Package 'splinemixmeta'

April 8, 2026

Title Additive Mixed Meta-Analysis with Spline Meta-Regression

Version 1.0.1

Description Fit additive mixed meta-analysis (AMMA) models, extending the 'mixmeta' package <https://cran.r-project.org/package=mixmeta> to allow for spline-based meta-regression. Functions combine features of 'mgcv' <https://cran.r-project.org/package=mgcv> for building spline components and 'mixmeta' for estimating general mixed-effects meta-analysis models.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

URL <https://github.com/perrydv/splinemixmeta>

BugReports https://github.com/perrydv/splinemixmeta/issues

Depends R (>= 4.1)

Imports ggplot2, mgcv, mixmeta

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Perry de Valpine [aut] (ORCID: <https://orcid.org/0000-0002-8329-6796>), Marcus Beck [aut, cre] (ORCID: <https://orcid.org/0000-0002-4996-0059>)

Maintainer Marcus Beck <mbeck@tbep.org>

Repository CRAN

Date/Publication 2026-04-08 09:40:02 UTC

Contents

`blup.splinemixmeta` *Obtain predictions (BLUPs) from a* `splinemixmeta` *object*

Description

Obtain predictions (BLUPs) from a `splinemixmeta` object

Usage

```
## S3 method for class 'splinemixmeta'
blup(
  object,
  se = FALSE,
  pi = FALSE,
  vcov = FALSE,
  pi.level = 0.95,
  type = "outcome",
  level,
  format,
  aggregate = "stat",
  ...
)
```

Arguments

`object`	An object of class `splinemixmeta`, returned by `splinemixmeta()`.
`se`	Logical indicating whether to return standard errors of the predictions.
`pi`	Logical indicating whether to return prediction intervals.
`vcov`	Logical indicating whether to return the variance-covariance matrix of the predictions.
`pi.level`	Numeric value between 0 and 1 indicating the confidence level for the prediction intervals. Default is 0.95.
`type`	Character string specifying the type of prediction: "outcome" for predicted outcomes or "residual" for predicted residuals.
`level`	Integer indicating the random effects level for which to obtain predictions. Default is the highest level.
`format`	Character string specifying the format of the output: "matrix" or "list". Default depends on the number of outcomes and whether `vcov` is requested.
`aggregate`	Character string specifying how to aggregate the output when multiple outcomes are present
`...`	Additional arguments (currently unused).

Details

This function is modified from `mixmeta::blup.mixmeta()` with acknowledgement of the original authors. It is modified to handle intermediate levels of random effects more carefully.

Value

A matrix or list of predicted values (BLUPs), optionally including standard errors, prediction intervals, and variance-covariance matrices.

`make_smm_smooth` *Create random effects and fixed effects specifications from an mgcv smooth term for use in* `splinemixmeta`.

Description

This function is for internal use by `splinemixmeta()`.

Usage

```
make_smm_smooth(
  smooth,
  data,
  vnames = character(),
  manual_fixed = FALSE,
  envir = parent.frame()
)
```

Arguments

`smooth`	A term created by `mgcv::s()`
`data`	Data frame containing the variables used in the smooth. See `envir`.
`vnames`	"a vector of names to avoid as dummy variable names in the random effects form", per `mgcv::smooth2random()`, to which `vnames` is passed.
`manual_fixed`	If `FALSE`, the unpenalized dimensions (typically a linear term) of the smooth are used as fixed effects. If `TRUE`, the user should provide any desired fixed effects directly. In either case, unpenalized dimensions of the smooth term are not included in the spline. See details.
`envir`	The environment in which to evaluate variable names if `data` is not provided.

Details

This function uses `mgcv::smoothCon()` and `mgcv::smooth2random()` to obtain basis functions, penalty matrix, and (optionally) fixed effects terms from the `smooth` specification.

The fixed effects (if `manual_fixed` is `FALSE`) represent unpenalized directions of the smooth. Typically, this means the fixed effects will include linear terms, because splines usually penalize curvature, so any parameters that give a line are unpenalized. If one is not particularly interested in

the linear terms (more generally, unpenalized terms), then the default of `manual_fixed = FALSE` is a good option. However, if one is interested in coefficients for the linear terms, it is important to note that when they are extracted from the basis function setup, they may be (typically will be) also re-scaled, and it is not particularly easy to determine the scaling factor. Hence, one may prefer to set `manual_fixed=FALSE` and provide the linear term directly in the `formula` argument to `splinemixmeta()`.

Value

A list with elements `basisFxns` (the basis functions for the random effects) and `x_fixed` (the fixed effects design matrix, or `NULL` if `manual_fixed` is `TRUE`).

`plot.splinemixmeta` *Plot results from a univariate* `splinemixmeta` *model*

Description

Plot results from a univariate `splinemixmeta` model

Usage

```
## S3 method for class 'splinemixmeta'
plot(
  x,
  xvar,
  title,
  xlab,
  ylab,
  ylim,
  linecolor = "blue",
  fillcolor = "blue",
  ...
)
```

Arguments

`x`	object of class `mixmeta` returned from `splinemixmeta`
`xvar`	name of the variable to plot on the horizontal axis. If missing, this will will be taken from the right side of the fixed effects formula.
`title`	to add to the plot
`xlab`	label for the horizontal axis
`ylab`	label for the vertical axis
`ylim`	limits for the vertical axis
`linecolor`	color for the prediction line
`fillcolor`	color for the prediction confidence band
`...`	additional arguments passed to `predict.splinemixmeta()`

Details

This is not a very general plotting function. It is intended to provide a basic feature for visualizing a univariate splinemixmeta fit in a way that:

- includes fixed effects and spline terms in the predicted values, with 95% confidence bands
- Shows the data points with 95% confidence intervals obtained as +/- 2 times the standard errors (se or diag(S)).
- returns a ggplot2 object that can be further updated.

If the x object comes from a call to splinemixmeta() that had a simple (fixed effects) formula, such as y ~ w, then xvar does not need to be provided, since it is easily determined to be w. However, if the formula was more complicated, or if w was not provided in formula because manual_fixed = FALSE (the default), then xvar = "w" must be provided so that this plotting function knows what to put on the horizontal axis.

Value

ggplot2 object

---

predict.splinemixmeta *Make predictions from a fitted* splinemixmeta *model*

Description

Make predictions from a fitted splinemixmeta model

Usage

```
## S3 method for class 'splinemixmeta'
predict(
  object,
  include_smooths = TRUE,
  include_REs = FALSE,
  include_residuals = FALSE,
  type = "outcome",
  ...
)
```

Arguments

object	A fitted mixmeta object returned from splinemixmeta()
include_smooths	TRUE to include the smooth (spline) terms in predictions. Typically one wants these.
include_REs	TRUE to include any additional random effects (beyond the smooths) that were provided in the random argument to splinemixmeta. Omit these if you want to see just the spline predictions.

include_residuals
: TRUE to include the random effects (one for each datum) for residual variation not accounted for in the measurement variation (S). Only matters if residual_re = TRUE when calling splinemixmeta, which should typically be the case. Typically one does not want these in predictions.

type
: Type of predictions. This can be "outcome" or "residual" and will be passed to the type argument of blup.splinemixmeta().

...
: Additional arguments (currently unused)

Details

This is a convenience function that calls blup (i.e. blup.splinemixmeta) without requiring one to know which random-effects "levels" of the fitted mixmeta object correspond to which parts of the model. Instead one can simply choose whether to include smooths, random effects, and/or residuals. For more fine-grained control (such as including one spline term but not another), one can use blup() directly.

Value

A matrix with columns "blup" for the predicted values, "se" for the standard errors of the predictions, and "vcov" for the variance of the predictions. These are returned from mixmeta::blup() with vcov=TRUE and se=TRUE.

---

splinemixmeta *Fit a mixed-effects meta-analysis model with inclusion of one or more spline terms*

Description

Fit a mixed-effects meta-analysis model with inclusion of one or more spline terms

Usage

```
splinemixmeta(
  smooth = NULL,
  formula,
  se,
  S = se^2,
  manual_fixed = FALSE,
  residual_re = TRUE,
  data,
  random = list(),
  method = "reml",
  bscov = "unstr",
  ...
)
```

Arguments

smooth	A smoothing term created by mgcv::s(), or a list of such terms.
formula	A formula for the fixed effects part of the model.
se	A vector of standard errors for the response variables.
S	As an alternative to se, S can be provided in several formats to give variance-covariance information for the response variables.
manual_fixed	If TRUE, the fixed-effect component (if any) of any smooth terms is being manually included in the formula and hence should not be obtained as the unpenalized component(s) of smooth. Normally the "fixed-effect component" is the linear component. Hence one should either provide smooth = s(x) with x *omitted* from formula (e.g. formula = y or formula = y ~ 1) and thus manual_fixed = FALSE, or provide smooth = s(x) with x *included* in formula (e.g. formula = y ~ x) and manual_fixed = TRUE. The model fits should be identical but the coefficient for x will differ because x will be scaled differently if it was automatically obtained from the spline basis functions (i.e. with manual_fixed = FALSE). In either case, the unpenalized dimensions of the smooth term and not include in the spline random effects.
residual_re	If TRUE, a datum-level random effect for residual variation (beyond the measurement error specified by se or S) is automatically included (similar to the default behavior of mixmeta::mixmeta() when random is not specified). Normally this should be TRUE unless there is a clear reason to set it FALSE.
data	A data frame containing the variables in the model. If not provided, variables are sought from where the function was called.
random	See mixmeta::mixmeta(). This is a list of one-sided formulas specifying additional random effects beyond those that will be created from the smooth argument and residual_re.
method	This *must* be reml. It is provided as an argument to make clear that only reml is supported for estimating models where spline formulations are set up as random effects. This simplifies catching cases where a user might try to pass a different method value to mixmeta::mixmeta() via ....
bscov	See mixmeta::mixmeta(). This is relevant only if random is provided.
...	Additional arguments passed to mixmeta::mixmeta().'

Details

This function combines capabilities of mgcv and mixmeta in order to provide spline meta-regression, which means a meta-regression model where the shape of the relationship is unspecified and estimated from the data with smoothing splines. Spline components built from mgcv can be represented as random effects (along with fixed effects, which are unpenalized, typically for linear terms). mixmeta supports fairly flexible specification of fixed and random effects in (univariate or multivariate) meta-analysis regression (meta-regression) models. splinemixmeta takes mgcv-style specifications of smooth (spline) terms, sets them up for mixmeta, and the calls mixmeta to fit the model by REML.

Only a limited set of s arguments and options are supported. Argument k should work. For bs, supported basis functions include "cr", "cs", and "cc". Note that the default choice bs = "tp" does

not work well and so results in a warning. (The supported basis functions are those for which `mgcv::smoothCon` can produce diagonal penalty matrices are supported ("cr", "cs", "cc"), with the exception of "tp". Arguments `fx`, `m`, `by`, `id`, and `sp` are not supported should not be provided. Argument `xt` should work but is untested. Argument `pc` is untested.

Note that `bs="cc"` (cyclic cubic regression spline) does not have unpenalized components, so if this is used, `manual_fixed` is not relevant.

`splinemixmeta` is not particularly optimized for large data sets.

Value

An object of class `splinemixmeta`, unless there are no `smooth` terms, in which case an object of class `mixmeta` is returned.

See Also

- `predict.splinemixmeta()` for predictions based on BLUPs (best linear unbiased predictors) from fitted `splinemixmeta` models. This is an S3 method that will be called from `predict(x)` where `x` is a `splinemixmeta` object.
- `plot.splinemixmeta()` for plotting fitted spline meta-regression models. This is an S3 method that will be called from `plot(x)` where `x` is a `splinemixmeta` object.
- `blup.splinemixmeta()` for obtaining BLUPs (best linear unbiased predictors) from fitted `splinemixmeta` models. This is used by `predict.splinemixmeta`, which is typically easier to call directly. This is an S3 method that will be called from `blup(x)` where `x` is a `splinemixmeta` object.
- `make_smm_smooth()` for the internal function that sets up spline terms for use in `splinemixmeta`.

Index