Package: simglm (via r-universe)

September 5, 2024

Type Package

Version 0.9.20

Date 2024-02-27

License MIT + file LICENSE

Title Simulate Models Based on the Generalized Linear Model

Description Simulates regression models, including both simple
regression and generalized linear mixed models with up to three
level of nesting. Power simulations that are flexible allowing
the specification of missing data, unbalanced designs, and
different random error distributions are built into the
package.

Depends R (>= 4.1.0)

Imports stats, methods, rlang, dplyr, purrr, broom, future.apply,
gtools

Suggests knitr, lme4, nlme, testthat, shiny, e1071, ggplot2, tidyr,
geepack, rmarkdown, future, splines, covr

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.1

Author Brandon LeBeau [aut, cre]

Maintainer Brandon LeBeau <lebebr01+simglm@gmail.com>

URL https://github.com/lebebr01/simglm

BugReports https://github.com/lebebr01/simglm/issues

Repository https://lebebr01.r-universe.dev

RemoteUrl https://github.com/lebebr01/simglm

RemoteRef HEAD

RemoteSha 56e9a70ddfc381793edbb082bc3b9b7d36612860

Contents

---

compute_density_values

Convenience function for computing density values for plotting.

Description

Convenience function for computing density values for plotting.

Usage

```
compute_density_values(data, group_var, parameter, values)
```

Arguments

data	A dataframe that contains the parameter estimates.
group_var	A group variable that specifies the attributes to group by. By default, this would likely be the term attribute, but can contain more than one attribute.
parameter	The attribute that represents the parameter estimate.
values	A list of numeric vectors that specifies the values for which the density values are computed for.

---

compute_statistics	*Compute Power, Type I Error, or Precision Statistics*

Description

Compute Power, Type I Error, or Precision Statistics

Usage

```
compute_statistics(
  data,
  sim_args,
  power = TRUE,
  type_1_error = TRUE,
  precision = TRUE,
  alternative_power = FALSE,
  type_s_error = FALSE,
  type_m_error = FALSE
)
```

Arguments

data	A list of model results generated by [replicate_simulation](#) function.
sim_args	A named list with special model formula syntax. See details and examples for more information. The named list may contain the following: • fixed: This is the fixed portion of the model (i.e. covariates) • random: This is the random portion of the model (i.e. random effects) • error: This is the error (i.e. residual term).
power	TRUE/FALSE flag indicating whether power should be computed. Defaults to TRUE.
type_1_error	TRUE/FALSE flag indicating whether type I error rate should be computed. Defaults to TRUE.
precision	TRUE/FALSE flag indicating whether precision should be computed. Defaults to TRUE.

alternative_power

>   TRUE/FALSE flag indicating whether alternative power estimates should be computed. If TRUE, this must be accompanied by thresholds specified within the power simulation arguments. Defaults to FALSE.

type_s_error   TRUE/FALSE flag indicating whether Type S error should be computed. Defaults to FALSE.

type_m_error   TRUE/FALSE flag indicating whether Type M error should be computed. Defaults to FALSE.

---

correlate_variables   *Correlate elements*

Description

Correlate elements

Usage

```
correlate_variables(data, sim_args, ...)
```

Arguments

data   Data simulated from other functions to pass to this function.

sim_args   A named list with special model formula syntax. See details and examples for more information. The named list may contain the following:

>   - fixed: This is the fixed portion of the model (i.e. covariates)
>   - random: This is the random portion of the model (i.e. random effects)
>   - error: This is the error (i.e. residual term).
>   - correlate: These are the correlations for random effects and/or fixed effects.

...   Additional arguments, currently not used.

---

desireVar   *Computes mixture normal variance*

Description

Input the desired variance, number of distributions, and mean of the distributions, returns a value of the variance of each mixture distribution.

Usage

```
desireVar(desVar, num_dist, means, equalWeight = TRUE)
```

Arguments

desVar	Desired overall variance of mixture normal distribution.
num_dist	Number of normal distributions.
means	Vector of means for each normal distribution. Must equal num_dist.
equalWeight	Should equal weights be used, only TRUE is currently supported.

Details

This function can be used to generate the inputs for the [rbimod](#) variances when a specific variance is desired. Especially useful when attempting to simulate a mixture normal/bimodal distribution.

---

extract_coefficients   *Extract Coefficients*

Description

Extract Coefficients

Usage

```
extract_coefficients(model, extract_function = NULL)
```

Arguments

model	A returned model object from a fitted model.
extract_function	
	A function that extracts model results. The function must take the model object as the only argument.

---

generate_missing   *Tidy Missing Data Function*

Description

Tidy Missing Data Function

Usage

```
generate_missing(data, sim_args)
```

Arguments

`data`	Data simulated from other functions to pass to this function.
`sim_args`	A named list with special model formula syntax. See details and examples for more information. The named list may contain the following:

- fixed: This is the fixed portion of the model (i.e. covariates)
- random: This is the random portion of the model (i.e. random effects)
- error: This is the error (i.e. residual term).

`generate_response`	*Simulate response variable*

Description

Simulate response variable

Usage

```
generate_response(data, sim_args, keep_intermediate = TRUE, ...)
```

Arguments

`data`	Data simulated from other functions to pass to this function.
`sim_args`	A named list with special model formula syntax. See details and examples for more information. The named list may contain the following:

- fixed: This is the fixed portion of the model (i.e. covariates)
- random: This is the random portion of the model (i.e. random effects)
- error: This is the error (i.e. residual term).

`keep_intermediate`	
	TRUE/FALSE flag indicating whether intermediate steps should be kept. This would include fixed effects times regression weights, random effect summations, etc. Default is TRUE.
`...`	Other arguments to pass to error simulation functions.

---

missing_data   *Missing Data Functions*

Description

Function that inputs simulated data and returns data frame with new response variable that includes missing data. Missing data types incorporated include dropout missing data, missing at random, and random missing data.

Usage

```
missing_data(
  sim_data,
  resp_var = "sim_data",
  new_outcome = "sim_data2",
  clust_var = NULL,
  within_id = NULL,
  miss_prop = NULL,
  dropout_location = NULL,
  type = c("dropout", "random", "mar"),
  miss_cov,
  mar_prop
)

dropout_missing(
  sim_data,
  resp_var = "sim_data",
  new_outcome = "sim_data2",
  clust_var = "clustID",
  within_id = "withinID",
  miss_prop = NULL,
  dropout_location = NULL
)

random_missing(
  sim_data,
  resp_var = "sim_data",
  new_outcome = "sim_data2",
  miss_prop,
  clust_var = NULL,
  within_id = "withinID"
)

mar_missing(
  sim_data,
  resp_var = "sim_data",
  new_outcome = "sim_data2",
```

```
    miss_cov,
    mar_prop
)
```

Arguments

sim_data	Simulated data frame
resp_var	Character string of response variable with complete data.
new_outcome	Character string of new outcome variable name that includes the missing data.
clust_var	Cluster variable used for the grouping, set to NULL by default which means no clustering.
within_id	ID variable within each cluster.
miss_prop	Proportion of missing data overall
dropout_location	
	A vector the same length as the number of clusters representing the number of data observations for each individual.
type	The type of missing data to generate, currently supports dropout, random, or missing at random (mar) missing data.
miss_cov	Covariate that the missing values are based on.
mar_prop	Proportion of missing data for each unique value specified in the miss_cov argument.

---

model_fit	*Tidy Model Fitting Function*

Description

Tidy Model Fitting Function

Usage

```
model_fit(data, sim_args, ...)
```

Arguments

data	A data object, most likely generated from within simglm
sim_args	A named list with special model formula syntax. See details and examples for more information. The named list may contain the following:
	• fixed: This is the fixed portion of the model (i.e. covariates)
	• random: This is the random portion of the model (i.e. random effects)
	• error: This is the error (i.e. residual term).
	• model_fit: These are arguments passed to the [model_fit](#) function.
...	Currently not used.

---

parse_correlation	*Parse correlation arguments*

Description

This function is used to parse user specified correlation attributes. The correlation attributes need to be in a dataframe to be processed internally. Within the dataframe, there are expected to be 3 columns, 1) names of variable/attributes, 2) the variable/attribute pair for 1, 3) the correlation.

Usage

```
parse_correlation(sim_args)
```

Arguments

sim_args   A named list with special model formula syntax. See details and examples for more information. The named list may contain the following:

- fixed: This is the fixed portion of the model (i.e. covariates)
- random: This is the random portion of the model (i.e. random effects)
- error: This is the error (i.e. residual term).
- correlate: These are the correlations for random effects and/or fixed effects.

---

parse_formula	*Parses tidy formula simulation syntax*

Description

A function that parses the formula simulation syntax in order to simulate data.

Usage

```
parse_formula(sim_args)
```

Arguments

sim_args   A named list with special model formula syntax. See details and examples for more information. The named list may contain the following:

- fixed: This is the fixed portion of the model (i.e. covariates)
- random: This is the random portion of the model (i.e. random effects)
- error: This is the error (i.e. residual term).

---

parse_multiplemember   *Parse Multiple Membership Random Effects*

Description

Parse Multiple Membership Random Effects

Usage

```
parse_multiplemember(sim_args, random_formula_parsed)
```

Arguments

sim_args   Simulation arguments

random_formula_parsed

                This is the output from parse_randomeffect.

---

parse_power   *Parse power specifications*

Description

Parse power specifications

Usage

```
parse_power(sim_args, samp_size)
```

Arguments

sim_args   A named list with special model formula syntax. See details and examples for
                   more information. The named list may contain the following:

- fixed: This is the fixed portion of the model (i.e. covariates)
- random: This is the random portion of the model (i.e. random effects)
- error: This is the error (i.e. residual term).

samp_size   The sample size pulled from the simulation arguments or the power model re-
                   sults when vary_arguments is used.

---

parse_randomeffect *Parses random effect specification*

Description

Parses random effect specification

Usage

```
parse_randomeffect(formula)
```

Arguments

formula   Random effect formula already parsed by parse_formula

---

parse_varyarguments *Parse between varying arguments*

Description

Parse between varying arguments

Usage

```
parse_varyarguments(sim_args)
```

Arguments

sim_args   A named list with special model formula syntax. See details and examples for
               more information. The named list may contain the following:

- fixed: This is the fixed portion of the model (i.e. covariates)
- random: This is the random portion of the model (i.e. random effects)
- error: This is the error (i.e. residual term).

---

parse_varyarguments_w   *Parse within varying arguments*

Description

Parse within varying arguments

Usage

```
parse_varyarguments_w(sim_args, name)
```

Arguments

sim_args   A named list with special model formula syntax. See details and examples for
                more information. The named list may contain the following:

                • fixed: This is the fixed portion of the model (i.e. covariates)
                • random: This is the random portion of the model (i.e. random effects)
                • error: This is the error (i.e. residual term).

name   The name of the within simulation condition. This is primarily an internal func-
                tion.

---

rbimod   *Simulating mixture normal distributions*

Description

Input simulation metrics returns mixture normal random variable.

Usage

```
rbimod(n, mean, var, num_dist)
```

Arguments

n   Number of random draws. Optionally can be a vector with number in each
                simulated normal distribution.

mean   Vector of mean values for each normal distribution. Must be the same length as
                num_dist.

var   Vector of variance values for each normal distribution. Must be the same length
                as num_dist.

num_dist   Number of normal distributions to use when simulating mixture normal distri-
                bution.

Details

Function to simulate mixture normal distributions. The function computes adds the specified number of normal distributions into a single vector.

Use of the function desireVar can be used to generate a mixture normal distribution with a specific global variance.

---

replicate_simulation   *Replicate Simulation*

Description

Replicate Simulation

Usage

```
replicate_simulation(sim_args, return_list = FALSE, future.seed = TRUE, ...)
```

Arguments

sim_args	A named list with special model formula syntax. See details and examples for more information. The named list may contain the following:
	• fixed: This is the fixed portion of the model (i.e. covariates)
	• random: This is the random portion of the model (i.e. random effects)
	• error: This is the error (i.e. residual term).
return_list	TRUE/FALSE indicating whether a full list output should be returned. If TRUE, the nested list is returned. If FALSE, replications are combined with a replication id appended.
future.seed	TRUE/FALSE or numeric. Default value is true, see future_replicate.
...	Currently not used.

---

run_shiny   *Run Shiny Application Demo*

Description

Function runs Shiny Application Demo

Usage

```
run_shiny()
```

Details

This function does not take any arguments and will run the Shiny Application. If running from RStudio, will open the application in the viewer, otherwise will use the default internet browser.

---

simglm   *Single wrapper function*

Description

This function is most useful to pass to `replicate_simulation`. The function attempts to determine automatically which aspects to add to the simulation/power generation based on the elements found in the sim_args argument.

Usage

```
simglm(sim_args)
```

Arguments

sim_args   A named list with special model formula syntax. See details and examples for more information. The named list may contain the following:

- fixed: This is the fixed portion of the model (i.e. covariates)
- random: This is the random portion of the model (i.e. random effects)
- error: This is the error (i.e. residual term).

---

simulate_error   *Tidy error simulation*

Description

Tidy error simulation

Usage

```
simulate_error(data, sim_args, ...)
```

Arguments

data   Data simulated from other functions to pass to this function.

sim_args   A named list with special model formula syntax. See details and examples for more information. The named list may contain the following:

- fixed: This is the fixed portion of the model (i.e. covariates)
- random: This is the random portion of the model (i.e. random effects)
- error: This is the error (i.e. residual term).

...   Other arguments to pass to error simulation functions.

---

simulate_fixed   *Tidy fixed effect formula simulation*

Description

This function simulates the fixed portion of the model using a formula syntax.

Usage

```
simulate_fixed(data, sim_args, ...)
```

Arguments

data	Data simulated from other functions to pass to this function. Can pass NULL if first in simulation string.
sim_args	A named list with special model formula syntax. See details and examples for more information. The named list may contain the following:

- fixed: This is the fixed portion of the model (i.e. covariates)
- random: This is the random portion of the model (i.e. random effects)
- error: This is the error (i.e. residual term).

...	Other arguments to pass to error simulation functions.

---

simulate_heterogeneity

Tidy heterogeneity of variance simulation

Description

This function simulates heterogeneity of level one error variance.

Usage

```
simulate_heterogeneity(data, sim_args, ...)
```

Arguments

data	Data simulated from other functions to pass to this function. This function needs to be specified after 'simulate_fixed' and 'simulate_error'.
sim_args	A named list with special model formula syntax. See details and examples for more information. The named list may contain the following:

- fixed: This is the fixed portion of the model (i.e. covariates)
- random: This is the random portion of the model (i.e. random effects)
- error: This is the error (i.e. residual term).

...	Other arguments to pass to error simulation functions.

---

simulate_knot   *Simulate knot locations*

Description

Function that generates knot locations. An example of usefulness of this function would be with generation of interrupted time series data. Another application may be with simulation of piecewise linear data structures.

Usage

```
simulate_knot(data, sim_args)
```

Arguments

data	Mostly internal argument.
sim_args	A named list with special model formula syntax. See details and examples for more information. The named list may contain the following:

- fixed: This is the fixed portion of the model (i.e. covariates)
- random: This is the random portion of the model (i.e. random effects)
- error: This is the error (i.e. residual term).

---

simulate_randomeffect   *Tidy random effect formula simulation*

Description

This function simulates the random portion of the model using a formula syntax.

Usage

```
simulate_randomeffect(data, sim_args, ...)
```

Arguments

data	Data simulated from other functions to pass to this function. Can pass NULL if first in simulation string.
sim_args	A named list with special model formula syntax. See details and examples for more information. The named list may contain the following:

- fixed: This is the fixed portion of the model (i.e. covariates)
- random: This is the random portion of the model (i.e. random effects)
- error: This is the error (i.e. residual term).

...	Other arguments to pass to error simulation functions.

---

sim_continuous2 *Simulate continuous variables*

Description

Function that simulates continuous variables. Any distribution function in R is supported.

Usage

```
sim_continuous2(
  n,
  dist = "rnorm",
  var_level = 1,
  variance = NULL,
  ther_sim = FALSE,
  ther_val = NULL,
  ceiling = NULL,
  floor = NULL,
  ...
)
```

Arguments

n	A list of sample sizes.
dist	A distribution function. This argument takes a quoted R distribution function (e.g. 'rnorm'). Default is 'rnorm'.
var_level	The level the variable should be simulated at. This can either be 1, 2, or 3 specifying a level 1, level 2, or level 3 variable respectively.
variance	The variance for random effect simulation.
ther_sim	A TRUE/FALSE flag indicating whether the error simulation function should be simulated, that is should the mean and standard deviation used for standardization be simulated.
ther_val	A vector of 2 that should include the theoretical mean and standard deviation of the generating function.
ceiling	A numeric value that specifies the ceiling (maximum) of an attribute being generated. Defaults to NULL meaning no ceiling effect. If a value is specified, any data larger than integer is rounded to that ceiling value.
floor	A numeric value that specifies the floor (minimum) of an attribute being generated. Defaults to NULL meaning no floor effect. If a value is specified, any data larger than integer is rounded to that floor value.
...	Additional parameters to pass to the dist_fun argument.

---

sim_factor2   *Simulate categorical or factor variables*

Description

Function that simulates factor or categorical variables. Is essentially a wrapper around the sample function from base R.

Usage

```
sim_factor2(n, levels, var_level = 1, replace = TRUE, force_equal = FALSE, ...)
```

Arguments

n	A list of sample sizes.
levels	Scalar indicating the number of levels for categorical or factor variable. Can also specify levels as a character vector.
var_level	The level the variable should be simulated at. This can either be 1, 2, or 3 specifying a level 1, level 2, or level 3 variable respectively.
replace	TRUE/FALSE indicating whether levels should be sampled with replacement. Default is TRUE.
force_equal	TRUE/FALSE indicating if the sample size should be forced to be equal. Should not be used with the 'replace = FALSE' argument.
...	Additional parameters passed to the sample function.

---

sim_ordinal2   *Simulate discrete variables*

Description

Function that simulates discrete variables. Is essentially a wrapper around the sample function from base R.

Usage

```
sim_ordinal2(n, levels, var_level = 1, replace = TRUE, ...)
```

Arguments

`n`	A list of sample sizes.
`levels`	Scalar indicating the number of levels for discrete variable. Can also specify levels as a character vector.
`var_level`	The level the variable should be simulated at. This can either be 1, 2, or 3 specifying a level 1, level 2, or level 3 variable respectively.
`replace`	TRUE/FALSE indicating whether levels should be sampled with replacement. Default is TRUE.
`...`	Additional parameters passed to the sample function.

`sim_time`	*Simulate Time*

Description

This function simulates data for the time variable of longitudinal data.

Usage

```
sim_time(n, time_levels = NULL, ...)
```

Arguments

`n`	Sample size of the levels.
`time_levels`	The values the time variable should take. If NULL (default), the time values are discrete integers starting at 0 and going to n - 1.
`...`	Currently not used.

`transform_outcome`	*Transform response variable*

Description

Transform response variable

Usage

```
transform_outcome(outcome, type, categories = NULL, ...)
```

Arguments

`outcome`	The outcome variable to transform.
`type`	Type of transformation to apply.
`categories`	A vector of named categories for multinomial sim
`...`	Additional arguments passed to distribution functions.

Index