Package: reconf (via r-universe)

October 21, 2026

Title Confidence Intervals for Random Effect Parameters in Linear Mixed Models

Version 0.2.0

Description Computes score-based confidence intervals and hypothesis tests for variance and covariance parameters of random effects in linear mixed models fitted with 'lme4'. The intervals invert a profile score statistic on the variance-covariance scale, with nuisance parameters estimated on an extended parameter set. The intervals have near-nominal coverage uniformly in the parameter, including near the boundary of the parameter space, where Wald and likelihood ratio intervals can be unreliable. The methods are described in Shedden and Ekvall (2026) <doi:10.48550/arXiv.2610.04181>, building on Ekvall and Bottai (2026) <doi:10.1093/biomet/asaf079>, Zhang, Ekvall, and Molstad (2025) <doi:10.1093/biomet/asaf010>, and Ekvall and Bottai (2022) <doi:10.1214/22-AOS2177>.

URL https://github.com/koekvall/reconf

BugReports https://github.com/koekvall/reconf/issues

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Depends R (>= 3.5.0)

LazyData true

Imports Matrix (>= 1.5-0), lme4 (>= 1.1-8), Rcpp, trust, methods, generics

LinkingTo Rcpp, RcppEigen

Suggests testthat (>= 3.0.0), numDeriv, knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author Karl Oskar Ekvall [aut, cre, cph], Yiqiao Zhang [ctb], Matias Shedden [ctb]

Maintainer Karl Oskar Ekvall <k.o.ekvall@gmail.com>

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

Date/Publication 2026-10-21 14:50:02 UTC

RemoteUrl https://github.com/cran/reconf

RemoteRef HEAD

RemoteSha 5742be2116a45f491aee0ab9528fb0903b1f7de0

Contents

fev1 2
print.vc_ci 3
tidy.vc_ci 3
vc_ci 4
vc_loglik 6
vc_model.lmerMod 7
vc_test 8

Index **10**

`fev1` *FEV1 data from the Six Cities Study*

Description

Repeated measurements of forced expiratory volume in one second (FEV1) on a random sample of 300 girls from Topeka, Kansas, in the Six Cities Study of Air Pollution and Health, with one to twelve measurements per girl.

Usage

```
fev1
```

Format

A data frame with 1994 rows and 6 variables:

id subject identifier.

ht height (meters).

age age (years).

baseht height at the first measurement (meters).

baseage age at the first measurement (years).

logfev1 natural logarithm of FEV1 (liters).

Source

Fitzmaurice, G. M., Laird, N. M., and Ware, J. H. (2011). *Applied Longitudinal Analysis*, 2nd edition. Wiley. https://content.sph.harvard.edu/fitzmaur/ala2e/. Data courtesy of D. W. Dockery.

References

Dockery, D. W., Berkey, C. S., Ware, J. H., Speizer, F. E., and Ferris, B. G. (1983). Distribution of FVC and FEV1 in children 6 to 11 years old. *American Review of Respiratory Disease* 128, 405–412.

---

print.vc_ci — *Print method for confidence intervals*

Description

Print method for confidence intervals

Usage

```
## S3 method for class 'vc_ci'
print(x, digits = 4, ...)
```

Arguments

x	A vc_ci object from vc_ci.
digits	Number of significant digits. Default 4.
...	Unused.

Value

x, invisibly.

---

tidy.vc_ci — *Tidy confidence intervals*

Description

Returns the intervals as a data frame with the column names of the broom package.

Usage

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

Arguments

x	A vc_ci object from vc_ci.
...	Unused.

Value

A data frame with columns term, estimate, conf.low, and conf.high.

---

vc_ci *Confidence intervals for covariance parameters*

Description

Computes confidence intervals for covariance parameters of a linear mixed model by inverting a signed profile score or likelihood ratio statistic. See the references for the theory.

Usage

```
vc_ci(
  object,
  parm = NULL,
  level = 0.95,
  statistic = c("score", "rlrt"),
  expected = TRUE,
  known = NULL,
  nonneg = TRUE,
  onestep = FALSE,
  step_size = NULL,
  num_points = 500L,
  method = c("auto", "q_side", "n_side", "spectral"),
  ...
)
```

Arguments

object	a fitted linear mixed model of class lmerMod from lme4::lmer, or a vc_model.
parm	parameters, as names (see Value) or as indices in the order of vc_model, which for an lme4 fit is that of as.data.frame(VarCorr(fit), order = "lower.tri"), with the error variance last. If NULL, all covariance parameters not in known.
level	confidence level.
statistic	"score" for the signed profile score statistic, standardized by the efficient information, or "rlrt" for the signed root of the profile likelihood ratio statistic, which for a restricted likelihood fit is the restricted likelihood ratio. See Details.
expected	if TRUE use the expected information, otherwise the observed information, both in the statistic and in the maximization over nuisance parameters.

known	parameters, as for parm, held fixed at their estimates.
nonneg	if TRUE, the interval for a variance, the error variance included, is intersected with $[0, \infty)$. An empty intersection gives NA bounds and a warning.
onestep	if TRUE, the nuisance parameters at each step are one iteration of the optimizer, with unbounded trust region, from those at the previous step. Requires statistic = "score".
step_size	initial step length of the search (see Details), on the scale of the parameter. If NULL, one fortieth of the standard error from the expected information at the estimate.
num_points	maximum number of steps in each direction. The search also stops at distance num_points * step_size from its origin.
method	computational method for the likelihood: "auto", "q_side", "n_side", or "spectral". See Details. A vc_model carries the method it was built with.
...	arguments passed to the optimizer trust: rinit, rmax, iterlim, fterm, and mterm; and warn_nonconv, which if FALSE suppresses the warning when the optimizer does not converge. The default iterlim is 1000.

Details

The interval at level $1 - \alpha$ for a parameter ψ_j is

$$\{\psi_j : |T(\psi_j)| \leq z_{1-\alpha/2}\},$$

where T is the signed statistic with the other parameters profiled out and $z_{1-\alpha/2}$ is the standard normal quantile. The parameter set is the set of covariance parameters for which the covariance matrix of the response is positive definite. It includes negative variances, and the nuisance parameters are maximized over it.

The bounds are found by a search outward from the estimate in each direction, with the nuisance parameters at each step computed from those at the previous step. Step lengths are chosen by secant extrapolation of the statistic, each at most twice the previous one. If the covariance matrix of the response is not positive definite at the start of a step, the step is halved, up to 20 times, after which a starting point is constructed as in vc_test. A bound is where the statistic crosses $\pm z_{1-\alpha/2}$, located to within step_size. If there is no crossing within the search distance, the bound is -Inf or Inf and a warning is issued, reporting the statistic at the end of the search. As a variance grows, the statistic tends to a finite limit, which for a grouping factor with few levels can be above $-z_{1-\alpha/2}$; the upper bound is then infinite for any search distance. If the statistic at the estimate exceeds the critical value, the search starts instead from the maximizer over the parameter set, with a warning.

With statistic = "rlrt", the search starts from the maximizer over the parameter set. A warning is issued if that maximizer has a negative variance; the interval for that variance can then lie entirely below zero.

The likelihood is computed by one of three methods. "q_side" uses sparse $q \times q$ matrices, where q is the number of random effects. "n_side" uses dense $n \times n$ matrices. For models with one random-effect variance, "spectral" uses one eigendecomposition and $O(n)$ operations per evaluation. "auto" uses "spectral" or "n_side" if $q \geq n$ and more than 10 percent of the entries of the random-effects design matrix are nonzero, and "q_side" otherwise.

Value

A matrix of class `vc_ci` with one row per parameter and columns `estimate`, `lower`, and `upper`. For an lme4 fit the estimates are those of lme4, and rows are named `"var_x|g"` for the variance of the random effect `x` with grouping factor `g`, `"cov_y.x|g"` for the covariance of `x` and `y`, and `"var_Residual"` for the error variance, following the names `lme4` uses for standard deviations and correlations.

References

Shedden, M. and Ekvall, K. O. (2026). Score-based confidence intervals for variance-covariance parameters in linear mixed models. arXiv:2610.04181.

See Also

`vc_model`, `vc_test`

Examples

```
library(lme4)
fit <- lmer(Reaction ~ Days + (Days | Subject), data = sleepstudy)

# 95% intervals for all covariance parameters
vc_ci(fit)

# The random slope variance, by name or by index
vc_ci(fit, parm = "var_Days|Subject")
vc_ci(fit, parm = 3)
```

`vc_loglik` *Log-likelihood as a function of the covariance parameters*

Description

Returns the log-likelihood of a linear mixed model, or its restricted log-likelihood for a restricted likelihood fit, as a function of the covariance parameters.

Usage

```
vc_loglik(object, method = c("auto", "q_side", "n_side", "spectral"))
```

Arguments

`object`	a fitted linear mixed model of class `lmerMod` from `lme4::lmer`, or a `vc_model`.
`method`	computational method for the likelihood: `"auto"`, `"q_side"`, `"n_side"`, or `"spectral"`. See Details. A `vc_model` carries the method it was built with.

Details

The likelihood is maximized over the fixed effects. Outside the parameter set, the set of covariance parameters for which the covariance matrix of the response is positive definite, the value is `-Inf` and the score and information are zero. With prior weights w_i in an lme4 fit, the value differs from `logLik()` by $\sum_i \log(w_i)/2$.

Value

A function of `psi`, the covariance parameters in the order of `vc_model`, with logical arguments `get_val`, `get_score`, and `get_inf` selecting what to compute and `expected` selecting the expected or the observed information, all `TRUE` by default. It returns a list with components `value`, `score`, and `information`, the last two named by the parameters; components not requested are `NULL`.

Examples

```
library(lme4)
fit <- lmer(Reaction ~ Days + (Days | Subject), data = sleepstudy)
ll <- vc_loglik(fit)
ll(c(600, 10, 35, 650))
```

`vc_model.lmerMod` *Model for inference on covariance parameters*

Description

Builds the object that `vc_ci`, `vc_test`, and `vc_loglik` work with: the likelihood of a linear mixed model as a function of its covariance parameters, with quantities precomputed once, and the parameter names and estimates. Those functions also accept a fitted model and build this object themselves; building it once avoids repeating the precomputation across calls.

Usage

```
## S3 method for class 'lmerMod'
vc_model(x, method = c("auto", "q_side", "n_side", "spectral"), ...)

vc_model(x, ...)

## S3 method for class 'vc_model'
print(x, digits = 4, ...)
```

Arguments

`x`	a fitted linear mixed model of class `lmerMod` from `lme4::lmer`; for `print`, a `vc_model`.
`method`	computational method for the likelihood; see `vc_ci`.
`...`	unused.
`digits`	number of significant digits printed.

Details

For an lme4 fit the likelihood is the restricted likelihood if the model was fitted by REML, and the estimates are those of lme4. The parameters are ordered as in `as.data.frame(VarCorr(x), order = "lower.tri")`, with the error variance last. Prior weights w_i and offsets are supported; as in `lme4`, observation i then has error variance ψ_r/w_i, with ψ_r the error variance parameter.

Value

An object of class `"vc_model"`. Printing it shows the covariance parameters, in the order that indices in `vc_ci` and the other functions refer to, and their estimates.

Examples

```
library(lme4)
fit <- lmer(Reaction ~ Days + (Days | Subject), data = sleepstudy)
m <- vc_model(fit)
m
vc_ci(m, parm = "var_Days|Subject")
vc_test(m, parm = "var_Days|Subject")
```

`vc_test` *Score test for covariance parameters*

Description

Computes the score test statistic for a hypothesis about the covariance parameters of a linear mixed model.

Usage

```
vc_test(
  object,
  parm = NULL,
  null_value = 0,
  expected = TRUE,
  known = NULL,
  method = c("auto", "q_side", "n_side", "spectral"),
  ...
)
```

Arguments

`object`	a fitted linear mixed model of class `lmerMod` from `lme4::lmer`, or a `vc_model`.
`parm`	the tested parameters, as names or indices as in `vc_ci`. If `NULL`, all random-effect covariance parameters not in `known`.
`null_value`	values of the tested parameters under the null hypothesis, recycled to the length of `parm`.

expected	if TRUE use the expected information, otherwise the observed information, both in the statistic and in the maximization over nuisance parameters.
known	parameters, as for parm, held fixed at their estimates.
method	computational method for the likelihood: "auto", "q_side", "n_side", or "spectral". See Details. A vc_model carries the method it was built with.
...	arguments passed to the optimizer trust: rinit, rmax, iterlim, fterm, and mterm; and warn_nonconv, which if FALSE suppresses the warning when the optimizer does not converge. The default iterlim is 1000.

Details

The statistic is the profile score statistic: the score for the tested parameters, evaluated at the maximizer of the likelihood with the tested parameters at their null values and the parameters in known at their estimates, and standardized by the efficient information. The maximization is over the other covariance parameters and, for a maximum likelihood fit, the fixed effects. It starts from the estimates in the model, with the covariances of a random effect whose variance is tested at zero set to zero. If the covariance matrix of the response is not positive definite at this start, free entries of Ψ are changed to make its blocks weakly diagonally dominant, and the error variance is doubled until it is.

Value

An object of class "htest" with components statistic; parameter, the degrees of freedom, which is the number of tested parameters; p.value, from the chi-squared distribution; null.value, the null values named as in vc_ci; alternative; method; and data.name.

References

Shedden, M. and Ekvall, K. O. (2026). Score-based confidence intervals for variance-covariance parameters in linear mixed models. arXiv:2610.04181.

Examples

```
library(lme4)
fit <- lmer(Reaction ~ Days + (Days | Subject), data = sleepstudy)

# All random-effect parameters zero
vc_test(fit)

# Random slope variance zero
vc_test(fit, parm = "var_Days|Subject")
```

Index

∗ **datasets**
    fev1, 2

fev1, 2

print.vc_ci, 3
print.vc_model *(*vc_model.lmerMod*)*, 7

tidy.vc_ci, 3
trust, *5*, *9*

vc_ci, *3*, 4, *4*, *7–9*
vc_loglik, 6, *7*
vc_model, *4*, *6–8*
vc_model *(*vc_model.lmerMod*)*, 7
vc_model.lmerMod, 7
vc_test, *5–7*, 8