maximized test set log-likelihood (1) or that minimized test set squared error loss (2)?
anova	matrix with three columns. The first two elements in a line indicate the subspace to which the line refers. The third element indicates the percentage of variance explained by that subspace.
twgtsum	sum of the test set case weights (only if <code>method = 1</code>).

Author(s)

Charles Kooperberg [⟨clk@fhcrc.org⟩](mailto:clk@fhcrc.org).

References

- Charles Kooperberg, Smarajit Bose, and Charles J. Stone (1997). Polychotomous regression. *Journal of the American Statistical Association*, **92**, 117–127.
- Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[polymars](#), [plot.polyclass](#), [summary.polyclass](#), [beta.polyclass](#), [cpolyclass](#), [ppolyclass](#), [rpolyclass](#).

Examples

```
data(iris)
fit.iris <- polyclass(iris[,5], iris[,1:4])
```

polymars

Polymars: multivariate adaptive polynomial spline regression

Description

An adaptive regression procedure using piecewise linear splines to model the response.

Usage

```
polymars(responses, predictors, maxsize, gcv = 4, additive = FALSE,
startmodel, weights, no.interact, knots, knot.space = 3, ts.resp,
ts.pred, ts.weights, classify, factors, tolerance, verbose = FALSE)
```

Arguments

responses	vector of responses, or a matrix for multiple response regression. In the case of a matrix each column corresponds to a response and each row corresponds to an observation. Missing values are not allowed.
predictors	matrix of predictor variables for the regression. Each column corresponds to a predictor and each row corresponds to an observation in the same order as they appear in the response argument. Missing values are not allowed.
maxsize	the maximum number of basis functions that the model is allowed to grow to in the stepwise addition procedure. Default is $\min(6 * (n^{1/3}), n/4, 100)$, where n is the number of observations.
gcv	parameter used to find the overall best model from a sequence of fitted models. The residual sum of squares of a model is penalized by dividing by the square of $1 - (\text{gcv} \times \text{model size}) / \text{cases}$. A larger gcv value would tend to produce a smaller model.
additive	Should the fitted model be additive in the predictors?
startmodel	the first model that is to be fit by <code>polymars</code> . It is either an object of the class <code>polymars</code> or a model dreamed up by the user. In that case, it takes the form of a $4 \times n$ matrix, where n is the number of basis functions in the starting model excluding the intercept. Each row corresponds to one basis function (with two possible components). Column 1 is the index of the first predictor involved. Column 2 is a possible knot in this predictor. If column 2 is NA, the first component is linear. Column 3 is the possible second predictor involved (if column 3

is NA the basis function only depends on one predictor). Column 4 contains the possible knot for the predictor in column 3, and it is NA when this component is linear. Example: if a row reads 3 NA 2 4.7, the corresponding basis function is $[X_3 * (X_2 - 4.7)_+]$; if a row reads 2 4.3 NA NA the corresponding basis function is $[(X_2 - 4.3)_+]$. A fifth column can be added with 1s and 0s. The 1s specify which basis functions of the startmodel must be in each model. Thus, these functions stay in the model during the whole stepwise fitting procedure. If `startmodel` is not specified `polymars` starts with a model that only contains the intercept.

<code>weights</code>	optional vector of observation weights; if supplied, the algorithm fits to minimize the sum of the weights multiplied by the squared residuals. The length of weights must be the same as the number of observations. The weights must be nonnegative.
<code>no.interact</code>	an optional matrix used if certain predictor interactions are not allowed in the model. It is given as a matrix of size $2 \times m$, with predictor indices as entries. The two predictors of any row cannot have interaction terms with each other.
<code>knots</code>	defines how the function is to find potential knots for the spline basis functions. This can be set to the maximum number of knots you would like to be considered for each predictor. Usually, to avoid the design matrix becoming singular the actual number of knots produced is constrained to at most every third order statistic in any predictor. This constraint can be adjusted using the <code>knot.space</code> argument. It can also be a vector with the number of potential knots for each predictor. Again the actual number of knots produced is constrained to be at most every third order statistic any predictor. A third possibility is to provide a matrix where each columns corresponds to the ordered knots you would like to have considered for that predictor. This matrix should be filled out to a rectangular data structure with NAs. The default is $\min(20, \text{round}(n/4))$ knots per predictor. When specifying knots as a vector an entry of -1 indicates that the predictor is a categorical variable and each unique entry in it's column is treated as a level. When specifying knots as a single number or a matrix and there are categorical variables these are specified separately as such using the <code>factor</code> argument.
<code>knot.space</code>	is an integer describing the minimum number of order statistics apart that two knots can be. Knots should not be too close to insure numerical stability.
<code>ts.resp</code>	testset responses for model selection. Should have the same number of columns as the training set response. A testset can be used for the model selection. Depending on the value of <code>classify</code> , either the model with the smallest testset residual sum of squares or the smallest testset classification error is provided. Overrides <code>gcv</code> .
<code>ts.pred</code>	testset predictors. Should have the same number of columns as the training set predictors.
<code>ts.weights</code>	testset observation weights. A vector of length equal to the number of cases of the testset. All weights must be non-negative.
<code>classify</code>	when the response is discrete (categorical), <code>polymars</code> can be used for classification. In particular, when <code>classify = TRUE</code> , a discrete response with K levels is replaced by K indicator variables as response. Model selection is still

	being carried out using <code>gcv</code> , except when a testset is provided, in which case testset misclassification is used to select the best model.
<code>factors</code>	used to indicate that certain variables in the predictor set are categorical variables. Specified as a vector containing the appropriate predictor indices (column numbers of categorical variables in predictors matrix). Factors can also be set when the <code>knots</code> argument is given as a vector, with <code>-1</code> as the appropriate entries for factors.
<code>tolerance</code>	for each possible candidate to be added/deleted the resulting residual sums of squares of the model, with/without this candidate, must be calculated. The inversion of the "X-transpose by X" matrix, X being the design matrix, is done by an updating procedure c.f. C.R. Rao - Linear Statistical Inference and Its Applications, 2nd. edition, page 33. In the inversion the size of the bottom right-hand entry of this matrix is critical. If its value is near zero or the value of its inverse is almost zero then the inversion procedure becomes somewhat inaccurate. The lower the tolerance value the more careful the procedure is in selecting candidates for addition to the model but it may exclude too conservatively. And the other hand if the tolerance is set too high a spurious result with a singular or otherwise sub-optimal model may occur. By default tolerance is set to <code>1.0e-5</code> .
<code>verbose</code>	when set to <code>TRUE</code> , the function will print out a line for each addition or deletion stage. For example, " <code>+ 8 : 5 3.25 2 NA</code> " means adding interaction basis function of predictor 5 with knot at 3.25 and predictor 2 (linear), to make a model of size 8, including intercept.

Value

An object of the class `polymars`. The returned object contains information about the fitting steps and the model selected. The first data frame contains a row for each step of the fitting procedure. In the columns are: a 1 for an addition step or a 0 for a deletion step, the size of the model at each step, residual sums of squares (RSS) and the generalized cross validation value (GCV), testset residual sums of squares or testset misclassification, whatever was used for the model selection. The second data frame, `model`, contains a row for each basis function of the model. Each row corresponds to one basis function (with two possible components). The `pred1` column contains the indices of the first predictor of the basis function. Column `knot1` is a possible knot in this predictor. If this column is `NA`, the first component is linear. If any of the basis functions of the model is categorical then there will be a `level1` column. Column `pred2` is the possible second predictor involved (if it is `NA` the basis function only depends on one predictor). Column `knot2` contains the possible knot for the predictor `pred2`, and it is `NA` when this component is linear. This is a similar format to the `startmodel` argument together with an additional first row corresponding to the intercept but the `startmodel` doesn't use a separate column to specify levels of a categorical variable. If any predictor in `pred2` is categorical then there will be a `level2` column. The column "coefs" (more than one column in the case of multiple response regression) contains the coefficients. The returned object also contains the fitted values and residuals of the data used in fitting the model.

Note

The algorithm employed by `polymars` is different from the MARS(tm) algorithm of Friedman (1991), though it has many similarities. (The name `polymars` has been used for this algorithm

well before MARS was trademarked.) Some of the main differences are:

`polymars` requires linear terms of a predictor to be in the model before nonlinear terms using the same predictor can be added;

`polymars` requires a univariate basis function to be in the model before a tensor-product basis function involving the univariate basis function can be in the model;

during stepwise deletion the same hierarchy is maintained;

`polymars` can be fit to multiple outcomes simultaneously, with categorical outcomes it can be used for multiple classification; and

`polyclass` uses the same modeling strategy as `polymars`, but uses a logistic (polychotomous) likelihood.

MARS is a registered trademark of Jeril, Inc and is used here with permission. Commercial licenses and versions of PolyMARS may be obtained from Salford Systems at <http://www.salford-systems.com>

Author(s)

Martin O'Connor.

References

Charles Kooperberg, Smarajit Bose, and Charles J. Stone (1997). Polychotomous regression. *Journal of the American Statistical Association*, **92**, 117–127.

Friedman, J. H. (1991). Multivariate adaptive regression splines (with discussion). *The Annals of Statistics*, **19**, 1–141.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

`polyclass`, `design.polymars`, `persp.polymars`, `plot.polymars`, `predict.polymars`, `summary.polymars`.

Examples

```
data(state)
state.pm <- polymars(state.region, state.x77, knots = 15, classify = TRUE)
state.pm2 <- polymars(state.x77[, 2], state.x77[,-2], gcv = 2)
plot(fitted(state.pm2), residuals(state.pm2))
```

predict.polymars *Polymars: multivariate adaptive polynomial spline regression*

Description

Produces fitted values for a model of class `polymars`.

Usage

```
## S3 method for class 'polymars':  
predict(object, x, classify = FALSE, intercept, ...)
```

Arguments

<code>object</code>	object of the class <code>polymars</code> , typically the result of <code>polymars</code> .
<code>x</code>	the predictor values at which the fitted values will be computed. The predictor values can be in a number of formats. It can take the form of a vector of length equal to the number of predictors in the original data set or it can be shortened to the length of only those predictors that occur in the model, in the same order as they appear in the original data set. Similarly, <code>x</code> can take the form of a matrix with the number of columns equal to the number of predictors in the original data set, or shortened to the number of predictors in the model.
<code>classify</code>	if the original call to <code>polymars</code> was for a classification problem and you would like the classifications (class predictions), set this option equal to <code>TRUE</code> . Otherwise the function returns a response column for each class (the highest values in each row is its class for the case when <code>classify = TRUE</code>).
<code>intercept</code>	Setting <code>intercept</code> equal to <code>FALSE</code> evaluates the object without intercept. The intercept may also be given any numerical value which overrides the fitted coefficient from the object. The default is <code>TRUE</code> .
<code>...</code>	other arguments are ignored.

Value

A matrix of fitted values. The number of columns in the returned matrix equals the number of responses in the original call to `polymars`.

Author(s)

Martin O'Connor.

References

Charles Kooperberg, Smarajit Bose, and Charles J. Stone (1997). Polychotomous regression. *Journal of the American Statistical Association*, **92**, 117–127.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[polymars](#), [design.polymars](#), [plot.polymars](#), [summary.polymars](#).

Examples

```
data(state)
state.pm <- polymars(state.region, state.x77, knots = 15, classify = TRUE, gcv = 1)
table(predict(state.pm, x = state.x77, classify = TRUE), state.region)
```

summary.hare

Hare: hazard regression

Description

This function summarizes both the stepwise selection process of the model fitting by [hare](#), as well as the final model that was selected using AIC/BIC.

Usage

```
## S3 method for class 'hare':
summary(object, ...)
## S3 method for class 'hare':
print(x, ...)
```

Arguments

`object, x` `hare` object, typically the result of [hare](#).
`...` other arguments are ignored.

Details

These function produce identical printed output. The main body consists of two tables.

The first table has six columns: the first column is a possible number of dimensions for the fitted model;

the second column indicates whether this model was fitted during the addition or deletion stage;

the third column is the log-likelihood for the fit;

the fourth column is $-2 * \text{loglikelihood} + \text{penalty} * (\text{dimension})$, which is the AIC criterion - `hare` selected the model with the minimum value of AIC;

the last two columns give the endpoints of the interval of values of penalty that would yield the model with the indicated number of dimensions (NAs imply that the model is not optimal for any choice of penalty).

At the bottom of the first table the dimension of the selected model is reported, as is the value of penalty that was used.

Each row of the second table summarizes the information about a basis function in the final model. It shows the variables involved, the knot locations, the estimated coefficient and its standard error and Wald statistic (estimate/SE).

Note

Since the basis functions are selected in an adaptive fashion, typically most Wald statistics are larger than (the magical) 2. These statistics should be taken with a grain of salt though, as they are inflated because of the adaptivity of the model selection.

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg, Charles J. Stone and Young K. Truong (1995). Hazard regression. *Journal of the American Statistical Association*, **90**, 78-94.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[hare](#), [plot.hare](#), [dhare](#), [hhare](#), [phare](#), [qhare](#), [rhare](#).

Examples

```
fit <- hare(testhare[,1], testhare[,2], testhare[,3:8])
summary(fit)
```

summary.heft

Heft: hazard estimation with flexible tails

Description

This function summarizes both the stepwise selection process of the model fitting by [heft](#), as well as the final model that was selected using AIC/BIC.

Usage

```
## S3 method for class 'heft':
summary(object, ...)
## S3 method for class 'heft':
print(x, ...)
```

Arguments

`object, x` heft object, typically the result of [heft](#).
`...` other arguments are ignored.

Details

These function produce identical printed output. The main body is a table with six columns:

the first column is a possible number of knots for the fitted model;

the second column is 0 if the model was fitted during the addition stage and 1 if the model was fitted during the deletion stage;

the third column is the log-likelihood for the fit;

the fourth column is $-2 * \text{loglikelihood} + \text{penalty} * (\text{dimension})$, which is the AIC criterion - heft selected the model with the minimum value of AIC;

the fifth and sixth columns give the endpoints of the interval of values of penalty that would yield the model with the indicated number of knots. (NAs imply that the model is not optimal for any choice of penalty.)

At the bottom of the table the number of knots corresponding to the selected model is reported, as are the value of penalty that was used and the coefficients of the log-based terms in the fitted model and their standard errors.

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg, Charles J. Stone and Young K. Truong (1995). Hazard regression. *Journal of the American Statistical Association*, **90**, 78-94.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[heft](#), [plot.heft](#), [dheft](#), [hheft](#), [pheft](#), [qheft](#), [rheft](#).

Examples

```
fit1 <- heft(testhare[,1], testhare[,2])
summary(fit1)
# modify tail behavior
fit2 <- heft(testhare[,1], testhare[,2], leftlog = FALSE, rightlog = FALSE,
            leftlin = TRUE)
summary(fit2)
fit3 <- heft(testhare[,1], testhare[,2], penalty = 0) # select largest model
summary(fit3)
```

`summary.logspline` *Logspline Density Estimation*

Description

This function summarizes both the stepwise selection process of the model fitting by `logspline`, as well as the final model that was selected using AIC/BIC. A `logspline` object was fit using the 1997 knot addition and deletion algorithm. The 1992 algorithm is available using the `oldlogspline` function.

Usage

```
## S3 method for class 'logspline':  
summary(object, ...)  
## S3 method for class 'logspline':  
print(x, ...)
```

Arguments

`object, x` `logspline` object, typically the result of `logspline`
`...` other arguments are ignored.

Details

These function produce identical printed output. The main body is a table with five columns: the first column is a possible number of knots for the fitted model;

the second column is the log-likelihood for the fit;

the third column is $-2 * \text{loglikelihood} + \text{penalty} * (\text{number of knots} - 1)$, which is the AIC criterion; `logspline` selected the model with the smallest value of AIC;

the fourth and fifth columns give the endpoints of the interval of values of penalty that would yield the model with the indicated number of knots. (NAs imply that the model is not optimal for any choice of penalty.) At the bottom of the table the number of knots corresponding to the selected model is reported, as is the value of penalty that was used.

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg and Charles J. Stone. Logspline density estimation for censored data (1992). *Journal of Computational and Graphical Statistics*, **1**, 301–328.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[logspline](#), [plot.logspline](#), [dlogspline](#), [plogspline](#), [qlogspline](#), [rlogspline](#), [oldlogspline](#).

Examples

```
y <- rnorm(100)
fit <- logspline(y)
summary(fit)
```

summary.lspec

Lspec: logspline estimation of a spectral distribution

Description

Summary of a model fitted with [lspec](#)

Usage

```
## S3 method for class 'lspec':
summary(object, ...)
## S3 method for class 'lspec':
print(x, ...)
```

Arguments

`object, x` [lspec](#) object, typically the result of [lspec](#).
`...` other options are ignored.

Details

These function produce an identical printed summary of an [lspec](#) object.

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg, Charles J. Stone, and Young K. Truong (1995). Log spline Estimation of a Possibly Mixed Spectral Distribution. *Journal of Time Series Analysis*, **16**, 359-388.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[lspec](#), [plot.lspec](#), [clspec](#), [dlspec](#), [plspec](#), [rlspec](#).

Examples

```
data(co2)
co2.detrend <- lm(co2~c(1:length(co2)))$residuals
fit <- lspec(co2.detrend)
summary(fit)
```

```
summary.oldlogspline
```

Logspline Density Estimation - 1992 version

Description

This function summarizes both the stepwise selection process of the model fitting by `oldlogspline`, as well as the final model that was selected using AIC/BIC. A `logspline` object was fit using the 1992 knot deletion algorithm (`oldlogspline`). The 1997 algorithm using knot deletion and addition is available using the `logspline` function.

Usage

```
## S3 method for class 'oldlogspline':
summary(object, ...)
## S3 method for class 'oldlogspline':
print(x, ...)
```

Arguments

`object, x` `oldlogspline` object, typically the result of `oldlogspline`
`...` other arguments are ignored.

Details

This function produces the same printed output. The main body is a table with five columns: the first column is a possible number of knots for the fitted model;

the second column is the log-likelihood for the fit;

the third column is $-2 * \text{loglikelihood} + \text{penalty} * (\text{number of knots} - 1)$, which is the AIC criterion; `logspline` selected the model with the smallest value of AIC;

the fourth and fifth columns give the endpoints of the interval of values of penalty that would yield the model with the indicated number of knots. (NAs imply that the model is not optimal for any choice of penalty.) At the bottom of the table the number of knots corresponding to the selected model is reported, as is the value of penalty that was used.

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg and Charles J. Stone. Log spline density estimation for censored data (1992). *Journal of Computational and Graphical Statistics*, **1**, 301–328.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[logspline](#), [oldlogspline](#), [plot.oldlogspline](#), [doldlogspline](#), [poldlogspline](#), [qoldlogspline](#), [roidlogspline](#).

Examples

```
y <- rnorm(100)
fit <- oldlogspline(y)
summary(fit)
```

summary.polyclass *Polyclass: polychotomous regression and multiple classification*

Description

This function summarizes both the stepwise selection process of the model fitting by [polyclass](#), as well as the final model that was selected

Usage

```
## S3 method for class 'polyclass':
summary(object, ...)
## S3 method for class 'polyclass':
print(x, ...)
```

Arguments

`object, x` [polyclass](#) object, typically the result of [polyclass](#).
`...` other arguments are ignored.

Value

These function summarize a [polyclass](#) fit identically. They also give information about fits that could have been obtained with other model selection options in [polyclass](#).

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg, Smarajit Bose, and Charles J. Stone (1997). Polychotomous regression. *Journal of the American Statistical Association*, **92**, 117–127.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

`polyclass`, `plot.polyclass`, `beta.polyclass`, `cpolyclass`, `ppolyclass`, `rpolyclass`.

Examples

```
data(iris)
fit.iris <- polyclass(iris[,5], iris[,1:4])
summary(fit.iris)
```

summary.polymars *Polymars: multivariate adaptive polynomial spline regression*

Description

Gives details of a `polymars` object.

Usage

```
## S3 method for class 'polymars':
summary(object, ...)
## S3 method for class 'polymars':
print(x, ...)
```

Arguments

`object, x` object of the class `polymars`, typically the result of `polymars`.
`...` other arguments are ignored.

Details

These two functions provide identical printed information. about the fitting steps and the model selected. The first data frame contains a row for each step of the fitting procedure. In the columns are: a 1 for an addition step or a 0 for a deletion step, the size of the model at each step, residual sums of squares (RSS) and the generalized cross validation value (GCV), testset residual sums of squares or testset misclassification, whatever was used for the model selection. The second data frame, `model`, contains a row for each basis function of the model. Each row corresponds to one basis function (with two possible components). The `pred1` column contains the indices of the first predictor of the basis function. Column `knot1` is a possible knot in this predictor. If this column is NA, the first component is linear. If any of the basis functions of the model is categorical then

there will be a level1 column. Column pred2 is the possible second predictor involved (if it is NA the basis function only depends on one predictor). Column knot2 contains the possible knot for the predictor pred2, and it is NA when this component is linear. This is a similar format to the startmodel argument together with an additional first row corresponding to the intercept but the startmodel doesn't use a separate column to specify levels of a categorical variable . If any predictor in pred2 is categorical then there will be a level2 column. The column "coefs" (more than one column in the case of multiple response regression) contains the coefficients.

Author(s)

Martin O'Connor.

References

Charles Kooperberg, Smarajit Bose, and Charles J. Stone (1997). Polychotomous regression. *Journal of the American Statistical Association*, **92**, 117–127.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[polymars](#), [design.polymars](#), [persp.polymars](#), [plot.polymars](#), [predict.polymars](#).

Examples

```
data(state)
state.pm <- polymars(state.region, state.x77, knots = 15, classify = TRUE)
summary(state.pm)
```

testhare

Fake survival data for Hare and Heft

Description

Fake survival analysis data set for testing [hare](#) and [heft](#)

Usage

```
testhare
```

Format

A matrix with 2000 lines (observations) and 8 columns. Column 1 is intended to be the survival time, column 2 the censoring indicator, and columns 3 through 8 are predictors (covariates).

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

Source

I started out with a real data set; then I sampled, transformed and added noise. Virtually no number is unchanged.

References

Charles Kooperberg, Charles J. Stone and Young K. Truong (1995). Hazard regression. *Journal of the American Statistical Association*, **90**, 78-94.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[hare](#), [heft](#).

Examples

```
harefit <- hare(testhare[,1], testhare[,2], testhare[,3:8])
heftfit <- heft(testhare[,1], testhare[,2])
```

unstrip

Reformat data as vector or matrix

Description

This function tries to convert a `data.frame` or a matrix to a no-frills matrix without labels, and a vector or time-series to a no-frills vector without labels.

Usage

```
unstrip(x)
```

Arguments

`x` one- or two-dimensional object.

Details

Many of the functions for [logspline](#), [oldlogspline](#), [lspec](#), [polyclass](#), [hare](#), [heft](#), and [polymars](#) were written in the “before `data.frame`” era; `unstrip` attempts to keep all these functions useful with more advanced input objects. In particular, many of these functions call `unstrip` before doing anything else.

Value

If `x` is two-dimensional a matrix without names, if `x` is one-dimensional a numerical vector

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

Examples

```
data(co2)
unstrip(co2)
data(iris)
unstrip(iris)
```

xhare

Hare: hazard regression

Description

Driver function for [dhare](#), [hhare](#), [phare](#), [qhare](#), and [rhare](#). This function is not intended for use by itself.

Usage

```
xhare(arg1, arg2, arg3, arg4)
```

Arguments

```
arg1, arg2, arg3, arg4
arguments.
```

Details

This function is used internally.

Note

This function is not intended for direct use.

Author(s)

Charles Kooperberg <clk@fhcrc.org>.

References

Charles Kooperberg, Charles J. Stone and Young K. Truong (1995). Hazard regression. *Journal of the American Statistical Association*, **90**, 78-94.

Charles J. Stone, Mark Hansen, Charles Kooperberg, and Young K. Truong. The use of polynomial splines and their tensor products in extended linear modeling (with discussion) (1997). *Annals of Statistics*, **25**, 1371–1470.

See Also

[hare](#), [dhare](#), [hhare](#), [phare](#), [qhare](#), [rhare](#).

Index

*Topic **classes**

unstrip, 52

*Topic **datasets**

testhare, 51

*Topic **distribution**

dhare, 6

dheft, 8

dlogspline, 9

doldlogspline, 10

hare, 12

heft, 14

logspline, 16

oldlogspline, 21

oldlogspline.to.logspline, 23

plot.hare, 25

plot.heft, 26

plot.logspline, 28

plot.oldlogspline, 30

summary.hare, 43

summary.heft, 44

summary.logspline, 46

summary.oldlogspline, 48

xhare, 53

*Topic **nonlinear**

beta.polyclass, 1

cpolyclass, 4

design.polymars, 5

persp.polymars, 24

plot.polyclass, 31

plot.polymars, 33

polyclass, 34

polymars, 38

predict.polymars, 42

summary.polyclass, 49

summary.polymars, 50

*Topic **smooth**

beta.polyclass, 1

clspec, 3

cpolyclass, 4

design.polymars, 5

dhare, 6

dheft, 8

dlogspline, 9

doldlogspline, 10

hare, 12

heft, 14

logspline, 16

lspec, 18

oldlogspline, 21

oldlogspline.to.logspline, 23

persp.polymars, 24

plot.hare, 25

plot.heft, 26

plot.logspline, 28

plot.lspec, 29

plot.oldlogspline, 30

plot.polyclass, 31

plot.polymars, 33

polyclass, 34

polymars, 38

predict.polymars, 42

summary.hare, 43

summary.heft, 44

summary.logspline, 46

summary.lspec, 47

summary.oldlogspline, 48

summary.polyclass, 49

summary.polymars, 50

xhare, 53

*Topic **survival**

dhare, 6

dheft, 8

hare, 12

heft, 14

plot.hare, 25

plot.heft, 26

summary.hare, 43

summary.heft, 44

- testhare, 51
- xhare, 53
- *Topic **ts**
 - clspec, 3
 - lspec, 18
 - plot.lspec, 29
 - summary.lspec, 47
- *Topic **utilities**
 - unstrip, 52
- beta.polyclass, 1, 5, 32, 36–38, 50
- clspec, 3, 20, 30, 47
- contour, 33
- cpolyclass, 2, 4, 32, 36, 38, 50
- design.polymars, 5, 34, 41, 43, 51
- dhare, 6, 13, 14, 26, 44, 53, 54
- dheft, 8, 15, 16, 27, 45
- dlogspline, 9, 17, 18, 29, 47
- dlspec, 20, 30, 47
- dlspec (clspec), 3
- doldlogspline, 10, 22, 31, 49
- fft, 19
- hare, 6, 7, 12, 13, 16, 25, 26, 43, 44, 51, 52, 54
- heft, 8, 9, 12, 14, 14, 27, 44, 45, 51, 52
- hhare, 13, 14, 26, 44, 53, 54
- hhare (dhare), 6
- hheft, 15, 16, 27, 45
- hheft (dheft), 8
- logspline, 9–11, 16, 16, 21, 22, 24, 28–31, 46–49, 52
- lspec, 3, 4, 18, 29, 30, 47, 52
- oldlogspline, 9–11, 16, 18, 21, 21, 22, 24, 28, 29, 31, 46–49, 52
- oldlogspline.to.logspline, 18, 22, 23
- persp, 33
- persp.polymars, 24, 41, 51
- phare, 13, 14, 26, 44, 53, 54
- phare (dhare), 6
- pheft, 15, 16, 27, 45
- pheft (dheft), 8
- plogspline, 17, 18, 29, 47
- plogspline (dlogspline), 9
- plot.hare, 7, 13, 14, 25, 44
- plot.heft, 9, 15, 16, 26, 45
- plot.logspline, 10, 17, 18, 28, 47
- plot.lspec, 4, 20, 29, 47
- plot.oldlogspline, 11, 22, 30, 49
- plot.polyclass, 2, 5, 31, 36, 38, 50
- plot.polymars, 6, 24, 25, 33, 41, 43, 51
- plspec, 20, 30, 47
- plspec (clspec), 3
- poldlogspline, 22, 31, 49
- poldlogspline (doldlogspline), 10
- polyclass, 2, 4, 5, 31, 32, 34, 35, 41, 49, 50, 52
- polymars, 6, 25, 33, 34, 38, 38, 40, 42, 43, 50–52
- ppolyclass, 2, 32, 36, 38, 50
- ppolyclass (cpolyclass), 4
- predict.polymars, 6, 34, 41, 42, 51
- print.hare (summary.hare), 43
- print.heft (summary.heft), 44
- print.logspline
 - (summary.logspline), 46
- print.lspec (summary.lspec), 47
- print.oldlogspline
 - (summary.oldlogspline), 48
- print.polyclass
 - (summary.polyclass), 49
- print.polymars
 - (summary.polymars), 50
- qhare, 13, 14, 26, 44, 53, 54
- qhare (dhare), 6
- qheft, 15, 16, 27, 45
- qheft (dheft), 8
- qlogspline, 17, 18, 29, 47
- qlogspline (dlogspline), 9
- qoldlogspline, 22, 31, 49
- qoldlogspline (doldlogspline), 10
- rhare, 13, 14, 26, 44, 53, 54
- rhare (dhare), 6
- rheft, 15, 16, 27, 45
- rheft (dheft), 8
- rlogspline, 17, 18, 29, 47
- rlogspline (dlogspline), 9
- rlspec, 20, 30, 47
- rlspec (clspec), 3
- roldlogspline, 22, 31, 49

`roldlogspline` (*doldlogspline*), 10
`rpolyclass`, 2, 32, 36, 38, 50
`rpolyclass` (*cpolyclass*), 4

`set.seed`, 36
`summary.hare`, 7, 12–14, 26, 43
`summary.heft`, 9, 14–16, 27, 44
`summary.logspline`, 10, 16–18, 29, 46
`summary.lspec`, 4, 20, 30, 47
`summary.oldlogspline`, 11, 22, 31, 48
`summary.polyclass`, 2, 5, 32, 36, 38, 49
`summary.polymars`, 6, 34, 41, 43, 50

`testhare`, 51

`unstrip`, 52

`xhare`, 53