Package 'lmeSplines'

May 8, 2026

Version 1.1.20

Title Add Smoothing Spline Modelling Capability to `nlme`

Description Adds smoothing spline modelling capability to nlme. Fits smoothing spline terms in Gaussian linear and nonlinear mixed-effects models.

Depends R (>= 3.5.0), nlme(>= 3.1-29)

LazyData true

Encoding UTF-8

RoxygenNote 7.3.2

License GPL (>= 2)

URL https://github.com/agalecki/lmeSplines

NeedsCompilation no

Author Rod Ball [aut],
Andrzej T. Galecki [cre] (ORCID:
<https://orcid.org/0000-0003-1542-4001>)

Maintainer Andrzej T. Galecki <agalecki@umich.edu>

Repository CRAN

Date/Publication 2025-08-31 15:10:02 UTC

Contents

approx.Z *Interpolating in Smoothing Spline Z-Matrix Columns*

Description

Interpolates the Z-matrix for LME smoothing spline fits from one set of time covariate values to another using linear interpolation of each column of the Z-matrix, regarded as a function of time.

Usage

```
approx.Z(Z, oldtimes, newtimes)
```

Arguments

Z	Z-matrix with rows corresponding to the sorted unique values of the time covariate (e.g., from smspline or smspline.v).
oldtimes	Numeric vector of original (sorted) time covariate values corresponding to the rows of Z.
newtimes	Numeric vector of new time covariate values to interpolate to.

Value

A matrix with the same number of columns as Z and rows corresponding to newtimes, containing the interpolated Z-matrix values. This can be used with smspline for fitting LME splines with random effects at different time points or as part of the newdata argument in predict.lme for predictions at new points.

Note

Linear interpolation works well because the spline basis functions are approximately piecewise linear.

Author(s)

Rod Ball <rod.ball@scionresearch.com>

See Also

smspline, lme, predict.lme

Examples

```
times1 <- 1:10
Zt1 <- smspline(~ times1)
times2 <- seq(1, 10, by = 0.1)
Zt2 <- approx.Z(Zt1, oldtimes = times1, newtimes = times2)
```

smspline *Smoothing Splines in NLME*

Description

Functions to generate matrices for a smoothing spline covariance structure, enabling the fitting of smoothing spline terms in linear mixed-effects models (LME) or nonlinear mixed-effects models (NLME). A smoothing spline can be represented as a mixed model, as described by Speed (1991) and Verbyla (1999). The generated Z-matrix can be incorporated into a data frame and used in LME random effects terms with an identity covariance structure (`pdIdent(~Z - 1)`).

The model formulation for a spline in time (`t`) is:

$$y = X_s\beta_s + Z_s u_s + e$$

where $X_s = [1|t]$, $Z_s = Q(t(Q)Q)^{-1}$, and $u_s \sim N(0, G_s)$ is a set of random effects. The random effects are transformed to independence via $u_s = Lv_s$, where $v_s \sim N(0, I\sigma_s^2)$ and L is the lower triangle of the Cholesky decomposition of G_s. The Z-matrix is transformed to $Z = Z_s L$.

Usage

```
smspline(formula, data)

smspline.v(time)
```

Arguments

`formula`	Model formula with the right-hand side specifying the spline covariate (e.g., `~ time`). Must contain exactly one variable.
`data`	Optional data frame containing the variable specified in `formula`. If not provided, the formula is evaluated in the current environment.
`time`	Numeric vector of spline time covariate values to smooth over.

Value

For `smspline`, a Z-matrix with the same number of rows as the input data frame or vector, representing the random effects design matrix for the smoothing spline. After fitting an LME model, the standard deviation parameter for the random effects estimates σ_s, and the smoothing parameter is $\lambda = \sigma^2/\sigma_s^2$.

For `smspline.v`, a list containing:

Xs Matrix for fixed effects, with columns `[1 | t]`.

Zs Matrix for random effects, computed as `Q (t(Q) %*% Q)^-1 L`.

Q Matrix used in the spline formulation.

Gs Covariance matrix for the random effects.

R Cholesky factor of `Gs`.

Note

The time points for the smoothing spline basis are, by default, the unique values of the time covariate. Model predictions at the fitted data points can be obtained using `predict.lme`. For predictions at different time points, use `approx.Z` to interpolate the Z-matrix.

Author(s)

Rod Ball <rod.ball@scionresearch.com>

References

Pinheiro, J. and Bates, D. (2000) *Mixed-Effects Models in S and S-PLUS*. Springer-Verlag, New York.

Speed, T. (1991) Discussion of "That BLUP is a good thing: the estimation of random effects" by G. Robinson. *Statistical Science*, 6, 42–44.

Verbyla, A. (1999) *Mixed Models for Practitioners*. Biometrics SA, Adelaide.

See Also

`approx.Z`, `lme`

Examples

```
# Smoothing spline curve fit
data(smSplineEx1)
smSplineEx1$all <- rep(1, nrow(smSplineEx1))
smSplineEx1$Zt <- smspline(~ time, data = smSplineEx1)
fit1s <- lme(y ~ time, data = smSplineEx1,
             random = list(all = pdIdent(~ Zt - 1)))
summary(fit1s)
plot(smSplineEx1$time, smSplineEx1$y, pch = "o", type = "n",
     main = "Spline fits: lme(y ~ time, random = list(all = pdIdent(~ Zt - 1)))",
     xlab = "time", ylab = "y")
points(smSplineEx1$time, smSplineEx1$y, col = 1)
lines(smSplineEx1$time, smSplineEx1$y.true, col = 1)
lines(smSplineEx1$time, fitted(fit1s), col = 2)

# Fit model with reduced number of spline points
times20 <- seq(1, 100, length = 20)
Zt20 <- smspline(times20)
smSplineEx1$Zt20 <- approx.Z(Zt20, times20, smSplineEx1$time)
fit1s20 <- lme(y ~ time, data = smSplineEx1,
               random = list(all = pdIdent(~ Zt20 - 1)))
anova(fit1s, fit1s20)
summary(fit1s20)

# Model predictions on a finer grid
times200 <- seq(1, 100, by = 0.5)
pred.df <- data.frame(all = rep(1, length(times200)), time = times200)
pred.df$Zt20 <- approx.Z(Zt20, times20, times200)
```

```
yp20.200 <- predict(fit1s20, newdata = pred.df)
lines(times200, yp20.200 + 0.02, col = 4)

# Mixed model spline terms at multiple levels of grouping
data(Spruce)
Spruce$Zday <- smspline(~ days, data = Spruce)
Spruce$all <- rep(1, nrow(Spruce))
spruce.fit1 <- lme(logSize ~ days, data = Spruce,
                   random = list(all = pdIdent(~ Zday - 1),
                                 plot = ~ 1, Tree = ~ 1))
spruce.fit2 <- lme(logSize ~ days, data = Spruce,
                   random = list(all = pdIdent(~ Zday - 1),
                                 plot = pdBlocked(list(~ days, pdIdent(~ Zday - 1))),
                                 Tree = ~ 1))
anova(spruce.fit1, spruce.fit2)
summary(spruce.fit1)
```

`smSplineEx1` *Simulated Data for Smoothing Spline Curve Fitting*

Description

Simulated dataset to demonstrate smoothing spline curve fitting with `smspline` and `lme`. The data consists of 100 observations simulated around the curve $y = 10 - 6\exp(-4t/100)$ with independent normal random errors (standard deviation = 1).

Simulated dataset to demonstrate smoothing spline curve fitting with `smspline` and `lme`. The data consists of 100 observations simulated around the curve $y = 10 - 6\exp(-4t/100)$ with independent normal random errors (standard deviation = 1).

Usage

```
smSplineEx1

smSplineEx1
```

Format

A data frame with 100 rows and 3 variables:

time Time covariate.

y Simulated response values.

y.true True response values.

A data frame with 100 rows and 3 variables:

time Time covariate.

y Simulated response values.

y.true True response values.

Source

Generated by Rod Ball for the `lmeSplines` package.

Examples

```
data(smSplineEx1)
str(smSplineEx1)

data(smSplineEx1)
str(smSplineEx1)
plot(smSplineEx1$time, smSplineEx1$y, main = "Simulated Data", xlab = "Time", ylab = "y")
lines(smSplineEx1$time, smSplineEx1$y.true, col = "blue")
```

Index