Package 'reintervals'

November 20, 2026

Title Confidence, Prediction, Tolerance, and New-Group-Mean Intervals for Linear Mixed Models

Version 0.2.0

Description Confidence, prediction, tolerance, and new-group-mean (also called CInew or new-batch-mean) intervals for linear mixed models fitted with 'lme4' ('lmerMod' objects), returned through one uniform interface. Several validated interval engines are provided and selected automatically for the fitted model, with a manual override: an expected-mean-square, modified large-sample construction for random-intercept models (Francq, Lin and Hoyer (2019) <doi:10.1002/sim.8386>); a restricted-maximum-likelihood closed form for random-slope models; an unweighted lot-mean analysis-of-variance closed form for unbalanced single-observation fixed-slope stability designs (Montes, Burdick and Leblond (2019) <doi:10.5731/pdajpst.2018.008839>) and, without a covariate, for the one-way random-effects model (Hoffman and Kringle (2005) <doi:10.1081/BIP-200048826>); and a generalized-pivotal-quantity Monte Carlo engine (Oliva-Aviles and Hauser (2025) <doi:10.1080/00401706.2024.2407324>). Intervals are two-sided or one-sided; one-sided tolerance limits follow Hoffman (2010) <doi:10.1198/TECH.2010.09171>. General fixed effects are supported for random-intercept models; random-slope models are supported for a single continuous covariate (the stability design). The methods serve general linear-mixed-model interval estimation, with particular support for pharmaceutical stability and shelf-life work and for bioassay applications through the tolerance and new-group-mean intervals.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

LazyData true

Depends R (>= 4.1)

Imports lme4, Matrix, numDeriv, stats, withr (>= 2.5.0)

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

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/percevalsondag/reintervals

BugReports https://github.com/percevalsondag/reintervals/issues

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Perceval Sondag [aut, cre, cph] (ORCID:
<https://orcid.org/0009-0001-9183-6991>)

Maintainer Perceval Sondag <reintervals@gmail.com>

Repository CRAN

Date/Publication 2026-11-20 13:00:02 UTC

Contents

ci_lmm 2
hk_assay 4
lmm_interval_control 5
lmm_predict 6
mbl_appendix_a 8
new_group_mean_lmm 9
oliva_hauser_dgp 11
pi_lmm 12
reintervals-models 13
ti_anova_raw 15
ti_gpq_raw 17
ti_lmm 18

Index **22**

`ci_lmm` *Confidence interval for the mean of a population, from a linear mixed model*

Description

One of the four interval verbs (`ci_lmm()`, `pi_lmm()`, `ti_lmm()`, `new_group_mean_lmm()`). `ci_lmm()` gives a confidence interval for the mean response at `newdata`. It dispatches by random structure: random-intercept designs (including a fixed slope) use the expected-mean-square engine; random-slope designs use the REML closed form. See reintervals and `vignette("choosing-an-interval")` for the engine map, reintervals-models for the design tags, and `lmm_predict()` for the unified entry point.

Usage

```
ci_lmm(
  model,
  newdata = NULL,
  alpha = NULL,
  level = NULL,
  sides = c("two.sided", "lower", "upper"),
  method = "auto",
  control = lmm_interval_control()
)
```

Arguments

model	A fitted lmerMod (from lme4::lmer()).
newdata	Row(s) giving the fixed-effect combination(s) to evaluate at. NULL uses the first model-frame row.
alpha, level	The interval confidence, as a mutually-exclusive pair: give at most one. level is 1 - alpha; the default is alpha = 0.05 (95%).
sides	"two.sided" (default), "lower", or "upper". A one-sided interval keeps the two-sided interval's center, variance, and df and uses t_{1 - alpha} in place of t_{1 - alpha/2}.
method	"auto" (default) chooses the algorithm by structure; one of "ems-mls", "reml-mls", "anova-mls", "gpq" forces that engine and errors if it is incompatible with the model.
control	An lmm_interval_control() object (the GPQ seed and M, and the confidence-interval df rule ci_df).

Value

An lmm_interval object (a data frame), one row per evaluation point, with columns, in order:

- the eval-point column(s), named by the actual model predictor(s) (e.g. the time covariate for a slope model, or each newdata column);
- type ("ci", "pi", "ti", or "new_group_mean"), estimate, lower, upper;
- level (the confidence, 1 - alpha); P (content proportion — the value for ti, NA for the coverage intervals); sides;
- method (the engine: "ems-mls", "reml-mls", "anova-mls", or "gpq"); design (the route-independent model class, see reintervals-models);
- the per-row flags singular, boundary, vg_floored.

attr(., "diagnostics") is a list of engine-specific extras (degrees of freedom df; for the random-intercept confidence interval its source df_method — "satterthwaite", "containment", or "kenward-roger"; for the random-slope tolerance interval the GPQ route_reason; standard error se; and any GPQ / ANOVA intermediates); attr(., "grouping") is the random-intercept topology ("oneway", "nested", "crossed", or NA for slope models).

Known limitation

On a fixed-slope model with single-observation lots (the release-only stability structure), the EMS confidence and prediction intervals can be anti-conservative (too narrow), because REML precision-weighting down-weights the single-observation lots. `ci_lmm()` and `pi_lmm()` emit a warning when at least one lot has a single observation (the interval is still returned, unchanged); staggered designs whose lots all have two or more observations do not warn. The tolerance interval (`ti_lmm()`) and the new-group-mean interval (`new_group_mean_lmm()`) use the unweighted lot-mean (Montes) closed forms on single-observation and staggered fixed-slope designs and do not warn; no validated alternative is currently available for the confidence or prediction interval.

See Also

`pi_lmm()`, `ti_lmm()`, `new_group_mean_lmm()`, `lmm_predict()`, reintervals-models.

Examples

```
fm <- lme4::lmer(Reaction ~ Days + (1 | Subject), data = lme4::sleepstudy)
ci_lmm(fm, newdata = data.frame(Days = 0))
```

---

hk_assay *Hoffman & Kringle (2007) Table III assay data*

Description

The one-way random-effects worked example of Hoffman & Kringle (2007), Table III (their Example 1): an assay concentration measured in 6 analytical runs with 3 replicates each. It is the published oracle for the Hoffman-Kringle (2005) two-sided tolerance interval: with content `P = 0.95` and confidence `0.90`, `ti_lmm()` (default `"anova-mls"`) or `ti_anova_raw()` with `time = NULL` reproduces the published `(0.846, 1.114)`; with `P = 0.667` it reproduces `(0.914, 1.046)`.

Usage

```
hk_assay
```

Format

A data frame with 18 rows and 3 columns:

run analytical run (factor, 6 levels).

replicate replicate number within run (integer, 1-3).

conc measured concentration.

Details

The same data appear in Hoffman & Kringle (2005), Table 6, with run 5 / replicate 3 misprinted as `1.020`. The correct value is `0.998`: only it is consistent with the published mean `0.9798` and mean squares (between `0.003251`, within `0.001167`). Francq, Lin & Hoyer (2019), Sections 5.2-5.3, used the misprinted value, which is why their intervals for this example differ slightly from Hoffman & Kringle's.

Source

Hoffman D, Kringle R (2007). A total error approach for the validation of quantitative analytical methods. *Pharmaceutical Research* 24(6):1157-1164. doi:10.1007/s1109500792423. Table III.

References

Hoffman D, Kringle R (2005). Two-sided tolerance intervals for balanced and unbalanced random effects models. *Journal of Biopharmaceutical Statistics* 15(2):283-293. doi:10.1081/BIP-200048826

Francq BG, Lin D, Hoyer W (2019). Confidence, prediction, and tolerance in linear mixed models. *Statistics in Medicine* 38(30):5603-5622. doi:10.1002/sim.8386

See Also

`ti_lmm()`, `ti_anova_raw()`, reintervals-models.

Examples

```
fm <- lme4::lmer(conc ~ 1 + (1 | run), data = hk_assay)
ti_lmm(fm, P = 0.95, level = 0.90)   # Hoffman-Kringle: (0.846, 1.114)
```

`lmm_interval_control` *Tuning controls for the interval engines*

Description

Tuning knobs only — the RNG seed and Monte-Carlo size used when the GPQ algorithm runs, and the denominator-df rule of the random-intercept confidence interval. Algorithm *selection* is the `method` argument, not this.

Usage

```
lmm_interval_control(
  seed = NULL,
  M = 10000L,
  ci_df = c("satterthwaite", "kenward-roger")
)
```

Arguments

`seed`	Optional integer seed for the GPQ draws (reproducibility / GxP traceability). `NULL` uses the ambient RNG state.
`M`	Number of GPQ Monte-Carlo realizations. Default 10000.

ci_df — Denominator-df rule for the confidence interval (ci_lmm()) on random-intercept models: "satterthwaite" (default; Satterthwaite with the observed-information covariance of the variance components, falling back to the SAS-style containment df — e.g. A - 1 for a one-way or nested design, min(A, B) - 1 for a crossed one — when it is unavailable, as on a singular fit) or "kenward-roger" (requires **pbkrtest** and errors if it is not installed; never a silent fallback). The df source is reported in attr(., "diagnostics")$df_method. Random-slope models use their own generalized Satterthwaite df and reject "kenward-roger".

Value

An lmm_interval_control list (passed to a verb's control argument); a list with elements seed, M, and ci_df.

See Also

ci_lmm(), ti_lmm(), ti_gpq_raw().

Examples

```
lmm_interval_control(seed = 1, M = 5000)
lmm_interval_control(ci_df = "kenward-roger")
```

---

lmm_predict *Unified interface to the four interval verbs*

Description

A predict-style wrapper over the four interval verbs (ci_lmm(), pi_lmm(), ti_lmm(), new_group_mean_lmm()), selected by interval. This is a plain function, not an S3 method — it does not override lme4's predict.merMod.

Usage

```
lmm_predict(
  model,
  interval = c("confidence", "prediction", "tolerance", "new_group_mean"),
  newdata = NULL,
  P = 0.95,
  over = c("observation", "group_mean"),
  alpha = NULL,
  level = NULL,
  sides = c("two.sided", "lower", "upper"),
  method = "auto",
  control = lmm_interval_control()
)
```

Arguments

model	A fitted lmerMod.
interval	Which interval: "confidence", "prediction", "tolerance", or "new_group_mean".
newdata	Row(s) giving the fixed-effect combination(s); the time column supplies the evaluation point(s). NULL uses the first model-frame row.
P, over	Content proportion and tolerance target; honored only for interval = "tolerance" (forwarded to ti_lmm()). Supplying a non-default P/over with any other interval is an error.
alpha, level	The confidence, as a mutually-exclusive pair: give at most one. level is 1 - alpha; default alpha = 0.05.
sides	"two.sided" (default), "lower", or "upper". One-sided tolerance limits: anova-mls uses Hoffman's (2010) one-sided limit; ems-mls and random-slope models (reml-mls, the default on every fit) use the package's extension of it, validated by simulation. See reintervals-models.
method	"auto" (default) chooses the algorithm by structure; one of "ems-mls", "reml-mls", "anova-mls", "gpq" forces it and errors if incompatible with the model. "anova-mls" is available for fixed-slope models and for intercept-only one-way models (Hoffman-Kringle).
control	lmm_interval_control() tuning (GPQ seed / M).

Value

An lmm_interval object (a data frame), one row per evaluation point, with columns, in order:

- the eval-point column(s), named by the actual model predictor(s) (e.g. the time covariate for a slope model, or each newdata column);
- type ("ci", "pi", "ti", or "new_group_mean"), estimate, lower, upper;
- level (the confidence, 1 - alpha); P (content proportion — the value for ti, NA for the coverage intervals); sides;
- method (the engine: "ems-mls", "reml-mls", "anova-mls", or "gpq"); design (the route-independent model class, see reintervals-models);
- the per-row flags singular, boundary, vg_floored.

attr(., "diagnostics") is a list of engine-specific extras (degrees of freedom df; for the random-intercept confidence interval its source df_method — "satterthwaite", "containment", or "kenward-roger"; for the random-slope tolerance interval the GPQ route_reason; standard error se; and any GPQ / ANOVA intermediates); attr(., "grouping") is the random-intercept topology ("oneway", "nested", "crossed", or NA for slope models).

See Also

ci_lmm(), pi_lmm(), ti_lmm(), new_group_mean_lmm(), reintervals-models.

Examples

```
fm <- lme4::lmer(Reaction ~ Days + (1 | Subject), data = lme4::sleepstudy)
lmm_predict(fm, interval = "confidence", newdata = data.frame(Days = 0))
```

`mbl_appendix_a` *Montes-Burdick-Leblond (2019) Appendix A stability data*

Description

The worked-example stability dataset from Montes, Burdick & Leblond (2019), Appendix A: 11 lots measured over time, mixing genuine multi-timepoint stability lots (A-D) with release-only single-observation lots (E-K). It is the published oracle for the random-intercept + fixed-slope tolerance interval: a two-sided P = 0.99 / gamma = 0.95 interval at `Month = 12` reproduces the published `(57.25, 75.41)` (via `ti_anova_raw()` or `ti_lmm()`); intermediates `beta = 0.457`, `n_E = 6.729`, `U = 10.820`.

Usage

```
mbl_appendix_a
```

Format

A data frame with 26 rows and 3 columns:

Lot lot identifier (character), 11 lots `A-K`.

Month time since release (numeric, months).

Y measured response.

Source

Montes RO, Burdick RK, Leblond DJ (2019). Simple approach to calculate random effects model tolerance intervals to set release and shelf-life specification limits of pharmaceutical products. *PDA Journal of Pharmaceutical Science and Technology*. doi:10.5731/pdajpst.2018.008839. Appendix A (reproduced from their Table II).

See Also

`ti_anova_raw()`, `ti_lmm()`, reintervals-models.

Examples

```
fm <- lme4::lmer(Y ~ Month + (1 | Lot), data = mbl_appendix_a)
ti_lmm(fm, newdata = data.frame(Month = 12), P = 0.99)
```

new_group_mean_lmm *New-group-mean (CInew / new-batch-mean) interval*

Description

One of the four interval verbs: an interval for the mean of one new, as-yet-unobserved group/batch (the CInew / new-batch-mean interval). Its coverage is `level`/`alpha`, so it carries no content `P` (unlike `ti_lmm()`).

Usage

```
new_group_mean_lmm(
  model,
  newdata = NULL,
  alpha = NULL,
  level = NULL,
  sides = c("two.sided", "lower", "upper"),
  method = "auto",
  control = lmm_interval_control()
)
```

Arguments

`model`	A fitted `lmerMod` (from `lme4::lmer()`).
`newdata`	Row(s) giving the fixed-effect combination(s) to evaluate at. `NULL` uses the first model-frame row.
`alpha`, `level`	The interval confidence, as a mutually-exclusive pair: give at most one. `level` is `1 - alpha`; the default is `alpha = 0.05` (95%).
`sides`	`"two.sided"` (default), `"lower"`, or `"upper"`. A one-sided interval keeps the two-sided interval's center, variance, and df and uses `t_{1 - alpha}` in place of `t_{1 - alpha/2}`.
`method`	`"auto"` (default) chooses the algorithm by structure; one of `"ems-mls"`, `"reml-mls"`, `"anova-mls"`, `"gpq"` forces that engine and errors if it is incompatible with the model.
`control`	An `lmm_interval_control()` object (the GPQ `seed` and `M`, and the confidence-interval df rule `ci_df`).

Value

An `lmm_interval` object (a data frame), one row per evaluation point, with columns, in order:

- the eval-point column(s), named by the actual model predictor(s) (e.g. the time covariate for a slope model, or each `newdata` column);
- `type` (`"ci"`, `"pi"`, `"ti"`, or `"new_group_mean"`), `estimate`, `lower`, `upper`;
- `level` (the confidence, `1 - alpha`); `P` (content proportion — the value for `ti`, `NA` for the coverage intervals); `sides`;

- method (the engine: "ems-mls", "reml-mls", "anova-mls", or "gpq"); design (the route-independent model class, see reintervals-models);
- the per-row flags singular, boundary, vg_floored.

attr(., "diagnostics") is a list of engine-specific extras (degrees of freedom df; for the random-intercept confidence interval its source df_method — "satterthwaite", "containment", or "kenward-roger"; for the random-slope tolerance interval the GPQ route_reason; standard error se; and any GPQ / ANOVA intermediates); attr(., "grouping") is the random-intercept topology ("oneway", "nested", "crossed", or NA for slope models).

Target

The target is the **true** mean of a new innermost cell: a new group in a one-way design, a new B-within-A cell in a nested design, a new (a, b) cell in a crossed design; for the fixed-slope and random-slope designs, the true mean of a new lot at t0.

Constructions

The interval is center +/- t_df * sqrt(W), with W, df and the rule reported in attr(., "diagnostics") (W, df, df_rule):

- fixed slope (one grouping factor, any balance): the unweighted lot-mean (Montes) center Yhat(t0) with W = S2_L (1 + 1/I) + S2_E (t0 - tbar)^2 / S_tt and a Satterthwaite df on the two mean squares (df_rule = "anova-satterthwaite", method "anova-mls");
- intercept-only one-way: the unweighted mean of group means with W = S2_L (1 + 1/I) and df = I - 1 ("anova-single", "anova-mls");
- nested, and one-way with other fixed effects: W = V_F + sum_j k_j EMS_j over the random terms (REML-implied expected mean squares, synthesis coefficients when unbalanced) with the top factor's df A - 1 ("top-factor", "ems-mls");
- crossed with interaction: the same W, with a Satterthwaite df in which V_F is attached to the two main-effect terms ("satterthwaite-vf", "ems-mls");
- random slope: the REML closed form $\mu(t_0) \pm t_{\nu_G} \sqrt{V_F + V_G}$ ("reml-mls").

A design without a closed-form variance decomposition (two crossed factors without their interaction term, or four or more grouping factors) is an error. One-sided intervals (sides = "lower" / "upper") use the same center, W, and df with t_{1 - alpha}.

See Also

ci_lmm(), pi_lmm(), ti_lmm(), lmm_predict(), reintervals-models.

Examples

```
fm <- lme4::lmer(Reaction ~ Days + (1 + Days | Subject),
                 data = lme4::sleepstudy)
new_group_mean_lmm(fm, newdata = data.frame(Days = 5))
```

`oliva_hauser_dgp` *Oliva-Aviles & Hauser (2025) Section 5 DGP specification (GPQ reference)*

Description

The published data-generating specification for the random-intercept + random-slope single-temperature stability model of Oliva-Aviles & Hauser (2025), Section 5, at variance ratio `kappa = 1`. The paper reports a simulation / coverage study rather than a single raw dataset, so the specification (not a dataset) is stored: it makes the closed-form content variance exactly reproducible and lets reference datasets be simulated deterministically. The GPQ tolerance interval targets the content variance `tau^2(t0) = z' V z`, `z = (1, t0)`; at `t0 = 20` this is `8.439388` (the paper's `8.439`), reproduced exactly by the engine's `V_G` machinery.

Usage

```
oliva_hauser_dgp
```

Format

A named list:

beta0, beta1 fixed intercept and slope (100, -0.03).

s2_0, s01, s2_1 random intercept variance, intercept-slope covariance, slope variance of `V` (at `kappa = 1`).

s2e residual variance (1).

kappa the variance ratio (1).

times the staggered measurement times.

t0 the evaluation time (20).

tau2_at_t0 the closed-form content variance at `t0` (8.439388).

citation the source reference.

Source

Oliva-Aviles C, Hauser P (2025). *Technometrics* 67(2):193-202 doi:10.1080/00401706.2024.2407324, Section 5.

See Also

`ti_gpq_raw()`, reintervals-models.

Examples

```
# the published content-variance functional, reproduced exactly:
oliva_hauser_dgp$tau2_at_t0
```

pi_lmm *Prediction interval for a future observation of a linear mixed model*

Description

One of the four interval verbs (ci_lmm(), pi_lmm(), ti_lmm(), new_group_mean_lmm()). pi_lmm() gives a prediction interval for one future observation at newdata. Dispatch and arguments are as for ci_lmm().

Usage

```
pi_lmm(
  model,
  newdata = NULL,
  alpha = NULL,
  level = NULL,
  sides = c("two.sided", "lower", "upper"),
  method = "auto",
  control = lmm_interval_control()
)
```

Arguments

model	A fitted lmerMod (from lme4::lmer()).
newdata	Row(s) giving the fixed-effect combination(s) to evaluate at. NULL uses the first model-frame row.
alpha, level	The interval confidence, as a mutually-exclusive pair: give at most one. level is 1 - alpha; the default is alpha = 0.05 (95%).
sides	"two.sided" (default), "lower", or "upper". A one-sided interval keeps the two-sided interval's center, variance, and df and uses t_{1 - alpha} in place of t_{1 - alpha/2}.
method	"auto" (default) chooses the algorithm by structure; one of "ems-mls", "reml-mls", "anova-mls", "gpq" forces that engine and errors if it is incompatible with the model.
control	An lmm_interval_control() object (the GPQ seed and M, and the confidence-interval df rule ci_df).

Value

An lmm_interval object (a data frame), one row per evaluation point, with columns, in order:

- the eval-point column(s), named by the actual model predictor(s) (e.g. the time covariate for a slope model, or each newdata column);
- type ("ci", "pi", "ti", or "new_group_mean"), estimate, lower, upper;
- level (the confidence, 1 - alpha); P (content proportion — the value for ti, NA for the coverage intervals); sides;

- method (the engine: "ems-mls", "reml-mls", "anova-mls", or "gpq"); design (the route-independent model class, see reintervals-models);
- the per-row flags singular, boundary, vg_floored.

attr(., "diagnostics") is a list of engine-specific extras (degrees of freedom df; for the random-intercept confidence interval its source df_method — "satterthwaite", "containment", or "kenward-roger"; for the random-slope tolerance interval the GPQ route_reason; standard error se; and any GPQ / ANOVA intermediates); attr(., "grouping") is the random-intercept topology ("oneway", "nested", "crossed", or NA for slope models).

Known limitation (few top-level groups)

With few levels of the top random factor (five or fewer) and a dominant between-group variance (an intraclass correlation near 0.9), the Francq et al. (2019) prediction interval used for random-intercept models can under-cover: the worst coverage observed in the package's validation was about 0.93 for a nominal 0.95, consistent with Francq et al. (2019, Fig. 2).

Known limitation

On a fixed-slope model with single-observation lots (the release-only stability structure), the EMS confidence and prediction intervals can be anti-conservative (too narrow), because REML precision-weighting down-weights the single-observation lots. ci_lmm() and pi_lmm() emit a warning when at least one lot has a single observation (the interval is still returned, unchanged); staggered designs whose lots all have two or more observations do not warn. The tolerance interval (ti_lmm()) and the new-group-mean interval (new_group_mean_lmm()) use the unweighted lot-mean (Montes) closed forms on single-observation and staggered fixed-slope designs and do not warn; no validated alternative is currently available for the confidence or prediction interval.

See Also

ci_lmm(), ti_lmm(), new_group_mean_lmm(), lmm_predict(), reintervals-models.

Examples

```
fm <- lme4::lmer(Reaction ~ Days + (1 | Subject), data = lme4::sleepstudy)
pi_lmm(fm, newdata = data.frame(Days = 0))
```

---

reintervals-models *Mixed-model classes reintervals computes intervals for*

Description

reintervals covers four linear-mixed-model classes: random intercepts (one-way, nested, or crossed; general fixed effects), a random intercept with a fixed slope, and a random intercept with a correlated or independent random slope (single grouping factor, fixed part ~ 1 + time). The interval verbs (ci_lmm(), pi_lmm(), ti_lmm(), new_group_mean_lmm()) detect the class from the fitted model and route to the engine that is exact/validated for it (engine map: reintervals, vignette("choosing-an-interval")); the class is reported back, route-independently, in the design column of the result. This page records the mapping from each model definition to its design tag.

Model classes

lme4 formula	`design` tag	description
`y ~ 1 + (1 | g)` (and nested/crossed)	`random_intercept`	random intercept only
`y ~ time + (1 | g)`	`random_intercept_fixed_slope`	random intercept with a fixed (population) slope
`y ~ time + (1 + time | g)`	`random_slope_correlated`	random intercept and random slope, correlated
`y ~ time + (1 | g) + (0 + time | g)`	`random_slope_independent`	random intercept and random slope, independent

Random-intercept topology (the `grouping` attribute)

For random-intercept models (no random slope) the one-way / nested / crossed topology is reported separately in `attr(result, "grouping")` (`"oneway"`, `"nested"`, or `"crossed"`); it is `NA` for random-slope models.

Extrapolation in time (slope models)

For the random-slope and fixed-slope models the between-group variance `V_G(t0) = z0' Sigma z0` is quadratic in the time covariate. Evaluating an interval at a time point outside the observed range therefore extrapolates that quadratic; the package computes it without a guard or warning, so out-of-range `t0` should be requested knowingly.

new_group_mean (CInew / new-batch-mean)

`new_group_mean_lmm()` is the CInew / new-batch-mean interval — a prediction interval for the mean of one new, as-yet-unobserved group/batch (the prediction interval with the residual component removed). It is listed here under both names so it is discoverable as "CInew" or "new batch mean".

One-sided intervals

Every verb accepts `sides = "lower"` or `"upper"`.

- Confidence, prediction, and new-group-mean intervals: the same center, variance, and df as the two-sided interval, with `t_{1 - alpha}` in place of `t_{1 - alpha/2}` (all engines).
- Tolerance intervals, `anova-mls` (Hoffman-Kringle one-way and Montes fixed slope): Hoffman's (2010) one-sided limit `Yhat -/+ (z_P sqrt(U) + z_{1-alpha} sqrt(Vhat(Yhat)))`, with the same MLS bound `U` as the two-sided interval (Montes et al. 2019, Eqs. 2-3, 6-7).
- Tolerance intervals, `ems-mls` (Francq): the same structure with Francq's MLS bound on the total variance and `Vhat = V_F` — the package's extension of Hoffman (2010), validated by simulation.
- Tolerance intervals, random slope: Hoffman's structure with the REML closed form's bound and `Vhat = V_F` (`reml-mls`), on every fit including singular ones — the package's extension of Hoffman (2010), validated by simulation. `method = "gpq"` is available but under-covered near the variance-component boundary in validation.

References

Hoffman D (2010). One-sided tolerance limits for balanced and unbalanced random effects models. *Technometrics* 52(3):303-312. doi:10.1198/TECH.2010.09171

See Also

ci_lmm(), pi_lmm(), ti_lmm(), new_group_mean_lmm(), lmm_predict(), ti_anova_raw(), ti_gpq_raw().

`ti_anova_raw` *ANOVA/MLS tolerance interval from raw data (direct engine, advanced)*

Description

Tier-2 direct engine: the published closed-form (ANOVA mean-square / modified large-sample) tolerance interval from raw vectors, without a fitted `lmer` model. Two cases share one estimator (the *unweighted lot-mean* ANOVA):

Usage

```
ti_anova_raw(
  y,
  time,
  lot,
  t0 = NULL,
  P = 0.95,
  alpha = NULL,
  level = NULL,
  sides = c("two.sided", "lower", "upper")
)
```

Arguments

`y`	Numeric response.
`time`	Numeric time/age covariate (same length as `y`), or `NULL` for the one-way (Hoffman-Kringle) interval with no covariate.
`lot`	Lot/batch (grouping) factor (same length as `y`).
`t0`	Time point(s) at which to evaluate the interval; must be `NULL` (the default) when `time = NULL`.
`P`	Content proportion (default 0.95; pharma stability typically 0.99).
`alpha, level`	Confidence as a mutually-exclusive pair (give at most one); `level = 1 - alpha`, default `alpha = 0.05`.
`sides`	`"two.sided"` (default), `"lower"`, or `"upper"` (Hoffman's one-sided limit).

Details

- **With a time covariate** (`time`, `t0` given): the Montes, Burdick & Leblond (2019, Appendix A) interval for the random-intercept + fixed-slope stability model. `ti_lmm()` routes single-observation / staggered fixed-slope designs here automatically.
- **Without one** (`time = NULL`, `t0 = NULL`): the Hoffman & Kringle (2005) two-sided interval for the one-way random-effects model, balanced or unbalanced (restated in Hoffman & Kringle 2007, Eqs. 1-2). It is the same computation with the slope removed (residual df `N - I`, center = the unweighted mean of the group means). `ti_lmm()` uses it by default for an intercept-only model with one random-intercept factor.

One-sided limits (both cases) are Hoffman's (2010) limit `Yhat -/+ (z_P sqrt(U) + z_{1-alpha} sqrt(Vhat(Yhat)))`, with the same MLS bound `U` as the two-sided interval and `Vhat(Yhat)` the variance of the center (Montes et al. 2019, Eqs. 2-3 and 6-7).

Method-pure (never touches `lme4`). The between- and within-lot mean squares are sums of squares over their degrees of freedom and so are non-negative by construction; no ANOVA-estimate truncation.

Value

A data frame (one row per `t0`, or one row when `time = NULL`): `time` (only with a time covariate), `type = "TI"`, `estimate`, `lower`, `upper`, `content`, `conf`, `df`, `sides`, `design` (`"random_intercept_fixed_slope"`, or `"random_intercept"` when `time = NULL`), `method = "anova-mls"`, and `singular`/`boundary` flags. ANOVA intermediates are attached as attributes: `n_E`, `U`, `Vhat_Y` (Hoffman-Kringle's `N_e` and `sTOT2` when `time = NULL`), `C1`, `C2` (their `H1`, `H2`), and `beta_hat` (with a time covariate).

References

Hoffman D (2010). One-sided tolerance limits for balanced and unbalanced random effects models. *Technometrics* 52(3):303-312. doi:10.1198/TECH.2010.09171

Montes RO, Burdick RK, Leblond DJ (2019). Simple approach to calculate random effects model tolerance intervals to set release and shelf-life specification limits of pharmaceutical products. *PDA Journal of Pharmaceutical Science and Technology*. doi:10.5731/pdajpst.2018.008839

Hoffman D, Kringle R (2005). Two-sided tolerance intervals for balanced and unbalanced random effects models. *Journal of Biopharmaceutical Statistics* 15(2):283-293. doi:10.1081/BIP-200048826

Hoffman D, Kringle R (2007). A total error approach for the validation of quantitative analytical methods. *Pharmaceutical Research* 24(6):1157-1164. doi:10.1007/s1109500792423

See Also

`ti_lmm()` (the dispatcher), `ti_gpq_raw()` (the GPQ raw engine), hk_assay, reintervals-models.

Examples

```
lot <- rep(c("A", "B", "C", "D", "E", "F"), c(6, 5, 2, 5, 2, 1))
ti_anova_raw(
  y    = c(70, 71, 69, 68, 66, 65, 71, 69, 68, 66, 66, 71, 69,
           73, 73, 71, 71, 70, 72, 73, 70),
```

```
  time = c(0, 1, 3, 6, 9, 12, 0, 1, 3, 6, 9, 0, 3, 0, 1, 3, 6, 9, 0, 1, 0),
  lot  = lot, t0 = 12, P = 0.99
)
# one-way, no covariate: Hoffman & Kringle (2005), here (0.846, 1.114)
ti_anova_raw(hk_assay$conc, time = NULL, lot = hk_assay$run,
             P = 0.95, level = 0.90)
```

`ti_gpq_raw` *GPQ tolerance interval from raw data (direct engine, advanced)*

Description

The Oliva-Aviles & Hauser (2025) generalized-pivotal- quantity tolerance interval for the random-intercept + random-slope stability model, from raw `(y, time, batch)` vectors without a fitted `lmer` model. The fit-based dispatcher `ti_lmm()` routes random-slope designs here automatically (singular fits, fits whose closed-form Satterthwaite df is unavailable at a given `t0`, and the batch-mean target `over = "group_mean"`); use `ti_gpq_raw()` directly when you have data but no model object.

Usage

```
ti_gpq_raw(
  y,
  time,
  batch,
  t0,
  P = 0.95,
  alpha = NULL,
  level = NULL,
  sides = c("two.sided", "lower", "upper"),
  c_resid = 1,
  control = lmm_interval_control()
)
```

Arguments

`y`	Numeric response.
`time`	Numeric time/age covariate (same length as `y`).
`batch`	Batch factor (same length as `y`).
`t0`	Time point(s) at which to evaluate the interval.
`P`	Content proportion (default 0.95; pharma stability typically 0.99).
`alpha, level`	Confidence as a mutually-exclusive pair (give at most one); `level = 1 - alpha`, default `alpha = 0.05`.
`sides`	`"two.sided"` (default), `"lower"`, or `"upper"`.
`c_resid`	Residual multiplier `c` in `N(theta, tau2 + c*s2e)`: `1` (default) a future observation, `0` the batch-mean.
`control`	`lmm_interval_control()` tuning (the GPQ `seed` and `M`).

Value

A data frame, one row per `t0`: `method = "gpq"`, `time`, `estimate`, `lower`, `upper`, `design = "random_slope_correlated"`, plus audit fields (`tau2_med`, `s2e_med`, `frac_boundary`, `K`, `df_resid`, `phi1`).

References

Oliva-Aviles C, Hauser P (2025). *Technometrics* 67(2):193-202. doi:10.1080/00401706.2024.2407324

See Also

`ti_lmm()` (the dispatcher), `ti_anova_raw()` (the ANOVA raw engine), reintervals-models.

Examples

```
fm <- lme4::lmer(Reaction ~ Days + (1 + Days | Subject), data = lme4::sleepstudy)
d <- lme4::sleepstudy
ti_gpq_raw(d$Reaction, d$Days, d$Subject, t0 = 5,
           control = lmm_interval_control(seed = 1, M = 2000))
```

`ti_lmm` *Tolerance interval for a linear mixed model*

Description

One of the four interval verbs, and the dispatched tolerance interval. It classifies the fitted model and routes by random/fixed structure, the tolerance target (`over`), the singular flag, and (random slope) whether the closed form's Satterthwaite df is available, to the engine that is exact/validated for that case — all transparently:

Usage

```
ti_lmm(
  model,
  newdata = NULL,
  P = 0.95,
  over = c("observation", "group_mean"),
  alpha = NULL,
  level = NULL,
  sides = c("two.sided", "lower", "upper"),
  method = "auto",
  control = lmm_interval_control()
)
```

Arguments

model	A fitted lmerMod.
newdata	Row(s) giving the fixed-effect combination(s); the time column supplies the evaluation point(s). NULL uses the first model-frame row.
P	Content proportion (the population fraction covered). Default 0.95 (pharma stability work typically passes P = 0.99).
over	What the interval covers: "observation" (default; a proportion P of future observations — the ordinary tolerance interval) or "group_mean" (a proportion P of new group means — the batch-mean tolerance interval).
alpha, level	The confidence, as a mutually-exclusive pair: give at most one. level is 1 - alpha; default alpha = 0.05.
sides	"two.sided" (default), "lower", or "upper". One-sided tolerance limits: anova-mls uses Hoffman's (2010) one-sided limit; ems-mls and random-slope models (reml-mls, the default on every fit) use the package's extension of it, validated by simulation. See reintervals-models.
method	"auto" (default) chooses the algorithm by structure; one of "ems-mls", "reml-mls", "anova-mls", "gpq" forces it and errors if incompatible with the model. "anova-mls" is available for fixed-slope models and for intercept-only one-way models (Hoffman-Kringle).
control	lmm_interval_control() tuning (GPQ seed / M).

Details

- **Random slope.** over = "observation" uses the REML closed-form MLS interval when the fit is non-singular and the GPQ interval when the fit is singular or, at a given t0, when the Satterthwaite df of the closed form's between-group or residual variance is unavailable (its containment fallback fired); over = "group_mean" always uses GPQ. One-sided intervals (sides = "lower" / "upper", future-observation target) use Hoffman's (2010) one-sided limit on the REML closed form on every fit, singular and df-unavailable fits included (validated by simulation on all of them); method = "gpq" remains available, but in the package's validation the one-sided GPQ limit under-covered on near-boundary singular fits at the edge of the observed time range (worst cell 0.928 for a nominal 0.95). The reason is reported per row in attr(., "diagnostics")$route_reason ("none", "singular", "df_unavailable", "group_mean_target", or "method" when forced).
- **Random intercept + fixed slope.** Clean balanced -> EMS engine; single- observation / staggered / unbalanced -> the Montes ANOVA/MLS closed form.
- **Intercept-only one-way** (y ~ 1 + (1 | g), one grouping factor, no covariates), balanced or not -> the Hoffman & Kringle (2005) closed form ("anova-mls"). method = "ems-mls" gives the Francq et al. (2019) interval instead; the two coincide on balanced data with a non-singular fit and differ on singular or unbalanced data.
- **Any other random-intercept model, no fixed slope** (one-way with covariates, nested, crossed) -> EMS engine.

For slope models the between-group variance V_G(t0) is quadratic in the time covariate, so evaluating at a t0 outside the observed time range is an extrapolation of that quadratic and should be done knowingly (the interval is computed without a guard or warning). The same applies to the other slope verbs (ci_lmm(), pi_lmm(), new_group_mean_lmm()).

Value

An `lmm_interval` object (a data frame), one row per evaluation point, with columns, in order:

- the eval-point column(s), named by the actual model predictor(s) (e.g. the time covariate for a slope model, or each `newdata` column);
- `type` (`"ci"`, `"pi"`, `"ti"`, or `"new_group_mean"`), `estimate`, `lower`, `upper`;
- `level` (the confidence, `1 - alpha`); `P` (content proportion — the value for `ti`, `NA` for the coverage intervals); `sides`;
- `method` (the engine: `"ems-mls"`, `"reml-mls"`, `"anova-mls"`, or `"gpq"`); `design` (the route-independent model class, see reintervals-models);
- the per-row flags `singular`, `boundary`, `vg_floored`.

`attr(., "diagnostics")` is a list of engine-specific extras (degrees of freedom `df`; for the random-intercept confidence interval its source `df_method` — `"satterthwaite"`, `"containment"`, or `"kenward-roger"`; for the random-slope tolerance interval the GPQ `route_reason`; standard error `se`; and any GPQ / ANOVA intermediates); `attr(., "grouping")` is the random-intercept topology (`"oneway"`, `"nested"`, `"crossed"`, or `NA` for slope models).

Availability

Tolerance intervals require a closed-form variance decomposition. They are available for one-way, nested, and crossed-with-interaction random-intercept designs, the fixed-slope stability design, and random-slope models. For a design without such a decomposition (two crossed factors with no interaction term, or four or more grouping factors), `ti_lmm()` and `new_group_mean_lmm()` signal an error; `ci_lmm()` and `pi_lmm()` remain available for those models.

References

Hoffman D (2010). One-sided tolerance limits for balanced and unbalanced random effects models. *Technometrics* 52(3):303-312. doi:10.1198/TECH.2010.09171

Hoffman D, Kringle R (2005). Two-sided tolerance intervals for balanced and unbalanced random effects models. *Journal of Biopharmaceutical Statistics* 15(2):283-293. doi:10.1081/BIP-200048826

See Also

`ci_lmm()`, `pi_lmm()`, `new_group_mean_lmm()`, `lmm_predict()`, reintervals-models.

Examples

```
# intercept-only one-way -> Hoffman-Kringle (anova-mls)
fh <- lme4::lmer(conc ~ 1 + (1 | run), data = hk_assay)
ti_lmm(fh, P = 0.95, level = 0.90)
# random intercept + fixed slope (clean balanced) -> EMS engine
fi <- lme4::lmer(Reaction ~ Days + (1 | Subject), data = lme4::sleepstudy)
ti_lmm(fi, newdata = data.frame(Days = 0))
# random slope, future-observation -> REML closed form
fs <- lme4::lmer(Reaction ~ Days + (1 + Days | Subject), data = lme4::sleepstudy)
ti_lmm(fs, newdata = data.frame(Days = 5))
```

```
# random slope, batch-mean -> GPQ
ti_lmm(fs, newdata = data.frame(Days = 5), over = "group_mean",
       control = lmm_interval_control(seed = 1))
```

Index

∗ **datasets**
hk_assay, 4
mbl_appendix_a, 8
oliva_hauser_dgp, 11

ci_lmm, 2
ci_lmm(), *2*, *4*, *6, 7*, *10*, *12, 13*, *15*, *19, 20*

hk_assay, 4, *16*

lme4::lmer(), *3*, *9*, *12*
lmm_interval_control, 5
lmm_interval_control(), *3*, *7*, *9*, *12*, *17*, *19*
lmm_predict, 6
lmm_predict(), *2*, *4*, *10*, *13*, *15*, *20*

mbl_appendix_a, 8

new_group_mean_lmm, 9
new_group_mean_lmm(), *2*, *4*, *6, 7*, *12–15*, *19, 20*

oliva_hauser_dgp, 11

pi_lmm, 12
pi_lmm(), *2*, *4*, *6, 7*, *10*, *12, 13*, *15*, *19, 20*

reintervals, *2*, *13*
reintervals-models, *2–5*, *7, 8*, *10, 11*, *13*, 13, *16*, *18–20*

ti_anova_raw, 15
ti_anova_raw(), *4, 5*, *8*, *15*, *18*
ti_gpq_raw, 17
ti_gpq_raw(), *6*, *11*, *15, 16*
ti_lmm, 18
ti_lmm(), *2*, *4–10*, *12, 13*, *15–18*, *20*