Package 'reformulas'

November 3, 2024

Title Machinery for Processing Random Effect Formulas

Version 0.4.0

Description Takes formulas including random-effects components (formatted as in 'lme4', 'glmmTMB', etc.) and processes them. Includes various helper functions.

URL https://github.com/bbolker/reformulas

License GPL-3

Encoding UTF-8

Imports stats, methods, Matrix, Rdpack

RdMacros Rdpack

Suggests lme4, tinytest

RoxygenNote 7.3.2

NeedsCompilation no

Author Ben Bolker [aut, cre] (<https://orcid.org/0000-0002-2127-0443>)

Maintainer Ben Bolker <bolker@mcmaster.ca>

Repository CRAN

Date/Publication 2024-11-03 05:30:11 UTC

Contents

anySpecial *Detect whether there are any 'specials' in a formula term*

Description

Detect whether there are any 'specials' in a formula term

Usage

```
anySpecial(term, specials = findReTrmClasses(), fast = FALSE)
```

Arguments

term	formula term
specials	values to detect
fast	(logical) use quick (syntactic) test for presence of specials?

Value

logical value

Examples

```
## should only detect s as the head of a function, s(...)
anySpecial(~diag(1))
anySpecial(~diag)
anySpecial(~diag[[1]])
anySpecial(~diag[1])
anySpecial(~s)
anySpecial(~s(hello+goodbye,whatever))
```

expandDoubleVerts *Expand terms with '||' notation into separate '|' terms*

Description

From the right hand side of a formula for a mixed-effects model, expand terms with the double vertical bar operator into separate, independent random effect terms.

Usage

```
expandDoubleVerts(term)
```

Arguments

term	a mixed-model formula

Value

the modified term

See Also

formula, model.frame, model.matrix.

Other utilities: mkReTrms(), nobars(), subbars()

`expandGrpVar` *apply*

Description

apply

Usage

```
expandGrpVar(f)
```

Arguments

f	a language object (an atom of a formula) expandGrpVar(quote(x*y)) expandGrpVar(quote(x/y))

`findReTrmClasses` *list of specials – taken from enum.R*

Description

list of specials – taken from enum.R

Usage

```
findReTrmClasses()
```

isNested *Is f1 nested within f2?*

Description

Does every level of f1 occur in conjunction with exactly one level of f2? The function is based on converting a triplet sparse matrix to a compressed column-oriented form in which the nesting can be quickly evaluated.

Usage

```
isNested(f1, f2)
```

Arguments

f1	factor 1
f2	factor 2

Value

TRUE if factor 1 is nested within factor 2

Examples

```
if (requireNamespace("lme4")) {
   data("Pastes", package = "lme4")
   with(Pastes, isNested(cask, batch))   ## => FALSE
   with(Pastes, isNested(sample, batch))  ## => TRUE
}
```

mkReTrms *Create list of structures needed for models with random effects*

Description

From the result of findbars applied to a model formula and and the evaluation frame, create the model matrix, etc. associated with random-effects terms. See the description of the returned value for a detailed list.

Usage

```
mkReTrms(
  bars,
  fr,
  drop.unused.levels = TRUE,
  reorder.terms = TRUE,
  reorder.vars = FALSE,
  calc.lambdat = TRUE
)
```

Arguments

bars	a list of parsed random-effects terms
fr	a model frame in which to evaluate these terms
drop.unused.levels	(logical) drop unused factor levels?
reorder.terms	arrange random effects terms in decreasing order of number of groups (factor levels)?
reorder.vars	arrange columns of individual random effects terms in alphabetical order?
calc.lambdat	(logical) compute Lambdat and Lind components? (At present these components are needed for lme4 machinery but not for glmmTMB, and may be large in some cases; see Bates *et al.* 2015

Value

a list with components

Zt	transpose of the sparse model matrix for the random effects
Ztlist	list of components of the transpose of the random-effects model matrix, separated by random-effects term
Lambdat	transpose of the sparse relative covariance factor
Lind	an integer vector of indices determining the mapping of the elements of the theta to the "x" slot of Lambdat
theta	initial values of the covariance parameters
lower	lower bounds on the covariance parameters
flist	list of grouping factors used in the random-effects terms
cnms	a list of column names of the random effects according to the grouping factors
Gp	a vector indexing the association of elements of the conditional mode vector with random-effect terms; if nb is the vector of numbers of conditional modes per term (i.e. number of groups times number of effects per group), Gp is c(0,cumsum(nb)) (and conversely nb is diff(Gp))
nl	names of the terms (in the same order as Zt, i.e. reflecting the reorder.terms argument)

References

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software*, **67**(1), 1–48. doi:10.18637/jss.v067.i01.)

See Also

Other utilities: `expandDoubleVerts()`, `nobars()`, `subbars()`

`nobars` *Omit terms separated by vertical bars in a formula*

Description

Remove the random-effects terms from a mixed-effects formula, thereby producing the fixed-effects formula.

Usage

```
nobars(term)

nobars_(term)
```

Arguments

`term`	the right-hand side of a mixed-model formula

Value

the fixed-effects part of the formula

Note

This function is called recursively on individual terms in the model, which is why the argument is called `term` and not a name like `form`, indicating a formula.

See Also

`formula`, `model.frame`, `model.matrix`.

Other utilities: `expandDoubleVerts()`, `mkReTrms()`, `subbars()`

Examples

```
nobars(Reaction ~ Days + (Days|Subject)) ## => Reaction ~ Days
```

no_specials *Drop 'specials' from a formula*

Description

Drop 'specials' from a formula

Usage

```
no_specials(term, specials = c("|", "||", "s"))
```

Arguments

term	a term or formula or list thereof
specials	function types to drop

Value

a call or language object (or list) with specials removed

Examples

```
no_specials(findbars_x(~ 1 + s(x) + (f|g) + diag(x|y)))
no_specials(~us(f|g))
```

RHSForm *extract right-hand side of a formula*

Description

extract right-hand side of a formula

Usage

```
RHSForm(form, as.form = FALSE)
```

Arguments

form	a formula object
as.form	(logical) return a formula (TRUE) or as a call/symbolic object (FALSE) ?

Value

a language object

Examples

```
RHSForm(y ~ x + (1|g))
```

subbars *"Substitute bars"*

Description

Substitute the '+' function for the '|' and '||' function in a mixed-model formula. This provides a formula suitable for the current model.frame function.

Usage

```
subbars(term)
```

Arguments

term	a mixed-model formula

Value

the formula with all | and || operators replaced by +

Note

This function is called recursively on individual terms in the model, which is why the argument is called term and not a name like form, indicating a formula.

See Also

formula, model.frame, model.matrix.

Other utilities: expandDoubleVerts(), mkReTrms(), nobars()

Examples

```
subbars(Reaction ~ Days + (Days|Subject)) ## => Reaction ~ Days + (Days + Subject)
```

Index