Package ‘polspline’

May 8, 2020

Version 1.1.18
Date 2020-05-08
Title Polynomial Spline Routines
Description Routines for the polynomial spline fitting routines
hazard regression, hazard estimation with flexible tails, logspline,
lspec, polyclas, and polymars, by C. Kooperberg and co-authors.
Imports stats, graphics
License GPL (>= 2)
NeedsCompilation yes
Author Charles Kooperberg [aut, cre],
Cleve Moler [ctb] (LINPACK routines in src),
Jack Dongarra [ctb] (LINPACK routines in src)
Maintainer Charles Kooperberg <clk@fredhutch.org>
Repository CRAN
Date/Publication 2020-05-08 18:10:06 UTC

R topics documented:

beta.polyclass ........................................ 2
cspec .................................................. 3
cpolyclass ............................................ 5
design.polymars ....................................... 6
dhare ................................................... 7
dheft .................................................... 9
dlogspline ............................................. 10
doldlogspline ......................................... 11
hare .................................................... 12
heft .................................................... 15
logspline .............................................. 17
lspec ................................................... 19
oldlogspline .......................................... 21
oldlogspline.to.logspline ........................... 24
**beta.polyclass**

*Polyclas*: polychotomous regression and multiple classification

**Description**

Produces a beta-plot for a polyclass object.

**Usage**

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

**Arguments**

- **fit**  
  polyclass object, typically the result of `polyclass`.
- **which**
  which classes should be compared? Default is to compare all classes.
- **xsp**
  location of the vertical line to the left of the axis. Useful for making high quality, device dependent, graphics.
- **cex**
  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@fredhutch.org>.

References


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)

---

**clspect**

*Lspec: logspline estimation of a spectral distribution*

**Description**

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

**Usage**

```r
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**: lspec object, typically the result of \texttt{lspec}.
- **cov**: compute autocovariances (\texttt{TRUE}) or autocorrelations (\texttt{FALSE}).
- **mm**: number of points used in integration and the fft. Default is the smallest power of two larger than \texttt{max(fit$sample, max(lag), 1024)} for \texttt{clspec} and \texttt{plspec} or the smallest power of two larger than \texttt{max(fit$sample, n, max(lag), 1024)} for \texttt{rlspec}.
- **freq**: vector of frequencies. For \texttt{plspec} frequencies should be between $-\pi$ and $\pi$.
- **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 (\texttt{clspec}); values of the spectral distribution at the requested frequencies. (\texttt{plspec}); random time series of length \texttt{n} (\texttt{rlspec}); or a list with three components (\texttt{dlspec}):

- **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@fredhutch.org>.

References


See Also

\texttt{lspec, plot.lspec, summary.lspec}. 
**Examples**

```r
data(co2)
co2.detrend <- lm(co2~c(1:length(co2)))$residuals
fit <- lspec(co2.detrend)
clspec(0:12, fit)
plspec((0:314)/100, fit)
dlspec((0:314)/100, fit)
rlspec(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**

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

**Arguments**

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


See Also

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

Examples

```r
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

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

Arguments

- **object**: object of the class *polymars*, typically the result of *polymars*.
- **x**: 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, `x` 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.
dhare

Author(s)
Charles Kooperberg

References


See Also
polymars, plot.polymars, predict.polymars, summary.polymars.

Examples
```r
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**

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

fit hare 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@fredhutch.org>`.

References


See Also

`hare`, `plot.hare`, `summary.hare`.

Examples

```r
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)
```
**Description**

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

**Usage**

```r
 dheft(q, fit)
 hheft(q, fit)
 pheft(q, fit)
 qheft(p, fit)
 rheft(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 `length(n)` is larger than 1, then `length(n)` random values are returned.
- `fit` 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@fredhutch.org>.

**References**


See Also

`heft`, `plot.heft`, `summary.heft`.

Examples

```r
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

**Logspline Density Estimation**

Description

Density (`dlogspline`), cumulative probability (`plogspline`), quantiles (`qlogspline`), and random samples (`rlogspline`) 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

```r
dlogspline(q, fit)
plogspline(q, fit)
qlogspline(p, fit)
rlogspline(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 `length(n)` is larger than 1, then `length(n)` random values are returned.
- `fit` logspline object, typically the result of `logspline`.

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 logspline density that was fitted using knot addition and deletion.

Author(s)

Charles Kooperberg <clk@fredhutch.org>.
References


See Also

`logspline`, `plot.logspline`, `summary.logspline`, `oldlogspline`.

Examples

```r
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))  # assesses 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))  # assesses 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

```r
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 length(n) is larger than 1, then length(n) 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@fredhutch.org>.

References


See Also

logspline, oldlogspline, plot.oldlogspline, summary.oldlogspline

Examples

```r
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))  # assess 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))  # assess the fit of the density
rr <- roldlogspline(100, fit)  # random sample from fit
```

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.
hare

Usage

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

Arguments

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

ncov number of covariates.
ndim number of dimensions of the fitted model.
fcts  matrix of size ndim x 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 ncov rows. Covariate i has row i+1, 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 i gives the range of the i-th covariate.

logl  matrix with two columns. The i-th element of the first column is the loglikelihood of the model of dimension i. 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@fredhutch.org>.

References


See Also

*heft, plot.hare, summary.hare, dhare, hhare, phare, qhare, rhare.*

Examples

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

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

data

vector of observations. Observations may or may not be right censored. All observations should be nonnegative.

delta

binary vector with the same length as data. Elements of data for which the corresponding element of delta is 0 are assumed to be right censored, elements of data for which the corresponding element of delta is 1 are assumed to be uncensored. If delta is missing, all observations are assumed to be uncensored.

penalty

the parameter to be used in the AIC criterion. The method chooses the number of knots that minimizes -2 * loglikelihood + penalty * (dimension). The default is to use penalty = log(samplesize) as in BIC. The effect of this parameter is summarized in summary.heft.

knots

ordered vector of values, which forces the method to start with these knots. If knots is not specified, a default knot-placement rule is employed.

leftlin

if leftlin is TRUE an extra basis-function, which is linear to the left of the first knot, is included in the basis. If any of data is exactly 0, the default of leftlin is TRUE, otherwise it is FALSE.

shift

parameter for the log terms. Default is quantile(data[delta == 1], .75).

leftlog

coefficient of log \( \frac{x}{x + \text{shift}} \), which must be greater than -1. (In particular, if leftlog equals zero no log \( \frac{x}{x + \text{shift}} \) term is included.) If leftlog is missing its maximum likelihood estimate is used. If any of data is exactly zero, leftlog is set to zero.

rightlog

coefficient of log \( x + \text{shift} \), which must be greater than -1. (In particular, if leftlog equals zero no log \( x + \text{shift} \) term is included.) If rightlog is missing its maximum likelihood estimate is used.

maxknots

maximum number of knots allowed in the model (default is \( 4 \times n^{0.2} \)), where \( n \) is the length of data.

mindist

minimum distance in order statistics between knots. The default is 5.

silent

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:

- **knots**: vector of the locations of the knots in the heft model.
- **logl**: the k-th element is the log-likelihood of the fit with k knots.
- **thetak**: 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.
- **thetap**: 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 `leftlog` and the second element equals `rightlog`.
- **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 `thetal`.
- **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@fredhutch.org>.

References


See Also

`hare`, `plot.heft`, `summary.heft`, `dheft`, `hheft`, `pheft`, `qheft`, `rheft`.

Examples

```r
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
```
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

x  
data vector. The data needs to be uncensored. oldlogspline can deal with right- left- and interval-censored data.

lbound, ubound  
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 lbound = 0. However, if the density is essentially zero near 0, one does not need to specify lbound.

maxknots  
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 maxknots if this parameter is not specified.

knots  
ordered vector of values (that should cover the complete range of the observations), which forces the method to start with these knots. Overrules knots. If knots is not specified, a default knot-placement rule is employed.

nknots  
forces the method to start with nknots knots. The method has an automatic rule for selecting nknots if this parameter is not specified.

penalty  
the parameter to be used in the AIC criterion. The method chooses the number of knots that minimizes \(-2 \times \text{loglikelihood} + \text{penalty} \times (\text{number of knots} - 1)\). The default is to use a penalty parameter of \(\text{penalty} = \log(\text{samplesize})\) as in BIC. The effect of this parameter is summarized in summary.logspline.

silent  
should diagnostic output be printed?

mind  
minimum distance, in order statistics, between knots.

error.action  
how should logspline deal with non-convergence problems? Very-very rarely in some extreme situations logspline 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. logspline can deal with this in three ways. If error.action is 2, the same data is re-run with the slightly more stable, but less flexible oldlogspline. The object is translated in a logspline object using oldlogspline.to.logspline, so this is almost invisible to the user. It is particularly useful when you run simulation studies, as he code can seemlessly continue. Only the lbound and ubound
options are passed on to oldlogspline, other options revert to the default. If error.action is 1, a warning is printed, and logspline returns nothing (but does not crash). This is useful if you run a simulation, but do not like to revert to oldlogspline. If error.action is 0, the code crashes using the stop 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:

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

Author(s)

Charles Kooperberg <clk@fredhutch.org>.

References


lspec

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

data time series (exactly one of data and period should be specified). If data is specified, lspec first computes the modulus of the fast Fourier transform of the series using the function \texttt{fft}, resulting in a periodogram of length \texttt{floor(length(data)/2)}.

period value of the periodogram for a time series at frequencies $\frac{2\pi j}{T}$, for $1 \leq j \leq T/2$. If period is specified, odd should indicate whether the length of the series $T$ is odd (odd = TRUE) or even (odd = FALSE). Exactly one of data and period should be specified.

Examples

```r
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 <- round(zz)
fit <- logspline(zz)
#
# you could rerun this with
# fit <- logspline(zz, error.action=0)
# or
# fit <- logspline(zz, error.action=1)
```

See Also

plot.logspline, summary.logspline, dlogspline, plogspline, qlogspline, rlogspline, oldlogspline, oldlogspline.to.logspline.
penalty the parameter to be used in the AIC criterion. The method chooses the number of basis functions that minimizes \(-2 \cdot \text{loglikelihood} + \text{penalty} \cdot (\text{number of basis functions})\). Default is to use a penalty parameter of \(\text{penalty} = \log(\text{length}(\text{period}))\) as in BIC.

minmass threshold value for atoms. No atoms having smaller mass than \(\text{minmass}\) are included in the model. If \(\text{minmass}\) takes its default value, in 95% of the samples, when data is Gaussian white noise, the model will not contain atoms.

knots ordered vector of values, which forces the method to start with these knots. If \(\text{knots}\) is not specified, the program starts with one knot at zero and then employs stepwise addition of knots and atoms.

maxknots maximum number of knots allowed in the model. Does not need to be specified, since the program has a default for \(\text{maxdim}\) and the number of dimensions equals the number of knots plus the number of atoms. If \(\text{maxknots} = 1\) the fitted spectral density function is constant.

atoms ordered vector of values, which forces the method to start with discrete components at these frequencies. The values of atoms are rounded to the nearest multiple of \(\frac{2\pi}{T}\). If \(\text{atoms}\) is not specified, the program starts with no atoms and then performs stepwise addition of knots and atoms.

maxatoms maximum number of discrete components allowed in the model. Does not need to be specified, since the program has a default for \(\text{maxdim}\) and the number of dimensions equals the number of knots plus the number of atoms. If \(\text{maxatoms} = 0\) a continuous spectral distribution is fit.

maxdim maximum number of basis functions allowed in the model (default is \(\max(15, 4 \times \text{length}(\text{period})^{0.2})\)).

odd see \(\text{period}\). If \(\text{period}\) is not specified, odd is not relevant.

updown the maximal number of times that \text{lspec} should go through a cycle of stepwise addition and stepwise deletion until a stable solution is reached.

silent should printing of information be suppressed?

Value

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

call the command that was executed.

thetap coefficients of the polynomial part of the spline.

nknots the number of knots that were retained.

knots vector of the locations of the knots in the logspline model. Only the knots that were retained are in this vector.

thetak coefficients of the knot part of the spline. The \(k\)-th coefficient is the coefficient of \((x - t(k))^3\).

natoms the number of atoms that were retained.
oldlogspline

atoms vector of the locations of the atoms in the model. Only the atoms that were retained are in this vector.

mass The k-th coefficient is the mass at atom[k].

logl the log-likelihood of the model.

penalty the penalty that was used.

minmass the minimum mass for an atom that was allowed.

sample the sample size that was used, either computed as length(data) or as (2 * length(period)) when odd = FALSE or as (2 * length(period) + 1) when odd = TRUE.

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@fredhutch.org>.

References


See Also

plot.lspec, summary.lspec, clspec, dlspec, plspec, rlspec.

Examples

data(co2)
c02.detrend <- unstrip(lm(co2~c(1:length(co2)))$residuals)
fit <- lspec(co2.detrend)

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.

right 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.

left 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.

interval 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.

lbound, ubound 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 lbound = 0. However, if the density is essentially zero near 0, one does not need to specify lbound. The default for lbound is -inf and the default for ubound is inf.

nknots forces the method to start with nknots knots (delete = TRUE) or to fit a density with nknots knots (delete = FALSE). The method has an automatic rule for selecting nknots if this parameter is not specified.

knots ordered vector of values (that should cover the complete range of the observations), which forces the method to start with these knots (delete = TRUE) or to fit a density with these knots delete = FALSE). Overrules nknots. If knots is not specified, a default knot-placement rule is employed.

penalty the parameter to be used in the AIC criterion. The method chooses the number of knots that minimizes \(-2 \times \text{loglikelihood} + \text{penalty} \times (\text{number of knots} - 1)\). The default is to use a penalty parameter of \(\text{penalty} = \log(\text{samplesize})\) as in BIC. The effect of this parameter is summarized in `summary.oldlogspline`.

delete 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), `roldlogspline` (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:

call the command that was executed.
knots vector of the locations of the knots in the oldlogspline model.
coef 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 \(-\infty\) it was something else; second element: lbound, if specified; third element: 0 - ubound was \(\infty\), 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@fredhutch.org>.

References


See Also

*logspline, oldlogspline, plot.oldlogspline, summary.oldlogspline, doldlogspline, poldlogspline, qoldlogspline, roldlogspline, oldlogspline.to.logspline.*

Examples

```r
# 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

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

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@fredhutch.org>.

References


See Also

logspline, oldlogspline.
Examples

```r
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
#
```

Description

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

Usage

```r
## 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


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

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

#### Arguments

- `x` hare object, typically the result of *hare*.
- `cov` a vector of length `fit$ncov`, indicating for which combination of covariates the plot should be made. Can be omitted only if `fit$ncov` is 0.
- `n` the number of equally spaced points at which to plot the function.
- `which` for which coordinate should the plot be made. 0: time; positive value i: covariate i. Note that if which is the positive value i, then the element corresponding to this covariate must be given in cov even though its actual value is irrelevant.
- `what` what should be plotted: "d" (density), "p" (distribution function), "s" (survival function) or "h" (hazard function).
- `time` if which is not equal to 0, the value of time for which the plot should be made.
- `add` should the plot be added to an existing plot?
- `xlim` 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.
- `xlab, ylab` labels for the axes. Per default no labels are printed.
- `type` plotting type. The default is lines.
- `...` all other plotting options are passed on.
plot.heft

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@fredhutch.org>.

References


See Also

hare, summary.hare, dhare, hhare, phare, qhate, rhare.

Examples

```r
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)
```

---

**Description**

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

**Usage**

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

- **x**: heft object, typically the result of `heft`.
- **n**: the number of equally spaced points at which to plot the function.
- **what**: what should be plotted: "d" (density), "p" (distribution function), "s" (survival function) or "h" (hazard function).
- **add**: should the plot be added to an existing plot?
- **xlim**: 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.
- **xlab,ylab**: labels for the axes. The default is no labels.
- **type**: plotting type. The default is lines.
- **...**: 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@fredhutch.org>.

References


See Also

-heft, summary.heft, dheft, hheft, pheft, qheft, rheft.

Examples

```r
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)
```
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

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

Arguments

- `x` logspline object, typically the result of `logspline`.
- `n` the number of equally spaced points at which to plot the density.
- `what` what should be plotted: "d" (density), "p" (distribution function), "s" (survival function) or "h" (hazard function).
- `add` should the plot be added to an existing plot.
- `xlim` 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.
- `xlab, ylab` labels plotted on the axes.
- `type` type of plot.
- `...` 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@fredhutch.org>.

References


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

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

Arguments

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

Note

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

See Also

- `logspline`, `summary.logspline`, `dlogspline`, `plogspline`, `qlogspline`, `rlogspline`, `oldlogspline`.

Examples

```r
y <- rnorm(100)
fit <- logspline(y)
plot(fit)
```
Author(s)
Charles Kooperberg <clk@fredhutch.org>.

References

See Also
`lspec`, `summary.lspec`, `clspec`, `dlspec`, `plspec`, `rlspec`.

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

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

Arguments
- `x` logspline object, typically the result of `logspline`.
- `n` the number of equally spaced points at which to plot the density.
- `what` what should be plotted: "d" (density), "p" (distribution function), "s" (survival function) or "h" (hazard function).
- `xlim` 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.
- `xlab, ylab` labels plotted on the axes.
### plot.polyclass

**type**

*type of plot.*

**add**

*should the plot be added to an existing plot.*

... *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@fredhutch.org>`.

### References


### See Also

`logspline`, `oldlogspline`, `summary.oldlogspline`, `doldlogspline`, `poldlogspline`, `qoldlogspline`, `roldlogspline`.

### Examples

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

### Description

Probability or classification plots for a `polyclass` model.

### Usage

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

---

**plot.polyclass**

*Polyclass: polychotomous regression and multiple classification*
Arguments

- **x**: polyclass object, typically the result of `polyclass`.
- **cov**: a vector of length `fit$ncov`, indicating for which combination of covariates the plot should be made. Can never be omitted. Should always have length `fit$ncov`, 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 `what` is 6 or larger and length two if `what` 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 `c(min,max)` if `what` is 6 or larger and a vector of length four of the form `c(xmin,xmax,ymin,ymax)` if `what` 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@fredhutch.org>.

References


See Also

`polyclass`, `summary.polyclass`, `beta.polyclass`, `cpolyclass`, `ppolyclass`, `rpolyclass`. 
Examples

data(iris)
fit.iris <- polymars(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

x polymars object, typically the result of polymars.
predictor1 the index of a predictor that was used when the polymars model was fit. For the
two dimensional plots, this variable is plotted along the X-axis.
response if the model was fitted to multiple response data the response index should be
specified.
predictor2 the index of a predictor that was used when the polymars model was fit. For the
three dimensional plots, this variable is plotted along the Y-axis. See xyz.
xx 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 predictor and predictor2 are not used, they should still
be provided as part of xx.
add should the plot be added to a previously created plot? Works only for two di-
menional plots.
n number of plotting points (2 dimensional plot) or plotting points along each axis
(3 dimensional plot). The default is n = 100 for 2 dimensional plots and n = 33
for 3 dimensional plots.
xyz 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 persp
function. Categorical predictors cannot be used for 3 dimensional plots.
contour.polymars

if the plot being made a 3 dimensional plot should it be made as a contour plot (TRUE) or a perspective plot (FALSE). function contour is being made.

intercept

Setting intercept equal to FALSE 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 TRUE.

xlim, ylim

Plotting limits. The function tries to choose intelligent limits itself

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


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)
Usage

polycast(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

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

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)?

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. `loss[i,j]` contains the loss for assigning action `j` to an object whose true class is `i`. The default is 1 minus the identity matrix. `loss` does not need to be square.

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 `.Random.seed`, otherwise the function `set.seed` is used. If seed is 0 or `rep(0,12)`, it is assumed that the user has already provided a (random) ordering. If seed is not provided, while a fit with an element `fit$seed` is provided, `.Random.seed` is set using `set.seed(fit$seed)`. Otherwise the present value of `.Random.seed` is used.

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 `nbas` x `(nclass + 4)`. 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 `ncov` rows. Covariate `i` has row `i+1`, 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 `method = 2`.
- `loss`: the loss matrix used in cross-validation and test set. Only provided if `method = 1` or `method = 2`. 
penalty

the parameter used in the AIC criterion. Only provided if method = 0.

method

0 = AIC, 1 = test set, 2 = cross-validation.

ranges

column i gives the range of the i-th covariate.

logl

matrix with eight or eleven columns. Summarizes fits. Column one indicates the
dimension, column 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 method = 0 or method = 2) 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 prvided if method = 1.

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 method = 2.

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
method = 2.

seed

the random seed that was used to determine the order of the cases for cross-
validation. Only provided if method = 2.

delete

were complete basis functions deleted at once (2), were only individual dimen-
sions deleted (1) or was only the addition stage of the model selection carried
out (0)?

beta

moments of basisfunctions. Needed for beta.polyclass.

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 method = 1).

Author(s)

Charles Kooperberg <clk@fredhutch.org>.
polymars

References


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 \(\min(6 \times (n^{1/3}), n/4, 100)\), 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 \(1-(gcv \times \text{model size})/\text{cases}\). A larger gcv value would tend to produce a smaller model. Models for which \(1-(gcv \times \text{model size})/\text{cases}\) is smaller or equal than 0 are never selected.
**additive**

Should the fitted model be additive in the predictors?

**startmodel**

the first model that is to be fit by polymars. It is either an object of the class polymars 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 $[X_3 \ast (X_2 - 4.7)]$; 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.

**weights**

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.

**no.interact**

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.

**knots**

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 knot.space 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 factor argument.

**knot.space**

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.

**ts.resp**

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 classify, either the model with the smallest testset residual sum of squares or the smallest testset classification error is provided. Overrides gcv.
testset predictors. Should have the same number of columns as the training set predictors.

testset observation weights. A vector of length equal to the number of cases of the testset. All weights must be non-negative.

classify when the response is discrete (categorical), polymars can be used for classification. In particular, when classify = TRUE, a discrete response with K levels is replaced by K indicator variables as response. Model selection is still being carried out using gcv, except when a testset is provided, in which case testset misclassification is used to select the best model.

factors 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 knots argument is given as a vector, with -1 as the appropriate entries for factors.

tolerance 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 1.0e-5.

verbose when set to TRUE, the function will print out a line for each addition or deletion stage. For example, "+ 8 : 5 3.25 2 NA" 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


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**

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

**Arguments**

- `object`  
  object of the class `polymars`, typically the result of `polymars`.

- `x`  
  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, `x` 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.

- `classify`  
  if the original call to polymars was for a classification problem and you would like the classifications (class predictions), set this option equal to `TRUE`. Otherwise the function returns a response column for each class (the highest values in each row is its class for the case when `classify = TRUE`).

- `intercept`  
  Setting `intercept` equal to `FALSE` 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 `TRUE`.

- `...`  
  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**


Summary:

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

```r
## 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 * loglikelihood + penalty * (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 (NA's 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).

Examples

```r
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)
```

See Also

polymars, design.polymars, plot.polymars, summary.polymars.
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@fredhutch.org>.

References


See Also

hare, plot.hare, dhare, hhare, phare, qhare, rhare.

Examples

```r
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

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

Arguments

- `object, x`  
  heft object, typically the result of heft.
- `...`  
  other arguments are ignored.
Details

These functions 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 $-2 \times \text{loglikelihood} + \text{penalty} \times (\text{dimension})$, 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. (NA s 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 &lt;clk@fredhutch.org&gt;.

References

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

See Also

heft, plot.heft, dheft, hheft, pheft, qheft, rheft.

Examples

```r
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)
```
### 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

```r
## 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 \(-2 \times \text{loglikelihood} + \text{penalty} \times (\text{number of knots} - 1)\), 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@fredhutch.org>.

### References

See Also

`logspline`, `plot.logspline`, `dlogspline`, `plogspline`, `qlogspline`, `rlogspline`, `oldlogspline`.

Examples

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

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

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

```r
## 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

These 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 \(-2 \times \text{loglikelihood} + \text{penalty} \times (\text{number of knots} -1)\), 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@fredhutch.org>`.
References


See Also

`logspline`, `oldlogspline`, `plot.oldlogspline`, `doldlogspline`, `poldlogspline`, `qoldlogspline`, `roldlogspline`.

Examples

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

### summary.polyclass

**Polyclas**: 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

```r
## 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@fredhutch.org>.


References


See Also

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

Examples

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

Description

Gives details of a polymars object.

Usage

```r
## 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. 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

See Also
polymars, design.polymars, persp.polymars, plot.polymars, predict.polymars.

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

Description
Fake survival data for 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@fredhutch.org>.
Source

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

References


See Also

*hare, heft*.

Examples

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

<table>
<thead>
<tr>
<th>unstrip</th>
<th>Reformat data as vector or matrix</th>
</tr>
</thead>
</table>

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

```r
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
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@fredhutch.org>.

References


See Also

hare, dhare, hhare, phare, qhare, rhare.
Index

*Topic **classes**
  unstrip, 53
*Topic **datasets**
  testhare, 52
*Topic **distribution**
  dhare, 7
dheft, 9
dlogspline, 10
doldlogspline, 11
hare, 12
heft, 15
logspline, 17
oldlogspline, 21
oldlogspline.to.logspline, 24
plot.hare, 26
plot.heft, 27
plot.logspline, 29
plot.oldlogspline, 31
summary.hare, 44
summary.heft, 45
summary.logspline, 47
summary.oldlogspline, 49
xhare, 54
*Topic **nonlinear**
  beta.polyclass, 2
cpolyclass, 5
design.polymars, 6
persp.polymars, 25
plot.polyclass, 32
plot.polymars, 34
polyclass, 35
polymars, 39
predict.polymars, 43
summary.polyclass, 50
summary.polymars, 51
*Topic **smooth**
  beta.polyclass, 2
clspec, 3
cpolyclass, 5
design.polymars, 6
dhare, 7
dheft, 9
dlogspline, 10
doldlogspline, 11
hare, 12
heft, 15
logspline, 17
lspec, 19
oldlogspline, 21
oldlogspline.to.logspline, 24
persp.polymars, 25
plot.hare, 26
plot.heft, 27
plot.logspline, 29
plot.lspec, 30
plot.oldlogspline, 31
plot.polyclass, 32
plot.polymars, 34
polyclass, 35
polymars, 39
predict.polymars, 43
summary.hare, 44
summary.heft, 45
summary.logspline, 47
summary.lspec, 48
summary.oldlogspline, 49
summary.polyclass, 50
summary.polymars, 51
xhare, 54
*Topic **survival**
  dhare, 7
dheft, 9
hare, 12
heft, 15
plot.hare, 26
plot.heft, 27
summary.hare, 44
summary.heft, 45
INDEX

*Topic ts
  clspec, 3
  lspec, 19
  plot.lspec, 30
  summary.lspec, 48
*Topic utilities
  unstrip, 53

beta.polymars, 2, 6, 33, 37–39, 51

clspec, 3, 20, 21, 31, 48
contour, 35
cpolyclass, 3, 5, 33, 37, 39, 51
design.polymars, 6, 35, 42, 44, 52
dhare, 7, 13, 14, 27, 45, 54, 55
dheft, 9, 16, 28, 46
dlogspline, 10, 18, 19, 30, 48
dlspec, 20, 21, 31, 48
dlspec (clspec), 3
doldlogspline, 11, 22, 23, 32, 50

fft, 19

hare, 7, 8, 12, 13, 16, 26, 27, 44, 45, 52, 53, 55
heft, 9, 10, 13, 14, 15, 28, 45, 46, 52, 53
hhare, 13, 14, 27, 45, 54, 55
h hare (dhare), 7
h heft, 16, 28, 46
h heft (dheft), 9

logspline, 10–12, 17, 17, 21, 23, 24, 29–32, 47–50, 53
lspec, 3, 4, 19, 30, 31, 48, 53

oldlogspline, 10–12, 17, 19, 21, 21, 23, 24, 29, 30, 47–50, 53
oldlogspline to logspline, 19, 22, 23, 24

persp, 34
persp.polymars, 25, 42, 52
phare, 13, 14, 27, 45, 54, 55
phare (dhare), 7
pheft, 16, 28, 46
pheft (dheft), 9
plogspline, 18, 19, 30, 48
plogspline (dlogspline), 10
plot.hare, 8, 13, 14, 26, 45

plot.heft, 10, 16, 27, 46
plot.llogspline, 11, 18, 19, 29, 48
plot.lspec, 4, 20, 21, 30, 48
plot.oldlogspline, 12, 22, 23, 31, 50
plot.polymars, 3, 6, 32, 37, 39, 51
plot.polymars, 7, 25, 26, 34, 42, 44, 52
plspec, 20, 21, 31, 48
plspec (clspec), 3
poldlogspline, 22, 23, 32, 50
poldlogspline (doldlogspline), 11
polymars, 2, 3, 5, 6, 32, 33, 35, 36, 42, 50, 51, 53
polymars, 6, 7, 26, 34, 35, 39, 41, 43, 44, 51–53
ppolymars, 3, 33, 37, 39, 51
ppolymars (cpolyclass), 5
predict.polymars, 7, 35, 42, 43, 52
print.hare (summary.hare), 44
print.heft (summary.heft), 45
print.llogspline (summary.llogspline), 47
print.lspec (summary.lspec), 48
print.oldlogspline
  (summary.oldlogspline), 49
print.polymars (summary.polymars), 50
print.polymars (summary.polymars), 51

qhare, 13, 14, 27, 45, 54, 55
qhare (dhare), 7
qheft, 16, 28, 46
qheft (dheft), 9
qlogspline, 18, 19, 30, 48
qlogspline (dlogspline), 10
qoldlogspline, 22, 23, 32, 50
qoldlogspline (doldlogspline), 11

rhare, 13, 14, 27, 45, 54, 55
rhare (dhare), 7
rheft, 16, 28, 46
rheft (dheft), 9
rlogspline, 18, 19, 30, 48
rlogspline (dlogspline), 10
rlspec, 20, 21, 31, 48
rlspec (clspec), 3
roldlogspline, 22, 23, 32, 50
roldlogspline (doldlogspline), 11
rpolymars, 3, 33, 37, 39, 51
rpolymars (cpolyclass), 5

set.seed, 37
INDEX

summary.hare, 8, 13, 14, 27, 44
summary.heft, 10, 15, 16, 28, 45
summary.logspine, 11, 17–19, 30, 47
summary.lspec, 4, 20, 21, 31, 48
summary.oldlogspine, 12, 22, 23, 32, 49
summary.polyclass, 3, 6, 33, 37, 39, 50
summary.polymars, 7, 35, 42, 44, 51

testhare, 52
unstrip, 53
xhare, 54