

Package: jglmm (via r-universe)

September 21, 2026

Type Package

Version 0.1.0.9001

Title Generalized Mixed-Effects Models in Julia

Description R package for interfacing with Julia's MixedModels library to fit generalized linear mixed-effects models, similar to the lme4 package in R
(<http://dmbates.github.io/MixedModels.jl/latest/>).

Encoding UTF-8

LazyData true

Depends R ($\geq 3.5.0$)

License GPL-3

URL <https://github.com/mikabr/jglmm>

BugReports <http://github.com/mikabr/jglmm/issues>

Imports dplyr ($\geq 0.7.4$), generics ($\geq 0.0.2$), glue ($\geq 1.3.0$), JuliaCall ($\geq 0.13.0$), purrr ($\geq 0.2.4$), lme4, rlang ($\geq 0.3.0$), stringr ($\geq 1.3.1$), tidyr

RoxygenNote 7.2.0

SystemRequirements Julia $\geq 1.5.0$, RCall.jl, DataFrames.jl, MixedModels.jl, StatsModels.jl

Suggests testthat ($\geq 3.0.0$)

Config/testthat/edition 3

Collate 'data.R' 'fit.R' 'extract.R' 'fit_stats.R' 'jglmm-package.R' 'predict.R' 'tidyvers.R'

Repository <https://mikabr.r-universe.dev>

Date/Publication 2022-11-11 19:09:14 UTC

RemoteUrl <https://github.com/mikabr/jglmm>

RemoteRef HEAD

RemoteSha 7378659a89511c768a06d2f3afe45754bf4868e2

Contents

cbpp	2
contra	3
d3	3
deviance.jglmm	4
dyestuff	4
dyestuff2	5
extractAIC.jglmm	5
fitted.jglmm	6
fixef.jglmm	7
grouseticks	7
hatvalues.jglmm	8
insteval	8
jglmm	9
jglmm_setup	10
jglmm_tidiers	10
kb07	12
logLik.jglmm	12
machines	13
ml1m	13
mmec	13
mrk17_exp1	14
nobs.jglmm	14
oxide	15
pastes	15
penicillin	15
predict.jglmm	16
ranef.jglmm	17
sigma.jglmm	17
simulate.jglmm	18
sleepstudy	19
vcov.jglmm	19
verbagg	20
Index	21

cbpp	cbpp
------	------

Description

cbpp

Usage

cbpp

Format

A tibble

contra	<i>contra</i>
--------	---------------

Description

contra

Usage

contra

Format

A tibble

d3	<i>d3</i>
----	-----------

Description

d3

Usage

d3

Format

A tibble

<code>deviance.jglmm</code>	<i>Model Deviance</i>
-----------------------------	-----------------------

Description

Model Deviance

Usage

```
## S3 method for class 'jglmm'
deviance(object, ...)
```

Arguments

<code>object</code>	An object of class ‘jglmm’, as returned by ‘jglmm()’.
<code>...</code>	Optional additional arguments, currently none are used.

Value

A numeric giving the deviance extracted from the fitted model.

Examples

```
## Not run:
jglmm_setup()
lm1 <- jglmm(Reaction ~ Days + (Days | Subject), lme4::sleepstudy)
deviance(lm1)

## End(Not run)
```

<code>dyestuff</code>	<i>dyestuff</i>
-----------------------	-----------------

Description

dyestuff

Usage

```
dyestuff
```

Format

A tibble

dyestuff2

*dyestuff2***Description**

dyestuff2

Usage

dyestuff2

Format

A tibble

extractAIC.jglmm

*Extract AIC from a Fitted Model***Description**

Extract AIC from a Fitted Model

Usage

```
## S3 method for class 'jglmm'
extractAIC(fit, scale = 0, k = 2, ...)
```

Arguments

<code>fit</code>	An object of class 'jglmm', as returned by 'jglmm'.
<code>scale</code>	Not currently used (see 'extractAIC' generic).
<code>k</code>	Numeric specifying the 'weight' of the <i>degrees of freedom</i> part in the AIC formula.
<code>...</code>	Optional additional arguments, currently none are used.

Value

A numeric vector of length 2, with first and second elements giving

- 'df' the 'degrees of freedom' for the fitted model in 'x'.
- 'AIC' the (generalized) Akaike Information Criterion for the fitted model in 'x'.

Examples

```
## Not run:
jglmm_setup()
lm1 <- jglmm(Reaction ~ Days + (Days | Subject), lme4::sleepstudy)
aic <- extractAIC(lm1)
bic <- extractAIC(lm1, k = log(nobs(lm1)))

## End(Not run)
```

fitted.jglmm	<i>Extract model fitted values</i>
--------------	------------------------------------

Description

Extract the fitted values from a ‘jglmm’ object.

Usage

```
## S3 method for class 'jglmm'
fitted(object, ...)
```

Arguments

<code>object</code>	An object of class ‘jglmm’, as returned by ‘jglmm()’.
<code>...</code>	Optional additional arguments, currently none are used.

Value

Vector of fitted values extracted from the model.

Examples

```
## Not run:
jglmm_setup()
cbpp <- dplyr::mutate(lme4::cbpp, prop = incidence / size)
gm <- jglmm(prop ~ period + (1 | herd), data = cbpp, family = "binomial",
            weights = cbpp$size)
fitted(gm)

## End(Not run)
```

fixef.jglmm	<i>Extract fixed-effects estimates</i>
-------------	--

Description

Extract the fixed-effects estimates from a ‘jglmm’ object.

Usage

```
## S3 method for class 'jglmm'
fixef(object, ...)
```

Arguments

object	An object of class ‘jglmm’, as returned by ‘jglmm()’.
...	Optional additional arguments, currently none are used.

Value

A named numeric vector of fixed-effects estimates.

Examples

```
## Not run:
jglmm_setup()
lm1 <- jglmm(Reaction ~ Days + (Days | Subject), lme4::sleepstudy)
fixef(lm1)

## End(Not run)
```

grouseticks	<i>grouseticks</i>
-------------	--------------------

Description

grouseticks

Usage

```
grouseticks
```

Format

A tibble

<code>hatvalues.jglmm</code>	<i>Diagonal elements of the hat matrix</i>
------------------------------	--

Description

Returns the values on the diagonal of the hat matrix, which is the matrix that transforms the response vector (minus any offset) into the fitted values (minus any offset). Note that this method should only be used for linear mixed models. It is not clear if the hat matrix concept even makes sense for generalized linear mixed models.

Usage

```
## S3 method for class 'jglmm'
hatvalues(model, ...)
```

Arguments

`model` An object of class ‘jglmm’, as returned by ‘jglmm()’.
`...` Optional additional arguments, currently none are used.

Value

A numeric vector containing the diagonal elements of the hat matrix.

Examples

```
## Not run:
jglmm_setup()
lm1 <- jglmm(Reaction ~ Days + (Days | Subject), lme4::sleepstudy)
hatvalues(lm1)

## End(Not run)
```

<code>insteval</code>	<i>insteval</i>
-----------------------	-----------------

Description

`insteval`

Usage

`insteval`

Format

A tibble

Description

Fitting Generalized Linear Mixed-Effects Models in Julia

Usage

```
jglmm(
  formula,
  data,
  family = "normal",
  link = NULL,
  weights = NULL,
  contrasts = NULL,
  REML = FALSE
)

## S3 method for class 'jglmm'
print(x, ...)

## S3 method for class 'jglmm'
summary(object, ...)
```

Arguments

formula	A two-sided linear formula object describing both the fixed-effects and random-effects part of the model, with the response on the left of a <code>~</code> operator and the terms, separated by <code>+</code> operators, on the right. Random-effects terms are distinguished by vertical bars (<code> </code>) separating expressions for design matrices from grouping factors.
data	A data frame containing the variables named in formula.
family	(optional) The distribution family for the response variable (defaults to "normal").
link	(optional) The model link function (defaults to "identity").
weights	(optional) A vector of prior case weights.
contrasts	(optional) A named list mapping column names of categorical variables in data to coding schemes (defaults to dummy coding all categorical variables).
REML	(optional) A logical indicating whether REML should be used instead of maximum likelihood (defaults to FALSE).
x	An object of class 'jglmm', as returned by <code>jglmm()</code> .
...	Optional additional arguments, currently none are used.
object	An object of class 'jglmm', as returned by <code>jglmm()</code> .

Value

An object of class ‘jglmm’.

Examples

```
## Not run:
jglmm_setup()
# linear model
lm1 <- jglmm(Reaction ~ Days + (Days | Subject), lme4::sleepstudy)

# logistic model
cbpp <- dplyr::mutate(lme4::cbpp, prop = incidence / size)
gm1 <- jglmm(prop ~ period + (1 | herd), data = cbpp, family = "binomial",
             weights = cbpp$size)
gm2 <- jglmm(prop ~ period + (1 | herd), data = cbpp, family = "binomial",
             weights = cbpp$size, contrasts = list(period = "effects"))

## End(Not run)
```

jglmm_setup

Set up Julia and required libraries

Description

Set up Julia and required libraries

Usage

```
jglmm_setup()
```

jglmm_tidiers

Tidying methods for jglmm models

Description

These methods tidy the coefficients and fitted values from ‘jglmm’ objects.

Usage

```
## S3 method for class 'jglmm'
glance(x, ...)

## S3 method for class 'jglmm'
tidy(x, ...)

## S3 method for class 'jglmm'
augment(x, ...)
```

Arguments

`x` An object of class 'jglmm', as returned by 'jglmm()'.
`...` Optional additional arguments, currently none are used.

Value

'glance' returns a data frame with one row and the columns:

'tidy' returns a data frame with one row for each estimated effect. It contains the columns:

<code>effect</code>	"fixed" for fixed effects, "ran_pars" for random effect parameters
<code>group</code>	the group within which the random effect is being estimated (NA for fixed effects)
<code>param</code>	parameter being estimated (<code>beta</code> for fixed effects, <code>sd</code> or <code>cor</code> for random effect parameters)
<code>term</code>	term being estimated
<code>estimate</code>	estimated coefficient
<code>std.error</code>	standard error
<code>statistic</code>	z-statistic (NA for modes)
<code>p.value</code>	p-value computed from z-statistic (NA for modes)

'augment' returns one row for each original observation, with these columns added:

<code>.fitted</code>	predicted values
<code>.resid</code>	residuals

Examples

```
## Not run:
jglmm_setup()
cbpp <- dplyr::mutate(lme4::cbpp, prop = incidence / size)
gm <- jglmm(prop ~ period + (1 | herd), data = cbpp, family = "binomial",
            weights = cbpp$size)
tidy(gm)
augment(gm)
glance(gm)

## End(Not run)
```

 kb07

kb07

Description

kb07

Usage

kb07

Format
 A tibble

logLik.jglmm

Extract Log-Likelihood

Description

Extract Log-Likelihood

Usage

```
## S3 method for class 'jglmm'
logLik(object, ...)
```

Arguments

<code>object</code>	An object of class 'jglmm', as returned by 'jglmm()'.
<code>...</code>	Optional additional arguments, currently none are used.

Value

An object of class `logLik`, a number with attribute `"df"` (degrees of freedom), giving the number of (estimated) parameters in the model.

Examples

```
## Not run:
jglmm_setup()
lm1 <- jglmm(Reaction ~ Days + (Days | Subject), lme4::sleepstudy)
logLik(lm1)

## End(Not run)
```

machines	<i>machines</i>
----------	-----------------

Description

machines

Usage

machines

Format

A tibble

ml1m	<i>ml1m</i>
------	-------------

Description

ml1m

Usage

ml1m

Format

A tibble

mmec	<i>mmec</i>
------	-------------

Description

mmec

Usage

mmec

Format

A tibble

mrk17_exp1

mrk17_exp1

Description

mrk17_exp1

Usage

mrk17_exp1

Format

A tibble

nobs.jglmm

Extract the Number of Observations from a Fit

Description

Extract the Number of Observations from a Fit

Usage

```
## S3 method for class 'jglmm'
nobs(object, ...)
```

Arguments

object An object of class 'jglmm', as returned by 'jglmm()'.
... Optional additional arguments, currently none are used.

Value

A numeric giving the number of observations.

Examples

```
## Not run:
jglmm_setup()
lm1 <- jglmm(Reaction ~ Days + (Days | Subject), lme4::sleepstudy)
nobs(lm1)

## End(Not run)
```

oxide	<i>oxide</i>
-------	--------------

Description

oxide

Usage

oxide

Format

A tibble

pastes	<i>pastes</i>
--------	---------------

Description

pastes

Usage

pastes

Format

A tibble

penicillin	<i>penicillin</i>
------------	-------------------

Description

penicillin

Usage

penicillin

Format

A tibble

predict.jglmm	<i>Predictions from a model</i>
---------------	---------------------------------

Description

Predictions from a model

Usage

```
## S3 method for class 'jglmm'
predict(object, newdata = NULL, type = "link", allow.new.levels = FALSE, ...)
```

Arguments

<code>object</code>	An object of class 'jglmm', as returned by 'jglmm'.
<code>newdata</code>	(optional) A dataframe of new data.
<code>type</code>	(optional) A character string - either "link" (default), where predictions are returned on the scale of the linear predictors, or "response", where predictions are returned on the scale of the response variable.
<code>allow.new.levels</code>	(optional) A logical indicating whether new random effects levels in newdata are allowed. If FALSE (default), such new values in newdata will trigger an error; if TRUE , then the prediction will use population-level values for data with previously unobserved values.
<code>...</code>	Optional additional arguments, currently none are used.

Value

A numeric vector of predicted values

Examples

```
## Not run:
jglmm_setup()
cbpp <- dplyr::mutate(lme4::cbpp, prop = incidence / size)
gm1 <- jglmm(prop ~ period + (1 | herd), data = cbpp, family = "binomial",
             weights = cbpp$size)
predict(gm1)
predict(gm1, type = "response")
newdata <- with(cbpp, expand.grid(period=unique(period), herd=unique(herd)))
predict(gm1, newdata)

## End(Not run)
```

ranef.jglmm	<i>Extract the modes of the random effects</i>
-------------	--

Description

Extract the conditional modes of the random effects from a ‘jglmm’ object.

Usage

```
## S3 method for class 'jglmm'
ranef(object, ...)
```

Arguments

object	An object of class ‘jglmm’, as returned by ‘jglmm()’.
...	Optional additional arguments, currently none are used.

Value

A list of data frames, one for each grouping factor for the random effects. The number of rows in the data frame is the number of levels of the grouping factor. The number of columns is the dimension of the random effect associated with each level of the factor. Each of the data frames has an attribute called “postVar”, which contains an array for each random-effects term with the variance-covariance matrices for each level of the grouping factor.

Examples

```
## Not run:
jglmm_setup()
cbpp <- dplyr::mutate(lme4::cbpp, prop = incidence / size)
gm <- jglmm(prop ~ period + (1 | herd), data = cbpp, family = "binomial",
            weights = cbpp$size)
ranef(gm)

## End(Not run)
```

sigma.jglmm	<i>Extract residual standard deviation ‘sigma’</i>
-------------	--

Description

Extract the estimated standard deviation of the errors, the “residual standard deviation” (also misnamed the “residual standard error”), from a ‘jglmm’ object.

Usage

```
## S3 method for class 'jglmm'
sigma(object, ...)
```

Arguments

object An object of class ‘jglmm’, as returned by ‘jglmm()’.
... Optional additional arguments, currently none are used.

Value

Estimate of σ , the standard deviation of the per-observation noise.

Examples

```
## Not run:
jglmm_setup()
lm1 <- jglmm(Reaction ~ Days + (Days | Subject), lme4::sleepstudy)
sigma(lm1)

## End(Not run)
```

simulate.jglmm	<i>Simulate responses from a model</i>
----------------	--

Description

Simulate responses from a model

Usage

```
## S3 method for class 'jglmm'
simulate(object, nsim = 1, seed = NULL, newdata = NULL, ...)
```

Arguments

object An object of class ‘jglmm’, as returned by ‘jglmm’.
nsim (optional) A positive integer indicating the number of responses to simulate
seed Not currently used (see ‘simulate’ generic).
newdata (optional) A dataframe of new data.
... Optional additional arguments, currently none are used.

Value

A numeric vector of simulated values

Examples

```
## Not run:
jglmm_setup()
cbpp <- dplyr::mutate(lme4::cbpp, prop = incidence / size)
gm1 <- jglmm(prop ~ period + (1 | herd), data = cbpp, family = "binomial",
             weights = cbpp$size)
simulate(gm1)
simulate(gm1, nsim = 5)
newdata <- with(cbpp, expand.grid(period=unique(period), herd=unique(herd)))
simulate(gm1, newdata = newdata)

## End(Not run)
```

sleepstudy

*sleepstudy***Description**

sleepstudy

Usage

sleepstudy

Format

A tibble

vcov.jglmm

*Calculate variance-covariance matrix for a fitted model object***Description**

Extract the variance-covariance matrix of the main parameters from a ‘jglmm’ object.

Usage

```
## S3 method for class 'jglmm'
vcov(object, ...)
```

Arguments

object	An object of class ‘jglmm’, as returned by ‘jglmm()’.
...	Optional additional arguments, currently none are used.

Value

A matrix of the estimated covariances between the parameter estimates in the linear or non-linear predictor of the model.

Examples

```
## Not run:
jglmm_setup()
lm1 <- jglmm(Reaction ~ Days + (Days | Subject), lme4::sleepstudy)
vcov(lm1)

## End(Not run)
```

verbagg	<i>verbagg</i>
---------	----------------

Description

verbagg

Usage

verbagg

Format

A tibble

Index

- * datasets
 - cbpp, [2](#)
 - contra, [3](#)
 - d3, [3](#)
 - dyestuff, [4](#)
 - dyestuff2, [5](#)
 - grouseticks, [7](#)
 - insteval, [8](#)
 - kb07, [12](#)
 - machines, [13](#)
 - ml1m, [13](#)
 - mmec, [13](#)
 - mrk17_exp1, [14](#)
 - oxide, [15](#)
 - pastes, [15](#)
 - penicillin, [15](#)
 - sleepstudy, [19](#)
 - verbag, [20](#)
- augment.jglmm (*jglmm_tidiers*), [10](#)
- cbpp, [2](#)
- contra, [3](#)
- d3, [3](#)
- deviance.jglmm, [4](#)
- dyestuff, [4](#)
- dyestuff2, [5](#)
- extractAIC.jglmm, [5](#)
- fitted.jglmm, [6](#)
- fixef.jglmm, [7](#)
- glance.jglmm (*jglmm_tidiers*), [10](#)
- grouseticks, [7](#)
- hatvalues.jglmm, [8](#)
- insteval, [8](#)
- jglmm, [9](#)
- jglmm_setup, [10](#)
- jglmm_tidiers, [10](#)
- kb07, [12](#)
- logLik.jglmm, [12](#)
- machines, [13](#)
- ml1m, [13](#)
- mmec, [13](#)
- mrk17_exp1, [14](#)
- nobs.jglmm, [14](#)
- oxide, [15](#)
- pastes, [15](#)
- penicillin, [15](#)
- predict.jglmm, [16](#)
- print.jglmm (*jglmm*), [9](#)
- ranef.jglmm, [17](#)
- sigma.jglmm, [17](#)
- simulate.jglmm, [18](#)
- sleepstudy, [19](#)
- summary.jglmm (*jglmm*), [9](#)
- tidy.jglmm (*jglmm_tidiers*), [10](#)
- vcov.jglmm, [19](#)
- verbag, [20](#)