Package ‘subformula’

October 14, 2022

Type Package
Title Create Subformulas of a Formula
Version 0.1.0
Description A formula 'sub' is a subformula of 'formula' if all the terms on the right hand side of 'sub' are terms of 'formula' and their left hand sides are identical. This package aids in the creation of subformulas.
License MIT + file LICENSE
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
Suggests testthat, rmarkdown, knitr, covr
VignetteBuilder knitr
URL https://github.com/JonasMoss/subformula,

BugReports https://github.com/JonasMoss/subformula/issues

NeedsCompilation no
Author Jonas Moss [aut, cre] (<https://orcid.org/0000-0002-6876-6964>)
Maintainer Jonas Moss <jonas.gjertsen@gmail.com>
Repository CRAN
Date/Publication 2019-11-15 17:10:02 UTC

R topics documented:

fapply ................................................................. 2
subformula ......................................................... 2

Index 4
**fapply**  
*Apply Formulas to a Model*

**Description**

fapply returns a list of the same length as formulas. Each element is the result of applying model the corresponding element of formulas.

**Usage**

fapply(formulas, model, ...)

**Arguments**

- **formulas**  
a list of formulas or objects coercible to formula by stats::as.formula.
- **model**  
a function taking a formula as its first argument.
- **...**  
additional arguments to be passed to model.

**Details**

This is a member of the apply family. It is similar to lapply, but handles the call slightly differently. This makes the output prettier.

**Value**

fapply returns a list of evaluated function calls.

**Examples**

```r
formulas = subformula(mpg ~ cyl + disp, protected = ~ cyl)
fapply(formulas, lm, data = mtcars) # Pretty output.
lapply(formulas, lm, data = mtcars) # Less pretty output.
```

---

**subformula**  
*Calculate Subformulas*

**Description**

A formula sub is a subformula of formula if (i) all the terms on the right hand side of sub are terms of formula and (ii) their left hand sides are identical. subformula finds every subformula of formula that contains each term in protected.

**Usage**

subformula(formula, protected = NULL, data = NULL)
subformula

Arguments

- **formula**: an object of class "formula" (or one that can be coerced to that class via `formula`).
- **protected**: a vector or formula specifying which covariates are *protected*. Protected formulas appear in all subformulas.
- **data**: an optional data frame (or object coercible by `as.data.frame` to a data frame). Used to fill out formulas as `y ~ ..`

Details

Protected terms will appear in every subformula. If the supplied formula includes the term 0 or -1, none of the subformulas will include the intercept. Otherwise, the intercept will be interpreted as being protected. If `formula` is is coerced to a `formula` object, its associated `environment` will be `NULL`. All subformulas will inherit their `.Environment` attribute from `formula`.

Value

`subformula` returns a list of `formula` objects.

Examples

```r
subformula(z ~ x + y)
subformula(y ~ x + y + y^2, protected = ~ x)
subformula(y ~ x + y + t + I(t^2), protected = c("x","I(t^2)"))
```
Index

apply, 2
as.data.frame, 3

call, 2

environment, 3

fapply, 2
formula, 2, 3

lapply, 2

stats::as.formula, 2
subformula, 2