Package 'insight'

March 5, 2019

Type Package

Title Easy Access to Model Information for Various Model Objects

Description A tool to provide an easy, intuitive and consistent access to information contained in various R models, like model formulas, model terms, information about random effects, data that was used to fit the model or data from response variables. 'insight' mainly revolves around two types of functions: Functions that find (the names of) information, starting with 'find_', and functions that get the underlying data, starting with 'get_'. The package has a consistent syntax and works with many different model objects, where otherwise functions to access these information are missing.

Version 0.1.2

Date 2019-02-27

Maintainer Daniel Lüdecke <d.luedecke@uke.de>

License GPL-3

URL https://easystats.github.io/insight/

BugReports https://github.com/easystats/insight/issues

Depends R (>= 3.0), stats

Suggests AER, betareg, brms, coxme, gam, gamm4, gee, GLMMadaptive, glmmTMB, gmnl, lfe, MASS, MCMCglmm, mlogit, lme4, mgcv, nnet, nlme, ordinal, plm, pscl, rstanarm, splines, survey, survival, truncreg, testthat, VGAM

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Daniel Lüdecke [aut, cre] (<https://orcid.org/0000-0002-8895-3206>), Dominique Makowski [aut, ctb] (<https://orcid.org/0000-0001-5375-9967>)

Repository CRAN

Date/Publication 2019-03-05 15:20:07 UTC

R topics documented:

`clean_names` *Get clean names of model terms*

Description

This function "cleans" names of model terms (or a character vector with such names) by removing patterns like `log()` or `as.factor()` etc.

Usage

```
clean_names(x)
```

Arguments

`x`	A fitted model, or a character vector.

Value

The "cleaned" variable names as character vector, i.e. pattern like `s()` for splines or `log()` are removed from the model terms.

Note

If `x` is a regression model, this function is equal to calling `find_terms()`.

Examples

```
# example from ?stats::glm
counts <- c(18, 17, 15, 20, 10, 20, 25, 13, 12)
outcome <- c(gl(3, 1, 9))
treatment <- gl(3, 3)
m <- glm(counts ~ log(outcome) + as.factor(treatment), family = poisson())
clean_names(m)
```

---

find_formula *Find model formula*

Description

Returns the formula(s) for the different parts of a model (like fixed or random effects, zero-inflated component, ...).

Usage

```
find_formula(x, ...)
```

Arguments

x	A fitted model.
...	Currently not used.

Value

A list of formulas that describe the model. For simple models, only one list-element, `conditional`, is returned. For more complex models, the returned list may have following elements:

- `conditional`, the "fixed effects" part from the model
- `random`, the "random effects" part from the model (or the `id` for gee-models and similar)
- `zero_inflated`, the "fixed effects" part from the zero-inflation component of the model
- `zero_inflated_random`, the "random effects" part from the zero-inflation component of the model
- `dispersion`, the dispersion formula
- `instruments`, for fixed-effects regressions like `ivreg`, `felm` or `plm`, the instrumental variables

Examples

```
data(mtcars)
m <- lm(mpg ~ wt + cyl + vs, data = mtcars)
find_formula(m)
```

`find_parameters` *Find names of model parameters*

Description

Returns the names of model parameters, like they typically appear in the `summary()` output. For Bayesian models, the parameter names equal the column names of the posterior samples after coercion from `as.data.frame()`.

Usage

```
find_parameters(x, ...)
```

Arguments

`x`	A fitted model.
`...`	Currently not used.

Value

A list of parameter names. For simple models, only one list-element, `conditional`, is returned. For more complex models, the returned list may have following elements:

- `conditional`, the "fixed effects" part from the model
- `random`, the "random effects" part from the model
- `zero_inflated`, the "fixed effects" part from the zero-inflation component of the model
- `zero_inflated_random`, the "random effects" part from the zero-inflation component of the model
- `dispersion`, the dispersion formula
- `within`, the within-subject effects of Anovas (`aov()`) with error term
- `between`, the between-subjects effects of Anovas (`aov()`) with error term

Examples

```
data(mtcars)
m <- lm(mpg ~ wt + cyl + vs, data = mtcars)
find_parameters(m)
```

find_predictors *Find names of model predictors*

Description

Returns the names of the predictor variables for the different parts of a model (like fixed or random effects, zero-inflated component, ...). Unlike find_parameters, the names from find_predictors() match the original variable names from the data that was used to fit the model.

Usage

```
find_predictors(x, effects = c("fixed", "random", "all"),
  component = c("all", "conditional", "zi", "zero_inflated",
  "dispersion", "instruments"), flatten = FALSE)
```

Arguments

x	A fitted model.
effects	Should variables for fixed effects, random effects or both be returned? Only applies to mixed models. May be abbreviated.
component	Should all predictor variables, predictor variables for the conditional model, the zero-inflated part of the model, the dispersion term or the instrumental variables be returned? Applies to models with zero-inflated and/or dispersion formula, or to models with instrumental variable (so called fixed-effects regressioms). May be abbreviated.
flatten	Logical, if TRUE, the values are returned as character vector, not as list.

Value

A list of character vectors that represent the name(s) of the predictor variables. Depending on the combination of the arguments effects and component, the returned list has following elements:

- conditional, the "fixed effects" terms from the model
- random, the "random effects" terms from the model
- zero_inflated, the "fixed effects" terms from the zero-inflation component of the model
- zero_inflated_random, the "random effects" terms from the zero-inflation component of the model
- dispersion, the dispersion terms
- instruments, for fixed-effects regressions like ivreg, felm or plm, the instrumental variables

Examples

```
data(mtcars)
m <- lm(mpg ~ wt + cyl + vs, data = mtcars)
find_predictors(m)
```

`find_random` *Find names of random effects*

Description

Return the name of the grouping factors from mixed effects models.

Usage

```
find_random(x, split_nested = FALSE, flatten = FALSE)
```

Arguments

`x`	A fitted mixed model.
`split_nested`	Logical, if `TRUE`, terms from nested random effects will be returned as separated elements, not as single string with colon. See 'Examples'.
`flatten`	Logical, if `TRUE`, the values are returned as character vector, not as list.

Value

A list of character vectors that represent the name(s) of the random effects (grouping factors). Depending on the model, the returned list has following elements:

- `random`, the "random effects" terms from the conditional part of model
- `zero_inflated_random`, the "random effects" terms from the zero-inflation component of the model

Examples

```
library(lme4)
data(sleepstudy)
sleepstudy$mygrp <- sample(1:5, size = 180, replace = TRUE)
sleepstudy$mysubgrp <- NA
for (i in 1:5) {
  filter_group <- sleepstudy$mygrp == i
  sleepstudy$mysubgrp[filter_group] <-
    sample(1:30, size = sum(filter_group), replace = TRUE)
}

m <- lmer(
  Reaction ~ Days + (1 | mygrp / mysubgrp) + (1 | Subject),
  data = sleepstudy
)

find_random(m)
find_random(m, split_nested = TRUE)
```

find_response *Find name of the response variable*

Description

Returns the name(s) of the response variable(s) from a model object.

Usage

```
find_response(x, combine = TRUE)
```

Arguments

x	A fitted model.
combine	Logical, if TRUE and the response is a matrix-column, the name of the response matches the notation in formula, and would for instance also contain patterns like "cbind(...)". Else, the original variable names from the matrix-column are returned. See 'Examples'.

Value

The name(s) of the response variable(s) from x as character vector.

Examples

```
library(lme4)
data(cbpp)
cbpp$trials <- cbpp$size - cbpp$incidence
m <- glm(cbind(incidence, trials) ~ period, data = cbpp, family = binomial)

find_response(m, combine = TRUE)
find_response(m, combine = FALSE)
```

find_terms *Find names of all model terms*

Description

Returns a list with the names of all model terms, including response value and random effects.

Usage

```
find_terms(x, effects = c("all", "fixed", "random"),
  component = c("all", "conditional", "zi", "zero_inflated",
  "dispersion", "instruments"), flatten = FALSE)
```

Arguments

x	A fitted model.
effects	Should variables for fixed effects, random effects or both be returned? Only applies to mixed models. May be abbreviated.
component	Should all predictor variables, predictor variables for the conditional model, the zero-inflated part of the model, the dispersion term or the instrumental variables be returned? Applies to models with zero-inflated and/or dispersion formula, or to models with instrumental variable (so called fixed-effects regressioms). May be abbreviated.
flatten	Logical, if TRUE, the values are returned as character vector, not as list.

Value

A list with (depending on the model) following elements (character vectors):

- response, the name of the response variable
- conditional, the names of the predictor variables from the *conditional* model (as opposed to the zero-inflated part of a model)
- random, the names of the random effects (grouping factors)
- zero_inflated, the names of the predictor variables from the *zero-inflated* part of the model
- zero_inflated_random, the names of the random effects (grouping factors)
- dispersion, the name of the dispersion terms
- instruments, the names of instrumental variables

Examples

```
library(lme4)
data(cbpp)
data(sleepstudy)
# some data preparation...
cbpp$trials <- cbpp$size - cbpp$incidence
sleepstudy$mygrp <- sample(1:5, size = 180, replace = TRUE)
sleepstudy$mysubgrp <- NA
for (i in 1:5) {
  filter_group <- sleepstudy$mygrp == i
  sleepstudy$mysubgrp[filter_group] <-
    sample(1:30, size = sum(filter_group), replace = TRUE)
}

m1 <- glmer(
  cbind(incidence, size - incidence) ~ period + (1 | herd),
  data = cbpp,
  family = binomial
)
find_terms(m1)

m2 <- lmer(
  Reaction ~ Days + (1 | mygrp / mysubgrp) + (1 | Subject),
```

```
  data = sleepstudy
)
find_terms(m2)
find_terms(m2, flatten = TRUE)
```

`find_variables` *Find names of all variables*

Description

Returns a list with the names of all variables, including response value and random effects, "as is". This means, on-the-fly tranformations like `log()`, `I()`, `as.factor()` etc. are preserved.

Usage

```
find_variables(x, ...)
```

Arguments

`x`	A fitted model.
`...`	Currently not used.

Value

A list with (depending on the model) following elements (character vectors):

- `response`, the name of the response variable
- `conditional`, the names of the predictor variables from the *conditional* model (as opposed to the zero-inflated part of a model)
- `random`, the names of the random effects (grouping factors)
- `zero_inflated`, the names of the predictor variables from the *zero-inflated* part of the model
- `zero_inflated_random`, the names of the random effects (grouping factors)
- `dispersion`, the name of the dispersion terms
- `instruments`, the names of instrumental variables

Examples

```
library(lme4)
data(sleepstudy)
m <- lmer(
  log(Reaction) ~ Days + I(Days^2) + (1 + Days + exp(Days) | Subject),
  data = sleepstudy
)

find_variables(m)
```

get_data *Get the data that was used to fit the model*

Description

This functions tries to get the data that was used to fit the model and returns it as data frame.

Usage

```
get_data(x, ...)

## S3 method for class 'hurdle'
get_data(x, component = c("all", "conditional", "zi",
  "zero_inflated", "dispersion"), ...)

## S3 method for class 'glmmTMB'
get_data(x, effects = c("all", "fixed", "random"),
  component = c("all", "conditional", "zi", "zero_inflated",
  "dispersion"), ...)

## S3 method for class 'merMod'
get_data(x, effects = c("all", "fixed", "random"), ...)

## S3 method for class 'clmm'
get_data(x, effects = c("all", "fixed", "random"), ...)

## S3 method for class 'lme'
get_data(x, effects = c("all", "fixed", "random"), ...)

## S3 method for class 'gee'
get_data(x, effects = c("all", "fixed", "random"), ...)

## S3 method for class 'MixMod'
get_data(x, effects = c("all", "fixed", "random"),
  component = c("all", "conditional", "zi", "zero_inflated",
  "dispersion"), ...)

## S3 method for class 'MCMCglmm'
get_data(x, effects = c("all", "fixed", "random"),
  ...)
```

Arguments

x	A fitted model.
...	Currently not used.

component	Should all predictor variables, predictor variables for the conditional model, the zero-inflated part of the model, the dispersion term or the instrumental variables be returned? Applies to models with zero-inflated and/or dispersion formula, or to models with instrumental variable (so called fixed-effects regressioms). May be abbreviated.
effects	Should model data for fixed effects, random effects or both be returned? Only applies to mixed models.

Value

The data that was used to fit the model.

Note

Unlike `model.frame()`, which may contain transformed variables (e.g. if `poly()` or `scale()` was used inside the formula to specify the model), `get_data()` aims at returning the "original", untransformed data.

Examples

```
data(cbpp, package = "lme4")
cbpp$trials <- cbpp$size - cbpp$incidence
m <- glm(cbind(incidence, trials) ~ period, data = cbpp, family = binomial)
head(get_data(m))
```

---

get_parameters *Get model parameters*

Description

Returns the point estimates (or posterior samples for Bayesian models) from a model.

Usage

```
get_parameters(x, ...)

## S3 method for class 'zeroinfl'
get_parameters(x, component = c("all", "conditional",
  "zi", "zero_inflated"), ...)

## S3 method for class 'hurdle'
get_parameters(x, component = c("all", "conditional",
  "zi", "zero_inflated"), ...)

## S3 method for class 'coxme'
get_parameters(x, effects = c("fixed", "random"), ...)

## S3 method for class 'merMod'
```

```
get_parameters(x, effects = c("fixed", "random"), ...)

## S3 method for class 'lme'
get_parameters(x, effects = c("fixed", "random"), ...)

## S3 method for class 'MixMod'
get_parameters(x, effects = c("fixed", "random"),
  component = c("all", "conditional", "zi", "zero_inflated",
  "dispersion"), ...)

## S3 method for class 'glmmTMB'
get_parameters(x, effects = c("fixed", "random"),
  component = c("all", "conditional", "zi", "zero_inflated",
  "dispersion"), ...)

## S3 method for class 'brmsfit'
get_parameters(x, effects = c("fixed", "random",
  "all"), component = c("all", "conditional", "zi", "zero_inflated",
  "dispersion"), ...)

## S3 method for class 'stanreg'
get_parameters(x, effects = c("fixed", "random",
  "all"), ...)

## S3 method for class 'stanmvreg'
get_parameters(x, effects = c("fixed", "random",
  "all"), ...)
```

Arguments

x	A fitted model.
...	Currently not used.
component	Should all predictor variables, predictor variables for the conditional model, the zero-inflated part of the model, the dispersion term or the instrumental variables be returned? Applies to models with zero-inflated and/or dispersion formula, or to models with instrumental variable (so called fixed-effects regressioms). May be abbreviated.
effects	Should variables for fixed effects, random effects or both be returned? Only applies to mixed models. May be abbreviated.

Value

For non-Bayesian models and if `effects = "fixed"`, a data frame with two columns: the parameter names and the related point estimates; if `effects = "random"`, a list with the random effects (as returned by `ranef()`. For Bayesian models, the posterior samples from the requested parameters as data frame. For Anova (`aov()`) with error term, a list of parameters for the conditional, the within-subject and the between-subjects parameters.

Examples

```
data(mtcars)
m <- lm(mpg ~ wt + cyl + vs, data = mtcars)
get_parameters(m)
```

---

get_predictors *Get the data from predictor variables*

Description

Returns the data from all predictor variables (fixed effects).

Usage

```
get_predictors(x)
```

Arguments

x	A fitted model.

Value

The data from all predictor variables, as data frame.

Examples

```
m <- lm(mpg ~ wt + cyl + vs, data = mtcars)
head(get_predictors(m))
```

---

get_random *Get the data from random effects terms*

Description

Returns the data from all random effects terms.

Usage

```
get_random(x)
```

Arguments

x	A fitted mixed model.

Value

The data from all random effects terms, as data frame. Or NULL if model has no random effects.

Examples

```
library(lme4)
data(sleepstudy)
# prepare some data...
sleepstudy$mygrp <- sample(1:5, size = 180, replace = TRUE)
sleepstudy$mysubgrp <- NA
for (i in 1:5) {
  filter_group <- sleepstudy$mygrp == i
  sleepstudy$mysubgrp[filter_group] <-
    sample(1:30, size = sum(filter_group), replace = TRUE)
}

m <- lmer(
  Reaction ~ Days + (1 | mygrp / mysubgrp) + (1 | Subject),
  data = sleepstudy
)

head(get_random(m))
```

---

get_response *Get the values from the response variable*

Description

Returns the values the response variable(s) from a model object. If the model is a multivariate response model, a data frame with values from all response variables is returned.

Usage

```
get_response(x, resp = NULL)
```

Arguments

x	A fitted model.
resp	Optional names of response variables for which to extract values.

Value

The values of the reponse variable, as vector, or a data frame if x has more than one defined response variable.

Examples

```
library(lme4)
data(cbpp)
data(mtcars)
cbpp$trials <- cbpp$size - cbpp$incidence

m <- glm(cbind(incidence, trials) ~ period, data = cbpp, family = binomial)
```

```
head(get_response(m))
get_response(m, resp = "incidence")

m <- lm(mpg ~ wt + cyl + vs, data = mtcars)
get_response(m)
```

`is_multivariate` *Checks if an object stems from a multivariate response model*

Description

to do...

Usage

```
is_multivariate(x)
```

Arguments

x	A model object, or an object returned by a function from this package.

Value

A logical, TRUE if either x is a model object and is a multivariate response model, or TRUE if a return value from a function of **insight** is from a multivariate response model.

Examples

```
## Not run:
library(rstanarm)
data("pbcLong")
model <- stan_mvmer(
  formula = list(
    logBili ~ year + (1 | id),
    albumin ~ sex + year + (year | id)
  ),
  data = pbcLong,
  chains = 1, cores = 1, seed = 12345, iter = 1000
)

f <- find_formula(model)
is_multivariate(model)
is_multivariate(f)

## End(Not run)
```

link_function *Get link-function from model object*

Description

Returns the link-function from a model object.

Usage

```
link_function(x, ...)
```

Arguments

x	A fitted model.
...	Currently not used.

Value

A function, describing the link-function from a model-object. For multivariate-response models, a list of functions is returned.

Examples

```
# example from ?stats::glm
counts <- c(18, 17, 15, 20, 10, 20, 25, 13, 12)
outcome <- gl(3, 1, 9)
treatment <- gl(3, 3)
m <- glm(counts ~ outcome + treatment, family = poisson())

link_function(m)(.3)
# same as
log(.3)
```

link_inverse *Get link-inverse function from model object*

Description

Returns the link-inverse function from a model object.

Usage

```
link_inverse(x, ...)
```

Arguments

x	A fitted model.
...	Currently not used.

Value

A function, describing the inverse-link function from a model-object. For multivariate-response models, a list of functions is returned.

Examples

```
# example from ?stats::glm
counts <- c(18, 17, 15, 20, 10, 20, 25, 13, 12)
outcome <- gl(3, 1, 9)
treatment <- gl(3, 3)
m <- glm(counts ~ outcome + treatment, family = poisson())

link_inverse(m)(.3)
# same as
exp(.3)
```

---

model_info *Access information from model objects*

Description

Retrieve information from model objects.

Usage

```
model_info(x, ...)
```

Arguments

x	A fitted model.
...	Currently not used.

Details

`model_info()` returns a list with information about the model for many different model objects. Following information is returned, where all values starting with `is_` are logicals.

- `is_binomial`: family is binomial (but not negative binomial)
- `is_poisson`: family is poisson
- `is_negbin`: family is negative binomial
- `is_count`: model is a count model (i.e. family is either poisson or negative binomial)

- is_beta: family is beta
- is_logit: model has logit link
- is_linear: family is gaussian
- is_ordinal: family is ordinal or cumulative link
- is_categorical: family is categorical link
- is_zeroinf: model has zero-inflation component
- is_mixed: model is a mixed effects model (with random effects)
- is_multivariate: model is a multivariate response model (currently only works for *brmsfit* objects)
- is_trial: model response contains additional information about the trials
- is_bayesian: model is a Bayesian model
- is_anova: model is an Anova object
- link_function: the link-function
- family: the family-object
- n_obs: number of observations
- model_terms: a list with all model terms, including terms such as random effects or from zero-inflated model parts.

Value

A list with information about the model, like family, link-function etc. (see 'Details').

Examples

```
library(glmmTMB)
data("Salamanders")
m <- glmmTMB(
  count ~ spp + cover + mined + (1 | site),
  ziformula = ~ spp + mined,
  dispformula = ~DOY,
  data = Salamanders,
  family = nbinom2
)

model_info(m)
```

---

n_obs *Get number of observations from a model*

Description

This method returns the number of observation that were used to fit the model, as numeric value.

Usage

```
n_obs(x, ...)
```

Arguments

x	A fitted model.
...	Currently not used.

Value

The number of observations used to fit the model, or NULL if this information is not available.

Examples

```
data(mtcars)
m <- lm(mpg ~ wt + cyl + vs, data = mtcars)
n_obs(m)
```

Index