Package 'splines2'

June 14, 2018

Title Regression Spline Functions and Classes

Version 0.2.8

Date 2018-06-14

Description Constructs B-splines and its integral, monotone splines (M-splines) and its integral (I-splines), convex splines (C-splines), and their derivatives of given order. Piecewise constant basis is allowed for B-splines and M-splines. See De Boor (1978) <doi:10.1002/zamm.19800600129>, Ramsay (1988) <doi:10.1214/ss/1177012761>, and Meyer (2008) <doi:10.1214/08-AOAS167> for more information.

Imports splines, stats

Suggests knitr, rmarkdown, testthat

Depends R (>= 3.2.3)

VignetteBuilder knitr

License GPL (>= 3)

URL https://github.com/wenjie2wang/splines2

BugReports https://github.com/wenjie2wang/splines2/issues

RoxygenNote 6.0.1

NeedsCompilation no

Author Wenjie Wang [aut, cre], Jun Yan [aut]

Maintainer Wenjie Wang <wenjie.2.wang@uconn.edu>

Repository CRAN

Date/Publication 2018-06-14 19:00:03 UTC

R topics documented:

`bSpline` *B-Spline Basis for Polynomial Splines*

Description

This function generates the B-spline basis matrix for a polynomial spline.

Usage

```
bSpline(x, df = NULL, knots = NULL, degree = 3L, intercept = FALSE,
        Boundary.knots = range(x, na.rm = TRUE), ...)
```

Arguments

`x`	The predictor variable. Missing values are allowed and will be returned as they were.
`df`	Degrees of freedom. One can specify `df` rather than `knots`, then the function chooses "df - degree" (minus one if there is an intercept) knots at suitable quantiles of `x` (which will ignore missing values). The default, `NULL`, corresponds to no inner knots, i.e., "degree - intercept". If `knots` was specified, `df` specified will be ignored.
`knots`	The internal breakpoints that define the spline. The default is `NULL`, which results in a basis for ordinary polynomial regression. Typical values are the mean or median for one knot, quantiles for more knots. See also `Boundary.knots`.
`degree`	Non-negative integer degree of the piecewise polynomial. The default value is 3 for cubic splines. Zero degree is allowed for this function, which is the only difference compared with `bs` in package `splines`.
`intercept`	If `TRUE`, an intercept is included in the basis; Default is `FALSE`.
`Boundary.knots`	Boundary points at which to anchor the B-spline basis. By default, they are the range of the non-`NA` data. If both `knots` and `Boundary.knots` are supplied, the basis parameters do not depend on `x`. Data can extend beyond `Boundary.knots`.
`...`	Optional arguments for future usage.

Details

It is an augmented function of bs in package splines for B-spline basis that allows piecewise constant (close on the left, open on the right) spline basis with zero degree. When the argument degree is greater than zero, it internally calls bs and generates a basis matrix for representing the family of piecewise polynomials with the specified interior knots and degree, evaluated at the values of x. The function has the same arguments with bs for ease usage.

Value

A matrix of dimension length(x) by df = degree + length(knots) (plus one if intercept is included). Attributes that correspond to the arguments specified are returned for usage of other functions in this package.

See Also

predict.bSpline2 for evaluation at given (new) values; dbs, deriv.bSpline2 for derivatives; ibs for integral of B-splines; mSpline for M-splines; iSpline for I-splines; cSpline for C-splines.

Examples

```
library(splines2)
x <- seq.int(0, 1, 0.01)
knots <- c(0.3, 0.5, 0.6)
bsMat <- bSpline(x, knots = knots, degree = 0, intercept = TRUE)

library(graphics)
matplot(x, bsMat, type = "l", ylab = "Piecewise constant B-spline bases")
abline(v = knots, lty = 2, col = "gray")
```

---

cSpline *C-Spline Basis for Polynomial Splines*

Description

This function generates the convex regression spline (called C-spline) basis matrix by integrating I-spline basis for a polynomial spline.

Usage

```
cSpline(x, df = NULL, knots = NULL, degree = 3L, intercept = FALSE,
        Boundary.knots = range(x, na.rm = TRUE), scale = TRUE, ...)
```

Arguments

x	The predictor variable. Missing values are allowed and will be returned as they were.
df	Degrees of freedom. One can specify df rather than knots, then the function chooses "df - degree" (minus one if there is an intercept) knots at suitable quantiles of x (which will ignore missing values). The default, NULL, corresponds to no inner knots, i.e., "degree - intercept".
knots	The internal breakpoints that define the spline. The default is NULL, which results in a basis for ordinary polynomial regression. Typical values are the mean or median for one knot, quantiles for more knots. See also Boundary.knots.
degree	Non-negative integer degree of the piecewise polynomial. The default value is 3 for cubic splines.
intercept	If TRUE, an intercept is included in the basis; Default is FALSE.
Boundary.knots	Boundary points at which to anchor the C-spline basis. By default, they are the range of the non-NA data. If both knots and Boundary.knots are supplied, the basis parameters do not depend on x. Data can extend beyond Boundary.knots.
scale	Logical value (TRUE by default) indicating whether scaling on C-spline basis is required. If TRUE, C-spline basis is scaled to have unit height at right boundary knot; the corresponding I-spline and M-spline basis matrices shipped in attributes are also scaled to the same extent.
...	Optional arguments for future usage.

Details

It is an implementation of the close form C-spline basis derived from the recursion formula of I-spline and M-spline. Internally, it calls iSpline and generates a basis matrix for representing the family of piecewise polynomials and their corresponding integrals with the specified interior knots and degree, evaluated at the values of x.

Value

A matrix of dimension length(x) by df = degree + length(knots) (plus on if intercept is included). The attributes that correspond to the arguments specified are returned for the usage of other functions in this package.

References

Meyer, M. C. (2008). Inference using shape-restricted regression splines. *The Annals of Applied Statistics*, 1013–1033. Chicago

See Also

predict.cSpline for evaluation at given (new) values; deriv.cSpline for derivatives; iSpline for I-splines; mSpline for M-splines.

Examples

```
library(splines2)
x <- seq.int(0, 1, 0.01)
knots <- c(0.3, 0.5, 0.6)

### when 'scale = TRUE' (by default)
csMat <- cSpline(x, knots = knots, degree = 2, intercept = TRUE)

library(graphics)
matplot(x, csMat, type = "l", ylab = "C-spline basis")
abline(v = knots, lty = 2, col = "gray")
isMat <- deriv(csMat)
msMat <- deriv(csMat, derivs = 2)
matplot(x, isMat, type = "l", ylab = "scaled I-spline basis")
matplot(x, msMat, type = "l", ylab = "scaled M-spline basis")

### when 'scale = FALSE'
csMat <- cSpline(x, knots = knots, degree = 2,
                 intercept = TRUE, scale = FALSE)
## the corresponding I-splines and M-splines (with same arguments)
isMat <- iSpline(x, knots = knots, degree = 2, intercept = TRUE)
msMat <- mSpline(x, knots = knots, degree = 2, intercept = TRUE)
## or using deriv methods (much more efficient)
isMat1 <- deriv(csMat)
msMat1 <- deriv(csMat, derivs = 2)
## equivalent
stopifnot(all.equal(isMat, isMat1, check.attributes = FALSE))
stopifnot(all.equal(msMat, msMat1, check.attributes = FALSE))
```

---

dbs *Derivative of B-Spline Basis for Polynomial Splines*

Description

This function produces the derivative of given order of B-splines. It is an implementation of the close form derivative of B-spline basis based on recursion relation. At knots, the derivative is defined to be the right derivative.

Usage

```
dbs(x, derivs = 1L, df = NULL, knots = NULL, degree = 3L,
    intercept = FALSE, Boundary.knots = range(x, na.rm = TRUE), ...)
```

Arguments

x	The predictor variable. Missing values are allowed and will be kept and returned as they were.
derivs	A positive integer specifying the order of derivative. By default, it is 1L for the first derivative.

df	Degrees of freedom of the B-spline basis to be differentiated. One can specify df rather than knots, then the function chooses "df - degree" (minus one if there is an intercept) knots at suitable quantiles of x (which will ignore missing values). The default, NULL, corresponds to no inner knots, i.e., "degree - intercept".
knots	The internal breakpoints that define the B-spline basis to be differentiated. The default is NULL, which results in a basis for ordinary polynomial regression. Typical values are the mean or median for one knot, quantiles for more knots. See also Boundary.knots.
degree	Non-negative integer degree of the piecewise polynomial to be differentiated. The default value is 3 for the integral of cubic B-splines.
intercept	If TRUE, an intercept is included in the basis; Default is FALSE.
Boundary.knots	Boundary points at which to anchor the B-spline basis to be differentiated. By default, they are the range of the non-NA data. If both knots and Boundary.knots are supplied, the basis parameters do not depend on x.
...	Optional arguments for future usage.

Details

The function is similar with splineDesign. However, it provides a more user-friendly interface, a more considerate NA's handling. Internally, it calls bSpline and generates a basis matrix for representing the family of piecewise polynomials and their corresponding derivative with the specified interior knots and degree, evaluated at the values of x. The function splineDesign in splines package can also be used to calculate derivative of B-splines.

Value

A matrix of dimension length(x) by df = degree + length(knots) (plus on if intercept is included). Attributes that correspond to the arguments specified are returned for usage of other functions in this package.

References

De Boor, Carl. (1978). *A practical guide to splines*. Vol. 27. New York: Springer-Verlag.

See Also

predict.dbs for evaluation at given (new) values; deriv.dbs for derivative method; bSpline for B-splines; ibs for integral of B-splines.

Examples

```
library(splines2)
x <- seq.int(0, 1, 0.01)
knots <- c(0.2, 0.4, 0.7)
## the second derivative of cubic B-splines with three internal knots
dMat <- dbs(x, derivs = 2L, knots = knots, intercept = TRUE)

## compare with the results from splineDesign
ord <- attr(dMat, "degree") + 1L
```

```
bKnots <- attr(dMat, "Boundary.knots")
aKnots <- c(rep(bKnots[1L], ord), knots, rep(bKnots[2L], ord))
res <- splines::splineDesign(aKnots, x = x, derivs = 2L)
stopifnot(all.equal(res, dMat, check.attributes = FALSE))
```

`deriv` *Derivative of Splines*

Description

`deriv` methods that obtain derivative of given order of B-splines, M-spline, I-splines, and C-splines, etc. At knots, the derivative is defined to be the right derivative. By default, the function returns the first derivative. For derivatives of order greater than one, the nested call such as `deriv(deriv(expr))` is supported but not recommended. For a better performance, argument `derivs` should be specified instead.

Usage

```
## S3 method for class 'bSpline2'
deriv(expr, derivs = 1L, ...)

## S3 method for class 'dbs'
deriv(expr, derivs = 1L, ...)

## S3 method for class 'ibs'
deriv(expr, derivs = 1L, ...)

## S3 method for class 'mSpline'
deriv(expr, derivs = 1L, ...)

## S3 method for class 'iSpline'
deriv(expr, derivs = 1L, ...)

## S3 method for class 'cSpline'
deriv(expr, derivs = 1L, ...)
```

Arguments

`expr`	Objects of class `bSpline2`, `ibs`, `dbs`, `mSpline`, `iSpline`, or `cSpline`, etc.
`derivs`	A positive integer specifying the order of derivatives. By default, it is `1L` for the first derivative.
`...`	Other arguments for further usage.

Details

The function is designed for most of the objects generated from this package. It internally extracts necessary information about the input spline basis matrix from its attributes. So the function will not work if some attribute is not available.

Value

A matrix of dimension `length(x)` by `df = degree + length(knots)` (plus one if intercept is included). Attributes that correspond to the arguments specified are returned for usage for other function in this package.

References

De Boor, Carl. (1978). *A practical guide to splines*. Vol. 27. New York: Springer-Verlag.

See Also

`bSpline` for B-splines; `ibs` for integral of B-splines; `mSpline` for M-splines; `iSpline` for I-splines; `cSpline` for C-splines.

Examples

```
library(splines2)
x <- c(seq.int(0, 1, 0.1), NA)  # NA's will be kept.
knots <- c(0.3, 0.5, 0.6)

## integal of B-splines and the corresponding B-splines integrated
ibsMat <- ibs(x, knots = knots)
bsMat <- bSpline(x, knots = knots)

## the first derivative
d1Mat <- deriv(ibsMat)
stopifnot(all.equal(bsMat, d1Mat, check.attributes = FALSE))

## the second derivative
d2Mat1 <- deriv(bsMat)
d2Mat2 <- deriv(ibsMat, derivs = 2L)
## nested calls are supported but not recommended
d2Mat3 <- deriv(deriv(ibsMat))
stopifnot(all.equal(d2Mat1, d2Mat2, d2Mat3, check.attributes = FALSE))

## C-splines, I-splines, M-splines and the derivatives
csMat <- cSpline(x, knots = knots, scale = FALSE)
isMat <- iSpline(x, knots = knots)
stopifnot(all.equal(isMat, deriv(csMat), check.attributes = FALSE))

msMat <- mSpline(x, knots = knots)
stopifnot(all.equal(msMat, deriv(isMat), deriv(csMat, 2),
                    deriv(deriv(csMat)), check.attributes = FALSE))

dmsMat <- mSpline(x, knots = knots, derivs = 1)
stopifnot(all.equal(dmsMat, deriv(msMat), deriv(isMat, 2),
                    deriv(deriv(isMat)), deriv(csMat, 3),
                    deriv(deriv(deriv(csMat))), check.attributes = FALSE))
```

ibs *Integral of B-Spline Basis for Polynomial Splines*

Description

This function generates the integral of B-spline basis matrix for a polynomial spline. The arguments are exactly the same with function bs in package splines.

Usage

```
ibs(x, df = NULL, knots = NULL, degree = 3, intercept = FALSE,
    Boundary.knots = range(x, na.rm = TRUE), ...)
```

Arguments

x	The predictor variable. Missing values are allowed and will be returned as they were.
df	Degrees of freedom of the B-spline basis to be integrated. One can specify df rather than knots, then the function chooses "df - degree" (minus one if there is an intercept) knots at suitable quantiles of x (which will ignore missing values). The default, NULL, corresponds to no inner knots, i.e., "degree - intercept".
knots	The internal breakpoints that define the B-spline basis to be integrated. The default is NULL, which results in a basis for ordinary polynomial regression. Typical values are the mean or median for one knot, quantiles for more knots. See also Boundary.knots.
degree	Non-negative integer degree of the piecewise polynomial to be integrated. The default value is 3 for the integral of cubic B-splines.
intercept	If TRUE, an intercept is included in the basis; Default is FALSE.
Boundary.knots	Boundary points at which to anchor the B-spline basis to be integrated. By default, they are the range of the non-NA data. If both knots and Boundary.knots are supplied, the basis parameters do not depend on x. Data can extend beyond Boundary.knots.
...	Optional arguments for future usage.

Details

It is an implementation of the close form integral of B-spline basis based on recursion relation. Internally, it calls bSpline and generates a basis matrix for representing the family of piecewise polynomials and their corresponding integrals with the specified interior knots and degree, evaluated at the values of x.

Value

A matrix of dimension length(x) by df = degree + length(knots) (plus on if intercept is included). Attributes that correspond to the arguments specified are returned for usage of other functions in this package.

References

De Boor, Carl. (1978). *A practical guide to splines*. Vol. 27. New York: Springer-Verlag.

See Also

`predict.ibs` for evaluation at given (new) values; `deriv.ibs` for derivative method. `bSpline` for B-splines; `dbs` for derivatives of B-splines;

Examples

```
library(splines2)
x <- seq.int(0, 1, 0.01)
knots <- c(0.2, 0.4, 0.7, 0.9)
ibsMat <- ibs(x, knots = knots, degree = 1, intercept = TRUE)

## the B-spline bases integrated by function bSpline (same arguments)
bsMat0 <- bSpline(x, knots = knots, degree = 1, intercept = TRUE)
## or by function deriv (recommended) that directly extracts the existing
## result from the attribute of ibsMat and thus is much more efficient.
bsMat <- deriv(ibsMat)
stopifnot(all.equal(bsMat0, bsMat, check.attributes = FALSE)) # equivalent

## plot B-spline basis with their corresponding integrals
library(graphics)
par(mfrow = c(1, 2))
matplot(x, bsMat, type = "l", ylab = "B-spline basis")
abline(v = knots, lty = 2, col = "gray")
matplot(x, ibsMat, type = "l", ylab = "Integral of B-spline basis")
abline(v = knots, lty = 2, col = "gray")
par(mfrow = c(1, 1))
```

---

iSpline *I-Spline Basis for Polynomial Splines or its derivatives*

Description

This function generates the I-spline (integral of M-spline) basis matrix for a polynomial spline or its derivatives of given order..

Usage

```
iSpline(x, df = NULL, knots = NULL, degree = 3L, intercept = FALSE,
        Boundary.knots = range(x, na.rm = TRUE), derivs = 0L, ...)
```

Arguments

x	The predictor variable. Missing values are allowed and will be returned as they were.
df	Degrees of freedom. One can specify df rather than knots, then the function chooses "df - degree" (minus one if there is an intercept) knots at suitable quantiles of x (which will ignore missing values). The default, NULL, corresponds to no inner knots, i.e., "degree - intercept".
knots	The internal breakpoints that define the spline. The default is NULL, which results in a basis for ordinary polynomial regression. Typical values are the mean or median for one knot, quantiles for more knots. See also Boundary.knots.
degree	Non-negative integer degree of the piecewise polynomial. The default value is 3 for cubic splines. Note that the degree of I-spline is defined to be the degree of the associated M-spline instead of actual polynomial degree. In other words, I-spline basis of degree 2 is defined as the integral of associated M-spline basis of degree 2.
intercept	If TRUE, an intercept is included in the basis; Default is FALSE.
Boundary.knots	Boundary points at which to anchor the I-spline basis. By default, they are the range of the non-NA data. If both knots and Boundary.knots are supplied, the basis parameters do not depend on x. Data can extend beyond Boundary.knots.
derivs	A non-negative integer specifying the order of derivatives of I-splines.
...	Optional arguments for future usage.

Details

It is an implementation of the close form I-spline basis based on the recursion formula of B-spline basis. Internally, it calls mSpline and bSpline, and generates a basis matrix for representing the family of piecewise polynomials and their corresponding integrals with the specified interior knots and degree, evaluated at the values of x.

Value

A matrix of dimension length(x) by df = degree + length(knots) (plus on if intercept is included). Attributes that correspond to the arguments specified are returned for usage of other functions in this package.

References

Ramsay, J. O. (1988). Monotone regression splines in action. *Statistical science*, 3(4), 425–441.

See Also

predict.iSpline for evaluation at given (new) values; deriv.iSpline for derivative method; mSpline for M-splines; cSpline for C-splines;

Examples

```
## Example given in the reference paper by Ramsay (1988)
library(splines2)
x <- seq.int(0, 1, by = 0.01)
knots <- c(0.3, 0.5, 0.6)
isMat <- iSpline(x, knots = knots, degree = 2, intercept = TRUE)

library(graphics)
matplot(x, isMat, type = "l", ylab = "I-spline basis")
abline(v = knots, lty = 2, col = "gray")

## the derivative of I-splines is M-spline
msMat1 <- iSpline(x, knots = knots, degree = 2, derivs = 1)
msMat2 <- mSpline(x, knots = knots, degree = 2)
stopifnot(all.equal(msMat1, msMat2))
```

---

mSpline *M-Spline Basis for Polynomial Splines and its Derivatives*

Description

This function generates the monotone regression spline (or simply called M-spline) basis matrix for a polynomial spline or its derivatives of given order.

Usage

```
mSpline(x, df = NULL, knots = NULL, degree = 3L, intercept = FALSE,
        Boundary.knots = range(x, na.rm = TRUE), derivs = 0L, ...)
```

Arguments

x	The predictor variable. Missing values are allowed and will be returned as they were.
df	Degrees of freedom. One can specify df rather than knots, then the function chooses "df - degree" (minus one if there is an intercept) knots at suitable quantiles of x (which will ignore missing values). The default, NULL, corresponds to no inner knots, i.e., "degree - intercept".
knots	The internal breakpoints that define the spline. The default is NULL, which results in a basis for ordinary polynomial regression. Typical values are the mean or median for one knot, quantiles for more knots. See also Boundary.knots.
degree	Non-negative integer degree of the piecewise polynomial. The default value is 3 for cubic splines. Zero degree is allowed for piecewise constant basis.
intercept	If TRUE, an intercept is included in the basis; Default is FALSE.
Boundary.knots	Boundary points at which to anchor the M-spline basis. By default, they are the range of the non-NA data. If both knots and Boundary.knots are supplied, the basis parameters do not depend on x. Data can extend beyond Boundary.knots.

derivs	A non-negative integer specifying the order of derivatives of M-splines. The default value is 0L for M-spline bases.
...	Optional arguments for future usage.

Details

It is an implementation of the close form M-spline basis based on relationship between M-spline basis and B-spline basis. In fact, M-spline basis is a rescaled version of B-spline basis. Internally, it calls function bSpline and generates a basis matrix for representing the family of piecewise polynomials with the specified interior knots and degree, evaluated at the values of x.

Value

A matrix of dimension length(x) by df = degree + length(knots) (plus one if intercept is included). Attributes that correspond to the arguments specified are returned for usage of other functions in this package.

References

Ramsay, J. O. (1988). Monotone regression splines in action. *Statistical science*, 3(4), 425–441.

See Also

predict.mSpline for evaluation at given (new) values; deriv.mSpline for derivative method; bSpline for B-splines; iSpline for I-splines; cSpline for C-splines.

Examples

```
## Example given in the reference paper by Ramsay (1988)
library(splines2)
x <- seq.int(0, 1, 0.01)
knots <- c(0.3, 0.5, 0.6)
msMat <- mSpline(x, knots = knots, degree = 2, intercept = TRUE)

library(graphics)
matplot(x, msMat, type = "l", ylab = "M-spline basis")
abline(v = knots, lty = 2, col = "gray")

## derivatives of M-splines
dmsMat <- mSpline(x, knots = knots, degree = 2,
                  intercept = TRUE, derivs = 1)
## or using the 'deriv' method
dmsMat1 <- deriv(msMat)
stopifnot(all.equal(dmsMat, dmsMat1, check.attributes = FALSE))
```

predict *Evaluate a Spline Basis*

Description

This function evaluates a predefined spline basis at (new) given values.

Usage

```
## S3 method for class 'bSpline2'
predict(object, newx, ...)

## S3 method for class 'ibs'
predict(object, newx, ...)

## S3 method for class 'dbs'
predict(object, newx, ...)

## S3 method for class 'mSpline'
predict(object, newx, ...)

## S3 method for class 'iSpline'
predict(object, newx, ...)

## S3 method for class 'cSpline'
predict(object, newx, ...)
```

Arguments

object	Objects of class bSpline2, ibs, mSpline, iSpline, or cSpline having attributes describing knots, degree, etc.
newx	The x values at which evaluations are required.
...	Optional argument for future usage.

Details

These are methods for the generic function predict for objects inheriting from class bSpline2, ibs, mSpline, iSpline, or cSpline. If newx is not given, the function returns the input object. For object returned by function cSpline, the mSpline and iSpline objects shipped in attributes should not be evaluated by this function if rescale is TRUE. See cSpline for details.

Value

An object just like the object input, except evaluated at the new values of x.

See Also

bSpline for B-splines; ibs for integral of B-splines; dbs for derivative of B-splines; mSpline for M-splines; iSpline for I-splines; cSpline for C-splines.

Examples

```
library(splines2)
x <- seq.int(0, 1, 0.2)
knots <- c(0.3, 0.5, 0.6)
newX <- seq.int(0.1, 0.9, 0.2)

## for B-splines
bsMat <- bSpline(x, knots = knots, degree = 2)
predict(bsMat, newX)

## for integral of B-splines
ibsMat <- ibs(x, knots = knots, degree = 2)
predict(ibsMat, newX)

## for derivative of B-splines
dbsMat <- dbs(x, knots = knots, degree = 2)
predict(dbsMat, newX)

## for M-spline
msMat <- mSpline(x, knots = knots, degree = 2)
predict(msMat, newX)

## for I-spline
isMat <- iSpline(x, knots = knots, degree = 2)
predict(isMat, newX)

## for C-spline
csMat <- cSpline(x, knots = knots, degree = 2)
predict(csMat, newX)
```

---

print *Print Out a Spline Basis Matrix*

Description

Print methods that simply print out the spline basis matrix without unnecessary attributes.

Usage

```
## S3 method for class 'bSpline2'
print(x, ...)

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

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

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

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

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

Arguments

x	Objects of class bSpline2, ibs, mSpline, iSpline, or cSpline, etc.
...	Optional argument for future usage.

Value

Object input.

---

splines2 *splines2: Regression Spline Functions and Classes*

Description

A supplementary package on splines providing functions constructing B-splines, integral of B-splines, monotone splines (M-splines) and its integral (I-splines), convex splines (C-splines), and their derivatives of given order. Piecewise constant basis is allowed for B-spline and M-spline basis.

Details

It is named after the package **splines**: "Regression Spline Functions and Classes". The tailing number two is simply "too" (and by no means for the generation two).

Index