

Package ‘buildmer’

February 25, 2017

Title Stepwise Elimination and Term Reordering for Mixed-Effects Regression

Version 0.1

Description The buildmer package attempts to build, from the user's specifications, the largest possible regression model that will still converge, and then eliminates all terms from it that do not significantly contribute to an improvement of model deviance. Optional reordering of the terms by their contribution to the deviance (like SPSS) is also supported.

Depends R (>= 3.2), mgcv, lme4

Suggests lmerTest, pbkrtest, gamm4

License 2-clause BSD

Encoding UTF-8

LazyData true

RoxygenNote 5.0.1

R topics documented:

add.terms	2
buildmer	2
buildmer-class	4
calcor	4
calcWald	5
conv	5
custround	5
devfun	6
diag,formula-method	6
get.last.random.slope	7
get.random.terms	7
has.smooth.terms	8
hasREML	8
innerapply	9
is.random.term	9
mcmc2tex	10
mer2tex	10
nohp	10
paperify	11
remove.terms	11
stars	11

stepwise	12
tblprintln	12
vowels	13

Index	14
--------------	-----------

add.terms	<i>Add terms to an lme4 formula.</i>
-----------	--------------------------------------

Description

Add terms to an lme4 formula.

Usage

```
add.terms(formula, add)
```

Arguments

formula	The lme4 formula to add terms to.
add	A vector of terms to add. To add terms nested in random-effect groups, use ‘(term group)’ syntax if you want to add an independent random effect (e.g. ‘(olderterm group) + (term group)’), or use ‘term group’ syntax if you want to add a dependent random effect to a pre-existing term group (if no such group exists, it will be created at the end of the formula).

Value

The updated formula.

See Also

buildmer

buildmer	<i>Construct and fit as complete a model as possible, optionally reorder terms by their contribution to the deviance, and perform stepwise elimination using the change in deviance</i>
----------	---

Description

Construct and fit as complete a model as possible, optionally reorder terms by their contribution to the deviance, and perform stepwise elimination using the change in deviance

Usage

```
buildmer(formula, data, family = gaussian, adjust.p.chisq = TRUE,
  reorder.terms = TRUE, parallel = TRUE, reduce.fixed = TRUE,
  reduce.random = TRUE, protect.intercept = TRUE, direction = "backward",
  calc.anova = TRUE, calc.summary = TRUE, ddf = "Wald", quiet = FALSE,
  ...)
```

Arguments

<code>formula</code>	The model formula for the maximal model you would like to fit, if possible. Supports lme4 random effects and gamm4 smooth terms.
<code>data</code>	The data to fit the models to.
<code>family</code>	The error distribution to use. Only relevant for generalized models; if the family is empty or 'gaussian', the models will be fit using <code>lm(er)</code> , otherwise they will be fit using <code>glm(er)</code> with the specified error distribution passed through.
<code>adjust.p.chisq</code>	Whether to adjust for overconservativity of the likelihood ratio test by dividing p-values by 2 (see Pinheiro & Bates 2000).
<code>reorder.terms</code>	Whether to reorder the terms by their contribution to the deviance before testing them.
<code>parallel</code>	Whether to, if possible, parallelize the fitting of the candidate models during term reordering.
<code>reduce.fixed</code>	Whether to reduce the fixed-effect structure.
<code>reduce.random</code>	Whether to reduce the random-effect structure.
<code>protect.intercept</code>	If TRUE, the fixed-effect intercept will not be removed from the model, even if it is deemed nonsignificant. This is strongly recommended.
<code>direction</code>	The direction for stepwise elimination; either 'forward' or 'backward' (default). Both or neither are also understood.
<code>calc.anova</code>	Whether to also calculate the ANOVA table for the final model after term elimination. This is useful if you want to calculate degrees of freedom by Kenward-Roger approximation (default), in which case generating the ANOVA table (via <code>lmerTest</code>) will be very slow, and preparing the ANOVA in advance can be advantageous.
<code>calc.summary</code>	Whether to also calculate the summary table for the final model after term elimination. This is useful if you want to calculate degrees of freedom by Kenward-Roger approximation (default), in which case generating the summary (via <code>lmerTest</code>) will be very slow, and preparing the summary in advance can be advantageous.
<code>ddf</code>	The method used for calculating p-values if <code>summary=T</code> . Options are 'Wald' (default), 'Satterthwaite' (if <code>lmerTest</code> is available), 'Kenward-Roger' (if <code>lmerTest</code> and <code>pbkrtest</code> are available), and 'lme4' (no p-values).
<code>quiet</code>	Whether to suppress progress messages.
<code>...</code>	Additional options to be passed to (g)lmer or gamm4. (They will also be passed to (g)lm in so far as they're applicable, so you can use arguments like 'subset=...' and expect things to work. The single exception is the 'control' argument, which is assumed to be meant only for (g)lmer and not for glm, and will NOT be passed on to glm.)

Value

A `buildmer` object containing the following slots:

- `table`: a dataframe containing the results of the elimination process
- `model`: the final model containing only the terms that survived elimination
- `messages`: any warning messages
- `summary`: the model's summary, if '`calc.summary=TRUE`' was passed
- `anova`: the model's anova, if '`calc.anova=TRUE`' was passed

Examples

```
buildmer(Reaction~Days+(Days|Subject),sleepstudy)
```

buildmer-class	<i>Make a buildmer object</i>
----------------	-------------------------------

Description

Make a buildmer object

Arguments

table	A dataframe containing the results of the elimination process.
model	The final model containing only the terms that survived elimination.
messages	Any warning messages.
summary:	The model's summary, if the model was built with 'summary=TRUE'.
anova:	The model's ANOVA, if the model was built with 'anova=TRUE'.

See Also

buildmer

calcor	<i>Calculate Odds Ratio; basically prints either 'exp(x):1' or '1:exp(x)'</i>
--------	---

Description

Calculate Odds Ratio; basically prints either 'exp(x):1' or '1:exp(x)'

Usage

```
calcor(x)
```

calcWald	<i>Calculate p-values based on Wald z-scores</i>
----------	--

Description

Calculate p-values based on Wald z-scores

Usage

```
calcWald(table, i, sqrt = F)
```

Arguments

table	A coefficient table from a summary or anova output.
i	The number of the column in that table containing the t-values.
sqrt	Whether we're testing F values or t values (default).

Value

The table augmented with p-values.

conv	<i>Test an merMod or equivalent object for convergence</i>
------	--

Description

Test an merMod or equivalent object for convergence

Usage

```
conv(model)
```

Arguments

model	The model object to test.
-------	---------------------------

Value

Whether the model converged or not.

custround	<i>Custom rounding function</i>
-----------	---------------------------------

Description

Custom rounding function

Usage

```
custround(i, neg = T, trunc = F)
```

devfun	<i>Extract a model's deviance</i>
--------	-----------------------------------

Description

Extract a model's deviance

Usage

```
devfun(model)
```

Arguments

model	The fitted model object.
-------	--------------------------

Value

The deviance or REML criterion.

diag,formula-method	<i>Diagonalize the random-effect covariance structure, possibly assisting convergence</i>
---------------------	---

Description

Diagonalize the random-effect covariance structure, possibly assisting convergence

Usage

```
## S4 method for signature 'formula'
diag(x)
```

Arguments

formula	A model formula.
---------	------------------

Value

The formula with all random-effect correlations forced to zero, per Pinheiro & Bates (2000).

get.last.random.slope	<i>Get the last random slope (or, if not available, the last random intercept) from a model formula</i>
-----------------------	---

Description

Get the last random slope (or, if not available, the last random intercept) from a model formula

Usage

```
get.last.random.slope(formula)
```

Arguments

formula	A model formula.
---------	------------------

Value

The last random slope (or, if not available, the last random intercept).

get.random.terms	<i>Get the level-2 terms contained within a level-1 lme4 random term</i>
------------------	--

Description

Get the level-2 terms contained within a level-1 lme4 random term

Usage

```
get.random.terms(term)
```

Arguments

term	The term.
------	-----------

Value

A list of random terms.

has.smooth.terms	<i>Test whether a formula contains gamm4 smooth terms</i>
------------------	---

Description

Test whether a formula contains gamm4 smooth terms

Usage

```
has.smooth.terms(formula)
```

Arguments

formula	The formula.
---------	--------------

Value

A logical indicating whether the formula has any gamm4 terms.

hasREML	<i>Test whether a model was fit with REML</i>
---------	---

Description

Test whether a model was fit with REML

Usage

```
hasREML(model)
```

Arguments

model	A fitted model object.
-------	------------------------

Value

TRUE or FALSE if the model was a linear mixed-effects model that was fit with REML or not, respectively; NA otherwise.

innerapply	<i>Convenience function to immediately descend into a level-2 term</i>
------------	--

Description

Convenience function to immediately descend into a level-2 term

Usage

```
innerapply(random.terms, FUN)
```

Arguments

random.terms	Vector of level-1 terms to descend into.
FUN	The function to apply to the level-2 terms.

Value

The modified random.terms

is.random.term	<i>Test whether a formula term contains lme4 random terms</i>
----------------	---

Description

Test whether a formula term contains lme4 random terms

Usage

```
is.random.term(term)
```

Arguments

term	The term.
------	-----------

Value

A logical indicating whether the term was a random-effects term.

mcmc2tex	<i>Convert an MCMCglmm model to LaTeX code (biased towards stress analysis)</i>
----------	---

Description

Convert an MCMCglmm model to LaTeX code (biased towards stress analysis)

Usage

```
mcmc2tex(model, label = "", aliases = list())
```

Arguments

model	The fitted model (not its summary!)
label	The LaTeX label to put below your 'Results' caption.

mer2tex	<i>Convert a buildmer (or compatible) summary to LaTeX code (biased towards vowel analysis)</i>
---------	---

Description

Convert a buildmer (or compatible) summary to LaTeX code (biased towards vowel analysis)

Usage

```
mer2tex(summary, vowel = "", formula = F, label = "", aliases = list())
```

Arguments

summary	The summary to convert.
vowel	The vowel you're analyzing.
formula	The formula as used in your final lmer object.
label	The LaTeX label to put below your 'Results' caption.

nohp	<i>Remove \phantom-</i>
------	-------------------------

Description

Remove \phantom-

Usage

```
nohp(x)
```

paperify	<i>Translate formula terms to aliases</i>
----------	---

Description

Translate formula terms to aliases

Usage

```
paperify(x, aliases)
```

remove.terms	<i>Remove terms from an lme4 formula</i>
--------------	--

Description

Remove terms from an lme4 formula

Usage

```
remove.terms(formula, remove = c(), formulize = T)
```

Arguments

formula	The lme4 formula.
remove	A vector of terms to remove. To remove terms nested inside random-effect groups, use ‘term group’ syntax. Note that marginality is respected, i.e. no effects will be removed if they participate in a higher-order interaction, and no fixed effects will be removed if a random slope is included over that fixed effect.
formulize	Whether to return a formula (default) or a simple list of terms.

See Also

buildmer

stars	<i>LaTeXify significance stars</i>
-------	------------------------------------

Description

LaTeXify significance stars

Usage

```
stars(x)
```

stepwise	<i>A simple interface to buildmer intended to mimic SPSS stepwise methods for term reordering and backward stepwise elimination</i>
----------	---

Description

A simple interface to buildmer intended to mimic SPSS stepwise methods for term reordering and backward stepwise elimination

Usage

```
stepwise(formula, data, family = gaussian, ...)
```

Arguments

formula	The model formula for the maximal model you would like to fit, if possible. Supports lme4 random effects and gamm4 smooth terms.
data	The data to fit the models to.
family	The error distribution to use. Only relevant for generalized models; if the family is empty or 'gaussian', the models will be fit using lm(er), otherwise they will be fit using glm(er) with the specified error distribution passed through. Commonly-used options are either nothing/'gaussian' (linear regression), 'binomial' (logistic regression), or 'poisson' (loglin regression), although many other families exist (e.g. cloglog, ...).
...	Additional parameters that override buildmer defaults, see 'buildmer'.

Value

A buildmer object, which you can use summary() on to get a summary of the final model, and elim() to get the list of eliminated terms.

Examples

```
buildmer(Reaction~Days+(Days|Subject),sleepstudy)
```

tblprintln	<i>Print an R vector as a LaTeX table line</i>
------------	--

Description

Print an R vector as a LaTeX table line

Usage

```
tblprintln(x)
```

vowels	<i>Vowel data from a pilot study.</i>
--------	---------------------------------------

Description

Vowel data from a pilot study.

Usage

```
data(vowels)
```

Format

A standard data frame.

Examples

```
#buildmer(f1 ~ vowel*timepoint*following + stress + information + (vowel*timepoint*following|participant),data=vowels)
buildmer(f1 ~ vowel + timepoint + stress + following + information + vowel:timepoint + timepoint:following + vowel:timepoint:following, data=vowels)
```

Index

- *Topic **LaTeX**
 - mcmc2tex, 10
 - mer2tex, 10
- *Topic **SPSS**,
 - stepwise, 12
- *Topic **add**
 - add.terms, 2
- *Topic **buildmer**,
 - buildmer, 2
 - stepwise, 12
- *Topic **buildmer**
 - buildmer-class, 4
- *Topic **convergence**
 - conv, 5
- *Topic **covariance**
 - diag, formula-method, 6
- *Topic **datasets**
 - vowels, 13
- *Topic **diagonal**
 - diag, formula-method, 6
- *Topic **elimination**,
 - buildmer, 2
 - stepwise, 12
- *Topic **fit**,
 - buildmer, 2
- *Topic **order**
 - buildmer, 2
 - stepwise, 12
- *Topic **remove**
 - remove.terms, 11
- *Topic **stepwise**
 - buildmer, 2
 - stepwise, 12
- *Topic **structure**
 - diag, formula-method, 6
- *Topic **terms**
 - add.terms, 2
 - remove.terms, 11
- *Topic **term**
 - buildmer, 2
 - stepwise, 12
- A (mcmc2tex), 10
- A (mer2tex), 10
- add.terms, 2
- aliases (mcmc2tex), 10
- aliases (mer2tex), 10
- buildmer, 2
- buildmer-class, 4
- calcor, 4
- calcWald, 5
- code. (mcmc2tex), 10
- code. (mer2tex), 10
- conv, 5
- custround, 5
- devfun, 6
- diag, formula-method, 6
- get.last.random.slope, 7
- get.random.terms, 7
- has.smooth.terms, 8
- hasREML, 8
- innerapply, 9
- is.random.term, 9
- LaTeX (mcmc2tex), 10
- LaTeX (mer2tex), 10
- list (mcmc2tex), 10
- list (mer2tex), 10
- mcmc2tex, 10
- mer2tex, 10
- mkBuildmer (buildmer-class), 4
- nohp, 10
- of (mcmc2tex), 10
- of (mer2tex), 10
- paperify, 11
- remove.terms, 11
- stars, 11
- stepwise, 12

`summary (mcmc2tex)`, [10](#)
`summary (mer2tex)`, [10](#)

`tblprintln`, [12](#)

`terms (mcmc2tex)`, [10](#)

`terms (mer2tex)`, [10](#)

`to (mcmc2tex)`, [10](#)

`to (mer2tex)`, [10](#)

`translating (mcmc2tex)`, [10](#)

`translating (mer2tex)`, [10](#)

`vowels`, [13](#)