Package: jlme (via r-universe)

October 22, 2026

Title Regression Modelling with 'GLM.jl' and 'MixedModels.jl' in 'Julia'

Version 0.4.2

Description Bindings to 'Julia' packages 'GLM.jl' <doi:10.5281/zenodo.3376013> and 'MixedModels.jl' <doi:10.5281/zenodo.12575371>, powered by 'JuliaConnectoR'. Fits (generalized) linear (mixed-effects) regression models in 'Julia' using familiar model fitting syntax from R. Offers 'broom'-style data frame summary functionalities for 'Julia' regression models.

License MIT + file LICENSE

URL https://yjunechoe.github.io/jlme/, https://github.com/yjunechoe/jlme/

BugReports https://github.com/yjunechoe/jlme/issues/

Depends R (>= 4.1)

Imports generics, JuliaConnectoR (>= 1.1.4), JuliaFormulae, MASS, stats, utils

Suggests broom, broom.mixed, lme4, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

SystemRequirements Julia (>= 1.8)

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

Date/Publication 2025-05-06 18:43:13 UTC

RemoteUrl https://github.com/yjunechoe/jlme

RemoteRef HEAD

RemoteSha b305910eeacf920b184beee6a6c5df8ca5e72f9e

Contents

check_julia_ok 2
jlm 3
jlme-model-reexports 4
parametricbootstrap 5
profilelikelihood 6
tidy.jlmeboot 7

Index **9**

`check_julia_ok` *Set up Julia connection for jlme*

Description

Set up Julia connection for jlme

Usage

```
check_julia_ok()

stop_julia()

jlme_status()

jlme_setup(
  ...,
  add = NULL,
  restart = FALSE,
  threads = NULL,
  verbose = interactive()
)
```

Arguments

`...`	Unused
`add`	A character vector of additional Julia packages to add and load.
`restart`	Whether to run `stop_julia()` first, before attempting setup
`threads`	Number of threads to start Julia with. Defaults to `1`
`verbose`	Whether to alert setup progress. Defaults to `interactive()`

Value

Invisibly returns `TRUE` on success

Examples

```
# Check whether Julia installation meets requirements
check_julia_ok()

# Connect to a Julia runtime for use with `{jlme}`
jlme_setup()

# Show information about the Julia runtime
jlme_status()

# Stop Julia runtime
stop_julia()
```

`jlm` *Fit a (mixed-effects) regression model in Julia*

Description

Fit a (mixed-effects) regression model in Julia

Usage

```
jlm(formula, data, family = "gaussian", contrasts = jl_contrasts(data), ...)

jlmer(
  formula,
  data,
  family = NULL,
  contrasts = jl_contrasts(data),
  ...,
  fit = TRUE,
  optsum = list(),
  progress = interactive()
)
```

Arguments

`formula`	A formula written in Julia syntax. Can be a string or a language object.
`data`	A data frame
`family`	A distribution family
`contrasts`	A Julia dictionary of contrasts Inferred from `data` by default.
`...`	Additional arguments to the `fit()` function called in Julia
`fit`	Whether to fit the model. If `FALSE`, returns the unfit model object.

optsum	A list of values to set for the optimizer. See $optsum of unfit model for possible options.
progress	Whether to print model fitting progress. Defaults to interactive()

Value

A julia model object of class jlme

Examples

```
jlme_setup(restart = TRUE)

# Fixed effects models
lm(mpg ~ hp, mtcars)
jlm(mpg ~ hp, mtcars)

# Auto-handling of contrasts
x <- mtcars
x$cyl_helm <- factor(x$cyl)
contrasts(x$cyl_helm) <- contr.helmert(3)
colnames(contrasts(x$cyl_helm)) <- c("4vs6", "4&6vs8")
lm(mpg ~ cyl_helm, x)
jlm(mpg ~ cyl_helm, x)

# Mixed effects models
library(lme4)

glmer(r2 ~ Anger + Gender + (1 | id), VerbAgg, family = "binomial")
jlmer(r2 ~ Anger + Gender + (1 | id), VerbAgg, family = "binomial")

# Set optimizer options via `optsum`
jlmer(
  r2 ~ Anger + Gender + (1 | id), VerbAgg, family = "binomial",
  optsum = list(
    optimizer = jl(":LN_NELDERMEAD"),
    maxfeval = 10L
  )
)

stop_julia()
```

---

jlme-model-reexports *Re-exported functions for interacting with Julia model objects*

Description

Re-exported functions for interacting with Julia model objects

Usage

```
propertynames(x)

issingular(x)

likelihoodratiotest(x, ...)
```

Arguments

x	Julia model object
...	Additional arguments passed to the Julia function

Value

An appropriate R or Julia object

Examples

```
jlme_setup(restart = TRUE)

x <- jlmer(r2 ~ Anger + (1 | id), lme4::VerbAgg, family = "binomial")

# `propertynames()` lists properties accessible via `$`
propertynames(x)

# `issingular()` reports whether model has singular fit
issingular(x)

# `likelihoodratiotest()` conducts a likelihood-ratio test between nested models
likelihoodratiotest(
  x,
  jlmer(r2 ~ 1 + (1 | id), lme4::VerbAgg, family = "binomial")
)

stop_julia()
```

---

parametricbootstrap *Parametric bootstrap for Julia mixed effects models*

Description

Parametric bootstrap for Julia mixed effects models

Usage

```
parametricbootstrap(
  x,
  nsim,
  seed,
  ...,
  optsum_overrides = list(ftol_rel = 1e-08)
)
```

Arguments

x	A Julia MixedModel of class jlme
nsim	Number of simulations
seed	Seed for the random number generator (Random.MersenneTwister)
...	Not implemented
optsum_overrides	Values to override in the OptSummary.

Value

MixedModels.parametricboostrap() output as object of class jlmeboot

Examples

```
jlme_setup(restart = TRUE)

jmod <- jlmer(Reaction ~ Days + (Days | Subject), lme4::sleepstudy)
tidy(jmod)

samp <- parametricbootstrap(jmod, nsim = 100L, seed = 42L)
samp

tidy(samp)

stop_julia()
```

---

profilelikelihood *Profile the likelihood surface of Julia mixed effects models*

Description

Profile the likelihood surface of Julia mixed effects models

Usage

```
profilelikelihood(x, ...)
```

Arguments

x	A Julia MixedModel of class jlme
...	Not implemented

Value

MixedModels.profile() output as object of class jlmeprof

Examples

```
jlme_setup(restart = TRUE)

jmod <- jlmer(Reaction ~ Days + (Days | Subject), lme4::sleepstudy)
tidy(jmod)

prof <- profilelikelihood(jmod)
prof

tidy(prof)

stop_julia()
```

---

tidy.jlmeboot *Tidier methods for Julia regression models*

Description

Tidier methods for Julia regression models

Usage

```
## S3 method for class 'jlmeboot'
tidy(x, effects = c("var_model", "ran_pars", "fixed"), ...)

## S3 method for class 'jlmeprof'
tidy(x, effects = c("var_model", "ran_pars", "fixed"), ...)

## S3 method for class 'jlme'
tidy(x, effects = c("var_model", "ran_pars", "fixed"), ...)

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

Arguments

x	An object of class jlme
effects	One of "var_model", "ran_pars", or "fixed"
...	Unused

Value

A data frame

Index

check_julia_ok, 2

glance.jlme *(*tidy.jlmeboot*)*, 7

issingular *(*jlme-model-reexports*)*, 4

jlm, 3
jlme-model-reexports, 4
jlme_setup *(*check_julia_ok*)*, 2
jlme_status *(*check_julia_ok*)*, 2
jlme_tidiers *(*tidy.jlmeboot*)*, 7
jlmer *(*jlm*)*, 3

likelihoodratiotest
        *(*jlme-model-reexports*)*, 4

parametricbootstrap, 5
profilelikelihood, 6
propertynames *(*jlme-model-reexports*)*, 4

stop_julia *(*check_julia_ok*)*, 2

tidy.jlme *(*tidy.jlmeboot*)*, 7
tidy.jlmeboot, 7
tidy.jlmeprof *(*tidy.jlmeboot*)*, 7