

Package ‘splines’

October 12, 2016

Version 3.3.1

Priority base

Imports graphics, stats

Title Regression Spline Functions and Classes

Author Douglas M. Bates <bates@stat.wisc.edu> and
William N. Venables <Bill.Venables@csiro.au>

Maintainer R Core Team <R-core@r-project.org>

Description Regression spline functions and classes.

License Part of R 3.3.1

Suggests Matrix, methods

R topics documented:

splines-package	2
asVector	2
backSpline	3
bs	4
interpSpline	5
ns	6
periodicSpline	7
polySpline	8
predict.bs	9
predict.bSpline	10
splineDesign	11
splineKnots	13
splineOrder	13
xyVector	14
Index	15

splines-package 回帰スプライン関数とクラス

Description

回帰スプライン関数とクラス.

Details

このパッケージは B スプラインベースの `bs` と自然3次スプラインベースの `ns` を使った回帰スプラインを用いた作業のための関数を提供する.

関数の完全なリストに対しては `library(help = "splines")` を使う.

Author(s)

Douglas M. Bates <bates@stat.wisc.edu> と William N. Venables <Bill.Venables@csiro.au>
管理者 : R Core Team <R-core@r-project.org>

asVector オブジェクトをベクトルへ強制変換する

Description

これは総称的関数である. この関数に対するメソッドは与えられたクラスのオブジェクトをベクトルに強制変換する.

Usage

```
asVector(object)
```

Arguments

`object` オブジェクト.

Details

新しいクラスでのベクトルへの強制変換メソッドは `as.vector` 関数ではなく `asVector` 総称的関数に対して作られなければならない. `as.vector` 関数は内部的であり簡単には拡張できない. 現在 `asVector` メソッドを持つ唯一のクラスは `xyVector` である.

Value

ベクトル.

Author(s)

Douglas Bates と Bill Venables

See Also[xyVector](#)**Examples**

```
require(stats)
ispl <- interpSpline( weight ~ height, women )
pred <- predict(ispl)
class(pred)
utils::str(pred)
asVector(pred)
```

backSpline	単調逆スプライン
------------	----------

Description

単調な自然スプラインの単調な逆を作る。

Usage

```
backSpline(object)
```

Arguments

object クラス `nbSpline` または `npolySpline` を継承するオブジェクト。つまり、オブジェクトは自然な補間スプラインを表現しなければならないが、B スプライン表現でも区分的に線形な多項式表現でも良い。スプラインはそれが単調かどうかチェックされる。

Value

クラス `polySpline` のオブジェクトで `object` で表されるスプラインの逆関数になるようなノット位置で適当な値と微分係数を持つ関数の区分的多項式関数表現を持つ。技術的にはこのオブジェクトはノット位置で2階微分係数が連続になるように拘束されていないためにスプラインではない。しかしながらこれはしばしば y/x 対への補間スプラインよりもかなり良い近似を与える。

Author(s)

Douglas Bates と Bill Venables

See Also[interpSpline](#)**Examples**

```
require(graphics)
ispl <- interpSpline( women$height, women$weight )
bspl <- backSpline( ispl )
plot( bspl ) # ノットの範囲上でプロット
points( women$weight, women$height )
```

bs 多項式スプラインに対する B スプラインのベース

Description

多項式スプラインに対する B スプラインベース行列を生成する。

Usage

```
bs(x, df = NULL, knots = NULL, degree = 3, intercept = FALSE,
   Boundary.knots = range(x))
```

Arguments

x	予測変数。欠損値が許される。
df	自由度: knots ではなく df を指定できる; bs() はそうすると df-degree 個のノット (切片があればマイナス 1) を x の適当なクォンタイル位置に設定する (欠損値は無視される)。既定の NULL は内部ノット無しに対応する, つまり degree - intercept.
knots	スプラインを定義する内部的な分点。既定値は NULL で, 通常 of 多項式回帰に対するベースに帰着する。典型的な値は一つのノットに対しては平均かメディアンで, より多くのノットに対してはクォンタイルである。Boundary.knots も見よ。
degree	区分的な多項式の次数—既定ではキュービックスプラインに対しては 3.
intercept	もし TRUE なら切片項がベースに含められる; 既定では FALSE.
Boundary.knots	B スプラインのベースを支える境界点 (既定では NA でないデータの範囲)。もし knots と Boundary.knots の両方が与えられると, ベースのパラメータは x に依存する。データは Boundary.knots を超えて伸びても良い。

Details

bs は関数 [spline.des](#) に依存している。これは指定された内部ノットと次数を持ち, x の値で評価された区分的な多項式族の表現のためのベースの行列を生成する。主要な使用目的はモデル中の区分的な多項式を直接に指定する公式のモデリングにある。

Boundary.knots が range(x) の内部に設定された時は, bs() は今や微分係数に対して重要なそれぞれの境界ノット内部で‘ピボット’を使う。R のバージョン ≤ 3.2.2 では, 境界ノット自体がピボットとして使われてきたが, これは少々良くない補外に繋がる。

Value

次元 c(length(x), df) の行列で, ここで df が提供されるか, またはもし knots が提供されるともし切片があれば df = length(knots) + degree プラス 1. bs への引数に対応する属性が返され, そして predict.bs() で使うために knots, Boundary.knots 等を明示的に与える。

Author(s)

Douglas Bates と Bill Venables. R Core により微修正され, Boundary.knots の“外側”への補外への修正パッチは Trevor Hastie による

References

Hastie, T. J. (1992) Generalized additive models. Chapter 7 of *Statistical Models in S* eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.

See Also

[ns](#), [poly](#), [smooth.spline](#), [predict.bs](#), [SafePrediction](#)

Examples

```
require(stats); require(graphics)
bs(women$height, df = 5)
summary(fm1 <- lm(weight ~ bs(height, df = 5), data = women))
## 安全な予測の例
plot(women, xlab = "Height (in)", ylab = "Weight (lb)")
ht <- seq(57, 73, length.out = 200)
lines(ht, predict(fm1, data.frame(height = ht)))
```

interpSpline	補間スプラインを作る
--------------	------------

Description

x と y ベクトル(default メソッド)からか、組み合わせ formula / data.frame (formula メソッド) から補間スプラインを作る。

Usage

```
interpSpline(obj1, obj2, bSpline = FALSE, period = NULL,
             na.action = na.fail, sparse = FALSE)
```

Arguments

obj1	x 値の数値ベクトルか公式.
obj2	もし obj1 が数値ならばこれは同じ長さの数値ベクトルでなければならない. もし obj1 が公式ならばこれは公式中の名前を評価するオプションのデータフレームとすることが出来る.
bSpline	もし TRUE ならば b スプライン表現が返され, さもなければ区分的多項式表現が返される. 既定では FALSE.
period	周期的な補間スプラインに対する周期を与えるオプションの正数.
na.action	データが NA を含む時何が起こるべきかを指示するオプション関数. 既定のアクション (na.omit) は全ての不完全な観測値を取り除く. もうひとつのアクション na.fail は不完全データがあれば interpSpline にエラーメッセージを出させ停止する.
sparse	基礎にある splineDesign に渡される論理値. もし真ならば, メモリを節約し数百以上の点があればより速くなる.

Value

(S3) クラス `spline` を継承するオブジェクト。オブジェクトは B スプライン表現であり得、その場合はそれは自然 B スプラインに対するクラス `nbSpline` になる。または区分的多項式表現になり、その場合それはクラス `npolySpline` になる。

Author(s)

Douglas Bates と Bill Venables

See Also

[splineKnots](#), [splineOrder](#), [periodicSpline](#).

Examples

```
require(graphics); require(stats)
ispl <- interpSpline( women$height, women$weight )
ispl2 <- interpSpline( weight ~ height, women )
# ispl と ispl2 は同じでなければならない
plot( predict( ispl, seq( 55, 75, length.out = 51 ) ), type = "l" )
points( women$height, women$weight )
plot( ispl ) # ノットの範囲上でプロット
points( women$height, women$weight )
splineKnots( ispl )
```

ns	自然な3次スプラインに対するベース行列を生成する
----	--------------------------

Description

自然な3次スプラインに対する B スプラインのベース行列を生成する。

Usage

```
ns(x, df = NULL, knots = NULL, intercept = FALSE,
   Boundary.knots = range(x))
```

Arguments

x	予測変数。欠損値が許される。
df	自由度。ノットの代わりに df を与えることができる；そうすると <code>ns()</code> は <code>df - 1 - intercept</code> 個のノットを x の(欠損値は無視される)適当に選ばれたクォンタイルに選ぶ。既定の <code>df = 1</code> はノット無しに対応する。
knots	スプラインを定義する分割点。既定ではノットはない；自然な境界条件と共にこれの結果は x に関する線形回帰に対するベースになる。典型的な値は一つのノットに対しては平均かメディアンで、複数のノットに対してはクォンタイル。 <code>Boundary.knots</code> も見よ。
intercept	もし TRUE ならば、切片項がベース中に含まれる；既定では FALSE。

`Boundary.knots` そこに自然な境界条件を課し B スプラインのベースを置く境界点(既定ではデータの範囲). もし `knots` と `Boundary.knots` が両方提供されると, ベースのパラメータは x に依存しない. データは `Boundary.knots` を超えて広がっても良い.

Details

`ns` は関数 `spline.des` に基づく. これは指定された内部ノットと自然な境界条件を持つ区分的3次スプライン族に対するベース行列を生成する. これらは関数が境界ノットを超えると線形であるという制約を課し. 境界ノットは与えることもでき, 既定ではデータの極値になる.

主要な用法はモデリング公式中でモデル中の自然スプライン項を直接指定することである.

Value

次元 $\text{length}(x) * \text{df}$ の行列で, ここで `df` は指定されたか, もし `knots` が提供されていると $\text{df} = \text{length}(\text{knots}) + 1 + \text{intercept}$ となる. 引数 `ns` に対応し `predict.ns()` で使用される `knots`, `Boundary.knots` 等が明示的に与えられる属性が返される.

References

Hastie, T. J. (1992) Generalized additive models. Chapter 7 of *Statistical Models in S* eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.

See Also

[bs](#), [predict.ns](#), [SafePrediction](#)

Examples

```
require(stats); require(graphics)
ns(women$height, df = 5)
summary(fm1 <- lm(weight ~ ns(height, df = 5), data = women))

## どのノットが選ばれたかを見るには
attr(terms(fm1), "predvars")

## 安全な予測の例
plot(women, xlab = "Height (in)", ylab = "Weight (lb)")
ht <- seq(57, 73, length.out = 200)
lines(ht, predict(fm1, data.frame(height = ht)))
```

periodicSpline

周期的な補間スプラインを作る

Description

x と y ベクトルからか, または公式/データフレームの組み合わせから周期的な補間スプラインを作る.

Usage

```
periodicSpline(obj1, obj2, knots, period = 2*pi, ord = 4)
```

Arguments

obj1 x 値の数値ベクトル化公式.

obj2 もし obj1 が数値ならばこれは同じ長さの数値ベクトルでなければなら
ない. もし obj1 が公式ならばこれは公式中の名前をその中で評価する
オプションのデータフレームに取ることが出来る.

knots ノット位置のオプション数値ベクトル.

period 周期的スプラインに対する酒器を与える正数値. 既定では $2 * \pi$.

ord スプラインの次数を与える整数で, 少なくとも 2. 既定では 4. スプ
ラインの次数の定義については [splineOrder](#) を見よ.

Value

クラス `spline` を継承するオブジェクト. オブジェクトは B スプライン表現になり
得, その場合それは `pbSpline` オブジェクト. または区分的多項式表現も可能である
(`ppolySpline` オブジェクト).

Author(s)

Douglas Bates と Bill Venables

See Also

[splineKnots](#), [interpSpline](#)

Examples

```
require(graphics); require(stats)
xx <- seq( -pi, pi, length.out = 16 )[-1]
yy <- sin( xx )
frm <- data.frame( xx, yy )
pispl <- periodicSpline( xx, yy, period = 2 * pi )
pispl
pispl2 <- periodicSpline( yy ~ xx, frm, period = 2 * pi )
stopifnot(all.equal(pispl, pispl2)) # pispl と pispl2 は同じ

plot( pispl )                    # 1周期について表示
points( yy ~ xx, col = "brown" )
plot( predict( pispl, seq(-3*pi, 3*pi, length.out = 101) ), type = "l" )
```

polySpline

スプラインの区分的多項式表現

Description

スプラインオブジェクトの区分的多項式表現を作る.

Usage

```
polySpline(object, ...)
as.polySpline(object, ...)
```

Arguments

object クラス spline を継承するオブジェクト。
 ... オプションの追加引数。現在追加引数は使われない。

Value

クラス polySpline を継承するオブジェクト。これは一変量スプライン関数の区分的多項式表現である。これはノットと呼ばれる異なった数値のセットにより定義される。スプライン関数はノットの隣り合った各対の間で多項式である。各内部ノットに於いて両端の多項式は関数値と幾つかのその微分係数が一致する様に拘束される。

Author(s)

Douglas Bates と Bill Venables

See Also

[interpSpline](#), [periodicSpline](#), [splineKnots](#), [splineOrder](#)

Examples

```
require(graphics)
ispl <- polySpline(interpSpline( weight ~ height, women, bSpline = TRUE))
print( ispl ) # 区分的多項式表現をプリント
plot( ispl ) # ノットの範囲上でプロット
points( women$height, women$weight )
```

predict.bs

スプラインのベースを評価する

Description

予め定義されたスプラインのベースを与えられた値で評価する。

Usage

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

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

Arguments

object knots, degree 等を記述する属性を持つ bs か ns への呼び出しの結果。
 newx そこで評価が求められている x 値。
 ... オプションの追加引数。現在追加引数は使われない。

Value

丁度 object のようなオブジェクトであるが、新しい x の値で評価されている。

これらは "bs" か "ns" を継承するオブジェクトに対する総称的関数 `predict` に対するメソッドである。この関数の一般的挙動については `predict` を見よ。

See Also

[bs](#), [ns](#), [poly](#).

Examples

```
require(stats)
basis <- ns(women$height, df = 5)
newX <- seq(58, 72, length.out = 51)
# ベースを新しいデータに於いて評価
predict(basis, newX)
```

predict.bSpline

新しい x 値に対してスプラインを評価する

Description

仮想クラス `bSpline` と `polySpline` を継承するクラスに対する `predict` メソッドはスプラインやその微分係数を評価するために使われる。スプラインオブジェクトに対する `plot` メソッドは先ず `predict` を欠損した引数 x で評価し、それから結果の `xyVector` を `type = "l"` でプロットする。

Usage

```
## S3 method for class 'bSpline'
predict(object, x, nseg = 50, deriv = 0, ...)
## S3 method for class 'nbSpline'
predict(object, x, nseg = 50, deriv = 0, ...)
## S3 method for class 'pbSpline'
predict(object, x, nseg = 50, deriv = 0, ...)
## S3 method for class 'npolySpline'
predict(object, x, nseg = 50, deriv = 0, ...)
## S3 method for class 'ppolySpline'
predict(object, x, nseg = 50, deriv = 0, ...)
```

Arguments

<code>object</code>	<code>bSpline</code> や <code>polySpline</code> クラスを継承するオブジェクト。
<code>x</code>	そこでスプラインを評価する x 値の数値ベクトル。もしこの値が欠損していると x の適当な値がノットの範囲を張る <code>nseg</code> 個の分割列として生成される。
<code>nseg</code>	<code>object</code> 中のノットの範囲を張る等間隔の分割を与える正整数。この値は x が欠損している場合にだけ使われる。
<code>deriv</code>	評価される微分係数を指定する 0 と <code>splineOrder(object) - 1</code> の間の整数。
<code>...</code>	他のメソッドへ・から渡される追加引数。

Value

次の成分を持つ xyVector

x	提供されたか推測された x 値
y	スプラインの値(または x 値におけるその deriv 階の微分係数)のベクトル

Author(s)

Douglas Bates と Bill Venables

See Also

[xyVector](#), [interpSpline](#), [periodicSpline](#)

Examples

```
require(graphics); require(stats)
ispl <- interpSpline( weight ~ height, women )
opar <- par(mfrow = c(2, 2), las = 1)
plot(predict(ispl, nseg = 201),      # plots over the range of the knots
      main = "Original data with interpolating spline", type = "l",
      xlab = "height", ylab = "weight")
points(women$height, women$weight, col = 4)
plot(predict(ispl, nseg = 201, deriv = 1),
      main = "First derivative of interpolating spline", type = "l",
      xlab = "height", ylab = "weight")
plot(predict(ispl, nseg = 201, deriv = 2),
      main = "Second derivative of interpolating spline", type = "l",
      xlab = "height", ylab = "weight")
plot(predict(ispl, nseg = 401, deriv = 3),
      main = "Third derivative of interpolating spline", type = "l",
      xlab = "height", ylab = "weight")
par(opar)
```

splineDesign

B スプラインに対する計画行列

Description

x 中の値の knots で定義された B スプラインに対する計画行列を評価する。

Usage

```
splineDesign(knots, x, ord = 4, derivs, outer.ok = FALSE,
             sparse = FALSE)
spline.des  (knots, x, ord = 4, derivs, outer.ok = FALSE,
             sparse = FALSE)
```

Arguments

knots	ノット位置の数値ベクトル(もし必要なら増加順にソートされる).
x	Bスプラインや微分係数をそこで評価する値の数値ベクトル. outer.ok が真でない限り, x 中の値は“内部”ノット knots[ord] と knots[length(knots) - (ord-1)] の間になければならない.
ord	スプライン関数の次数を与える正整数. これは区分的な各多項式中の係数の数であり, 従って3次のスプラインは次数4を持つ. 既定では4.
derivs	0 と ord - 1 の間の値を持つ整数値ベクトルであり, 概念的には x の長さまでリサイクルされる. 与えられた階数の微分係数は x 位置で評価される. 既定ではゼロである(または x と同じ長さのゼロのベクトル).
outer.ok	x が内部ノットの外側にも許されるかどうかを指示する論理値, x 引数を見よ.
sparse	結果がクラス "sparseMatrix" (パッケージ Matrix からの) を継承するべきかどうかを指示する論理値.

Value

length(x) 行と length(knots) - ord 列を持つ行列. 行列の i 番目の行は knot ベクトルで定義され x の i 番目の値で評価された B スプラインの係数(またはBスプラインの指示された微分係数)を含む. 各 B スプラインは ord 個の連続するノットのセットで定義されるので B スプラインの総数は length(knots) - ord となる.

Note

古い spline.des 関数は同じ引数を持つが knots, ord, derivs そして design を含む幾つかの成分を持つリストを返す. design 成分は splineDesign 関数の値と同じである.

Author(s)

Douglas Bates と Bill Venables

Examples

```
require(graphics)
splineDesign(knots = 1:10, x = 4:7)
splineDesign(knots = 1:10, x = 4:7, deriv = 1)
## バンド構造を視覚化
Matrix::drop0(zapsmall(6*splineDesign(knots = 1:40, x = 4:37, sparse = TRUE)))

knots <- c(1,1.8,3:5,6.5,7,8.1,9.2,10) # 10 => 10-4 = 6 ベースのスプライン
x <- seq(min(knots)-1, max(knots)+1, length.out = 501)
bb <- splineDesign(knots, x = x, outer.ok = TRUE)

plot(range(x), c(0,1), type = "n", xlab = "x", ylab = "",
      main = "B-splines - sum to 1 inside inner knots")
mtext(expression(B[j](x) * and "* sum(B[j](x), j == 1, 6)), adj = 0)
abline(v = knots, lty = 3, col = "light gray")
abline(v = knots[c(4,length(knots)-3)], lty = 3, col = "gray10")
lines(x, rowSums(bb), col = "gray", lwd = 2)
matlines(x, bb, ylim = c(0,1), lty = 1)
```

splineKnots	スプラインのノットのベクトル
-------------	----------------

Description

スプラインオブジェクトに対応するノットのベクトルを返す.

Usage

```
splineKnots(object)
```

Arguments

object クラス "spline" を継承するオブジェクト.

Value

ノット位置の非減少数値ベクトル.

Author(s)

Douglas Bates と Bill Venables

Examples

```
ispl <- interpSpline( weight ~ height, women )
splineKnots( ispl )
```

splineOrder	スプラインの次数の決定
-------------	-------------

Description

スプラインオブジェクトの次数を返す.

Usage

```
splineOrder(object)
```

Arguments

object クラス "spline" を継承するオブジェクト.

Details

スプラインの次数は区分的多項式表現の各多項式の係数の数である. 従って3次のスプラインは次数4を持つ.

Value

正整数.

Author(s)

Douglas Bates と Bill Venables

See Also[splineKnots](#), [interpSpline](#), [periodicSpline](#)**Examples**

```
splineOrder( interpSpline( weight ~ height, women ) )
```

xyVector	<i>list("xyVector")</i> オブジェクトを作る
----------	-----------------------------------

Description

x-y のペアのセットを表すオブジェクトを作る。結果のオブジェクトは行列としてもデータフレームとしてもそしてベクトルとしても扱うことが出来る。ベクトルとして扱われる場合はこれは y 成分だけに帰着する。

`predict.spline` のような関数の結果は `xyVector` オブジェクトとして返されるので y 位置を生成した x 値が例えばプロットを生成する目的のために保持される。

Usage

```
xyVector(x, y)
```

Arguments

x	数値ベクトル
y	x と同じ長さの数値ベクトル

Value

クラス `xyVector` のオブジェクトで次の成分を持つ

x	数値ベクトル
y	x と同じ長さの数値ベクトル

Author(s)

Douglas Bates と Bill Venables

Examples

```
require(stats); require(graphics)
ispl <- interpSpline( weight ~ height, women )
weights <- predict( ispl, seq( 55, 75, length.out = 51 ) )
class( weights )
plot( weights, type = "l", xlab = "height", ylab = "weight" )
points( women$height, women$weight )
weights
```

Index

*Topic **models**

- asVector, 2
- backSpline, 3
- interpSpline, 5
- periodicSpline, 7
- polySpline, 8
- predict.bSpline, 10
- splineDesign, 11
- splineKnots, 13
- splineOrder, 13
- splines-package, 2
- xyVector, 14

*Topic **package**

- splines-package, 2

*Topic **smooth**

- bs, 4
- ns, 6
- predict.bs, 9

- as.polySpline (polySpline), 8
- asVector, 2

- backSpline, 3
- bs, 2, 4, 7, 9, 10

- interpSpline, 3, 5, 8, 9, 11, 14

- NA, 4
- ns, 2, 5, 6, 9, 10

- periodicSpline, 6, 7, 9, 11, 14
- poly, 5, 10
- polySpline, 8
- predict, 10
- predict.bs, 5, 9
- predict.bSpline, 10
- predict.nbSpline (predict.bSpline), 10
- predict.npolySpline (predict.bSpline), 10
- predict.ns, 7
- predict.ns (predict.bs), 9
- predict.pbSpline (predict.bSpline), 10
- predict.ppolySpline (predict.bSpline), 10

- SafePrediction, 5, 7
- smooth.spline, 5
- sparseMatrix, 12
- spline.des, 4, 7
- spline.des (splineDesign), 11
- splineDesign, 5, 11
- splineKnots, 6, 8, 9, 13, 14
- splineOrder, 6, 8, 9, 13
- splines (splines-package), 2
- splines-package, 2
- xyVector, 3, 11, 14