Package 'pecanr'

October 20, 2026

Title Partial Eta-Squared for Crossed, Nested, and Mixed Linear Mixed Models

Version 0.4.0

Description Computes partial eta-squared effect sizes for fixed effects in linear mixed models fitted with the 'lme4' package. Supports crossed, nested, and mixed (crossed-and-nested) random effects structures with any number of grouping factors. Mixed designs handle cases where grouping factors are simultaneously crossed with some variables and nested within others (e.g., photos nested within models, but both crossed with participants). Factor predictors are supported directly. Random slope variances are translated to the outcome scale using a variance decomposition approach, correctly accounting for predictor scaling and interaction terms. Both general and operative effect sizes are provided, with optional parametric bootstrap confidence intervals. For correlated predictors, per-predictor effect sizes use unique (semipartial) variance by default. Methods are based on Correll, Mellinger, McClelland, and Judd (2020) <doi:10.1016/j.tics.2019.12.009>, Correll, Mellinger, and Pedersen (2022) <doi:10.3758/s13428-021-01687-2>, and Rights and Sterba (2019) <doi:10.1037/met0000184>.

License MIT + file LICENSE

URL https://github.com/bcohen0901/pecanr

BugReports https://github.com/bcohen0901/pecanr/issues

Encoding UTF-8

Imports lme4, stats

Suggests spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Language en-US

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Brandon Cohen [aut, cre] (ORCID: <https://orcid.org/0009-0004-1802-3435>), Joshua Correll [aut, ths] (ORCID: <https://orcid.org/0000-0003-1142-459X>)

Maintainer Brandon Cohen <brandon.cohen-1@colorado.edu>

Repository CRAN

Date/Publication 2026-10-20 19:30:02 UTC

Contents

batch_eta2p 2
eta2p 3

Index **12**

`batch_eta2p` *Batch Calculate Partial Eta-Squared for Multiple Effects*

Description

Calculates partial eta-squared for all fixed effects (excluding the intercept) in a model, returning one row per model coefficient. For a multi-level factor or multi-df interaction this yields one value *per contrast* (per design-matrix column), not a single factor-level value.

Usage

```
batch_eta2p(
  model,
  data,
  design = c("crossed", "nested", "mixed"),
  subj_var = NULL,
  item_var = NULL,
  cross_vars = NULL,
  nest_vars = NULL,
  operative = FALSE,
  partial_predictors = TRUE,
  verbose = FALSE
)
```

Arguments

`model`	A fitted model object from `lme4::lmer()`.
`data`	Data frame used to fit the model.
`design`	Character string: `"crossed"`, `"nested"`, or `"mixed"`. Use `"mixed"` when the model contains both crossed and nested random effects (e.g., photos nested within models, but both crossed with participants).
`subj_var`	Character string specifying the subject/participant grouping variable. Retained for backward compatibility; prefer `cross_vars`.
`item_var`	Character string specifying the item/stimulus grouping variable. Retained for backward compatibility; prefer `cross_vars`.

`cross_vars` Character vector of grouping variable names that are *crossed* with each other and with the nested hierarchy (e.g., `c("participant")` or `c("participant", "rater")`). Supersedes `subj_var` and `item_var` when provided.

`nest_vars` Character vector of nesting variables ordered from lowest to highest level (e.g., `c("photo", "model")` when photos are nested within models). Required for `design = "nested"` or `design = "mixed"`.

`operative` Logical. If `TRUE`, calculates operative effect size (excludes variance components that do not contribute to the SE of the effect). Default is `FALSE`.

`partial_predictors`
Logical. If `TRUE` (the default), the variance attributed to a single (non-interaction) predictor is its *unique* (semipartial) variance – the variance remaining after removing the part linearly predictable from the other fixed-effect predictors, equivalently $\mathrm{Var}(X) \times \mathrm{tol}(X)$ where tol is the predictor's tolerance. This yields a measure that reflects only the unique contribution of the predictor and declines as its redundancy with the others increases. If `FALSE`, the predictor's *total* variance is used instead (the raw decomposition). For centered, orthogonal designs the two are identical; they differ only when predictors are correlated. See Details. Has no effect on interaction terms (whose components are always centered) or when `var_x` is supplied.

`verbose` Logical. If `TRUE`, prints detailed results. Default is `FALSE`.

Value

A data frame with one row per effect containing:

`effect` Name of the fixed effect.

`eta2p` Partial eta-squared value.

`effect_level` Effect level (nested/mixed designs only).

`variance_effect`
Variance attributed to the effect.

`variance_error` Error variance used in the denominator.

`type` `"general"` or `"operative"`.

`within_subj`, `within_item`
(Crossed/mixed, operative only) Whether the effect is within subjects / within items.

Rows are sorted by `eta2p` in decreasing order.

`eta2p` *Partial Eta-Squared for Linear Mixed Models*

Description

Calculates partial eta-squared effect sizes for fixed effects in linear mixed models with crossed, nested, or mixed (crossed-and-nested) random effects.

Usage

```
eta2p(
  model,
  effect,
  data,
  design = c("crossed", "nested", "mixed"),
  subj_var = NULL,
  item_var = NULL,
  cross_vars = NULL,
  nest_vars = NULL,
  effect_level = NULL,
  var_x = NULL,
  slope = NULL,
  operative = FALSE,
  partial_predictors = TRUE,
  ci = FALSE,
  ci_level = 0.95,
  n_boot = 1000,
  seed = NULL,
  verbose = FALSE
)

## S3 method for class 'eta2p_lmm'
print(x, ...)

## S3 method for class 'eta2p_lmm'
summary(object, ...)
```

Arguments

model	A fitted model object from `lme4::lmer()`.
effect	Character string specifying the fixed effect to analyze. May be given either as the model coefficient name (e.g. `"Group75yr"`) or as the underlying variable name (e.g. `"Group"`); a variable name that maps to a single coefficient is resolved automatically (with a message). A variable that maps to several coefficients (a multi-level factor) raises an informative error pointing to `batch_eta2p()` (for per-coefficient values). Factor predictors and factor random slopes are read from the model design matrix / model frame, so no manual recoding to numeric is required.
data	Data frame used to fit the model.
design	Character string: `"crossed"`, `"nested"`, or `"mixed"`. Use `"mixed"` when the model contains both crossed and nested random effects (e.g., photos nested within models, but both crossed with participants).
subj_var	Character string specifying the subject/participant grouping variable. Retained for backward compatibility; prefer `cross_vars`.
item_var	Character string specifying the item/stimulus grouping variable. Retained for backward compatibility; prefer `cross_vars`.

`cross_vars` Character vector of grouping variable names that are *crossed* with each other and with the nested hierarchy (e.g., `c("participant")` or `c("participant", "rater")`). Supersedes `subj_var` and `item_var` when provided.

`nest_vars` Character vector of nesting variables ordered from lowest to highest level (e.g., `c("photo", "model")` when photos are nested within models). Required for `design = "nested"` or `design = "mixed"`.

`effect_level` Character string specifying the level at which the effect varies (e.g., `"L1"`, `"L2"`). If `NULL`, detected automatically from `nest_vars`.

`var_x` Optional numeric. Pre-computed variance of the predictor (or interaction product). If supplied, overrides the internal `var()` calculation from `data`. Useful when raw data are unavailable but the predictor variance is known (e.g., from design: a +/-1 binary predictor has `var_x = 1`).

`slope` Optional numeric. A hypothetical value for the focal effect's fixed-effect coefficient, used in place of the fitted coefficient in the numerator while the predictor variance and the error structure are held fixed from the model. This answers "what effect size would a slope of this size imply?" and is useful for sensitivity and power analyses (e.g. reporting effect sizes alongside **simr**). The effect must be present in the fitted model. Default `NULL` uses the fitted coefficient.

`operative` Logical. If `TRUE`, calculates operative effect size (excludes variance components that do not contribute to the SE of the effect). Default is `FALSE`.

`partial_predictors`
Logical. If `TRUE` (the default), the variance attributed to a single (non-interaction) predictor is its *unique* (semipartial) variance – the variance remaining after removing the part linearly predictable from the other fixed-effect predictors, equivalently $\mathrm{Var}(X) \times \mathrm{tol}(X)$ where tol is the predictor's tolerance. This yields a measure that reflects only the unique contribution of the predictor and declines as its redundancy with the others increases. If `FALSE`, the predictor's *total* variance is used instead (the raw decomposition). For centered, orthogonal designs the two are identical; they differ only when predictors are correlated. See Details. Has no effect on interaction terms (whose components are always centered) or when `var_x` is supplied.

`ci` Logical. If `TRUE`, computes a confidence interval for `eta2p` by parametric bootstrap (`lme4::bootMer`): new responses are simulated from the fitted model, the model is refit, and `eta2p` is recomputed on each replicate, with percentile limits taken from the resulting distribution. This is the appropriate interval for a ratio of REML variance components, which has no closed-form sampling distribution. Each replicate refits the model, so this is opt-in and can be slow on large models. Default is `FALSE`.

`ci_level` Numeric confidence level for the bootstrap interval (default `0.95`).

`n_boot` Integer number of bootstrap replicates when `ci = TRUE` (default `1000`). Use a smaller value (e.g. 100-200) to gauge runtime on large models before committing to the full count.

`seed` Optional integer seed for reproducible bootstrap intervals.

`verbose` Logical. If `TRUE`, prints detailed results. Default is `FALSE`.

`x` An `eta2p_lmm` object.

...	Further arguments (ignored).
object	An eta2p_lmm object.

Details

Variance decomposition: The function implements a variance decomposition approach for computing partial eta-squared in mixed models. The variance a fixed effect explains is

$$\mathrm{Var}_{\text{explained}} = b^2 \times \sigma_X^2,$$

where b is the (partial) fixed-effect coefficient and σ_X^2 is the variance of the predictor column. Random slope variances are translated to the outcome scale by the same principle,

$$\sigma^2_{\text{slope}}(Y) = \sigma_b^2 \times \sigma_X^2.$$

For an interaction, σ_X^2 is the variance of the product of the *mean-centered* constituent predictors (see "Centering" below). The var_x argument bypasses this computation when the variance is known a priori.

Centering and intended scope: These quantities are exact when predictors are **mean-centered**, which is standard and recommended practice for the experimental and contrast-coded designs the method primarily targets (sum/contrast coding centers factors automatically). Two points:

- **Interactions: components are centered automatically.** For an interaction the function mean-centers each *constituent* predictor and then takes the variance of their product (it does *not* center the product itself). This makes the interaction effect size invariant to the location of the constituents: re-centering a predictor by a constant leaves the model fit and the interaction coefficient unchanged, and now also leaves the effect size unchanged. (The variance of the *raw* product depends on the constituents' means and so is not well defined as an effect size; this is why the centering is applied. The constituent predictors should still be centered before fitting for the coefficients themselves to be interpretable, but the effect-size value no longer depends on it.)
- **Main effects / random slopes use σ_X^2, which equals $E[X^2]$ only when X is centered.** The function does not center main-effect predictors (doing so would not change $\sigma_X^2 = \mathrm{Var}(X)$, which is already location-invariant), but the random-slope translation $\sigma_b^2\sigma_X^2$ is exact only for centered predictors; center continuous predictors before fitting.

Under correlated predictors, each b is a partial coefficient (it controls for the other predictors). By default (partial_predictors = TRUE) it is multiplied by the predictor's *unique* variance – the variance remaining after the part linearly predictable from the other predictors is removed – so the variance attributed to the predictor is

$$b^2 \,\mathrm{Var}(X \mid X_{others}) = b^2 \,\mathrm{Var}(X)\,\mathrm{tol}(X),$$

where $\mathrm{tol}(X) = 1 - R^2_{X\sim X_{others}}$ is the predictor's tolerance. (The two forms are algebraically identical.) This yields a semipartial variance-explained measure: it reflects only the unique contribution of the predictor and declines as its redundancy with the others increases, matching the unique-variance target estimated by an explained- error (model-comparison) approach.

Setting partial_predictors = FALSE instead uses the predictor's *total* variance σ_X^2. The resulting per-predictor values are partial (conditional) effect sizes that do not isolate unique variance

when predictors are correlated. For centered, orthogonal designs the two options coincide exactly, because the covariance among predictors is zero, so the choice matters only under collinearity.

The tolerance correction is also what makes the interaction handling location-invariant: centering an interaction's components is equivalent to the $\mathrm{Var} \times \mathrm{tol}$ of the product, which is constant across re-centering. The choice is independent of the operative option, which concerns the error denominator rather than the numerator. See Rights & Sterba (2019) on the partial-versus-total distinction.

General vs. operative effect sizes: For **general effect sizes** (default), all variance components appear in the denominator. For **operative effect sizes** (`operative = TRUE`), only components that contribute to the standard error of the effect are included. In mixed designs, each grouping factor (both crossed and nested) is independently classified as `"within"` or `"between"` the focal effect; between-subjects intercept variances are excluded from the operative denominator.

The operative option concerns only the denominator; it is independent of the `partial_predictors` setting, which controls the numerator. The two compose: an operative effect size uses the same numerator a general one would (unique variance by default), paired with the reduced operative denominator.

Mixed designs: Use `design = "mixed"` when the model contains both crossed and nested random effects simultaneously. A canonical example is participants viewing multiple photos of each model (stimulus): photos are nested within models, but both levels are crossed with participants.

The corresponding **lme4** model and `eta2p` call would be:

```
fit <- lmer(y ~ x + (1 | participant) + (1 | model) + (1 | photo:model),
            data = dat)
eta2p(fit, "x", dat,
      design     = "mixed",
      cross_vars = "participant",
      nest_vars  = c("photo", "model"))
```

Residual variance is attributed to the crossed side and counted exactly once; the nested calculator's residual is subtracted to prevent double-counting.

Crossed designs: Supports any number of grouping factors via `cross_vars`. The two-argument form (`subj_var` + `item_var`) is retained for backward compatibility and is equivalent to `cross_vars = c(subj_var, item_var)`.

Confidence intervals: With `ci = TRUE`, a confidence interval is obtained by parametric bootstrap via `lme4::bootMer`: responses are simulated from the fitted model, the model is refit, and `eta2p` is recomputed on each replicate; `ci_lower` and `ci_upper` are the percentile limits. Because `eta2p` is a ratio of REML variance components with no closed-form sampling distribution, this bootstrap is the appropriate interval rather than an analytic (e.g. noncentral) formula. Set `seed` for reproducibility, and start with a small `n_boot` to gauge runtime on large models.

Value

An object of class `"eta2p_lmm"`. Printing it shows the estimate on one line; `summary()` shows the full breakdown. It contains:

`eta2p`	Partial eta-squared value.

`variance_effect`
: Variance explained by the effect (numerator).

`coefficient`
: The fixed-effect coefficient used in the numerator (the fitted value, or `slope` when supplied).

`coefficient_fitted`
: The fitted coefficient (differs from `coefficient` only when `slope` is supplied).

`slope_override`
: The user-supplied `slope`, or `NULL`.

`var_x`
: The predictor's (total) variance, σ^2_X.

`var_x_unique`
: The unique predictor variance entering the numerator ($\sigma^2_X \times$ tolerance); equals `var_x` unless `partial_predictors` residualized a correlated predictor.

`tolerance`
: $\mathrm{var_x_unique}/\mathrm{var_x}$ (1 when uncorrelated with the other predictors).

`variance_error`
: Error variance (denominator).

`effect`
: Name of the effect.

`design`
: Design type: `"crossed"`, `"nested"`, or `"mixed"`.

`operative`
: Whether operative effect size was calculated.

`ci_level`, `ci_lower`, `ci_upper`, `boot_n`, `boot_method`
: (Only when `ci = TRUE`) The confidence level, the lower and upper percentile bootstrap limits, the number of usable bootstrap replicates, and the method label (`"parametric (bootMer)"`).

`variance_components`
: List of individual variance components. For `"mixed"` designs this is a list with `$crossed` and `$nested` sub-lists.

`within_between`
: (Mixed/crossed designs) Named list or vector classifying each grouping factor as `"within"` or `"between"` with respect to the focal effect. Only populated when `operative = TRUE`.

`cross_vars`
: (Crossed/mixed designs) Crossed grouping variable names.

`nest_vars`
: (Nested/mixed designs) Nested grouping variable names.

`n_per_factor`
: (Crossed designs) Number of units per crossed factor.

`n_cross`
: (Mixed designs) Number of units per crossed factor.

`n_nested`
: (Mixed designs) Number of units per nested factor.

`n_levels`
: (Nested designs) Number of units per nested level.

`effect_level`
: (Nested/mixed designs) Detected or supplied effect level.

Functions

- `print(eta2p_lmm)`: Concise print method: reports the estimate (and CI, if present) on one line. Use `summary()` for the full breakdown.
- `summary(eta2p_lmm)`: Comprehensive summary: the estimate plus the numerator ingredients (coefficient, var(X), tolerance) and the error variance components that make up the denominator.

References

Correll, J., Mellinger, C., McClelland, G. H., & Judd, C. M. (2020). Avoid Cohen's 'Small', 'Medium', and 'Large' for Power Analysis. *Trends in Cognitive Sciences*, 24(3), 200–207. doi:10.1016/j.tics.2019.12.009

Correll, J., Mellinger, C., & Pedersen, E. J. (2022). Flexible approaches for estimating partial eta squared in mixed-effects models with crossed random factors. *Behavior Research Methods*, 54, 1626–1642. doi:10.3758/s13428021016872

Rights, J. D., & Sterba, S. K. (2019). Quantifying explained variance in multilevel models: An integrative framework for defining R-squared measures. *Psychological Methods*, 24(3), 309–338. doi:10.1037/met0000184

See Also

`batch_eta2p` to compute per-coefficient values for all fixed effects at once.

Examples

```
library(lme4)

#  Two crossed factors (backward-compatible call)
set.seed(42)
crossed_data <- data.frame(
  y         = rnorm(120),
  condition = rep(c(-0.5, 0.5), 60),
  subject   = factor(rep(1:20, each = 6)),
  item      = factor(rep(1:6, 20))
)
model_c <- lmer(y ~ condition + (1 | subject) + (1 | item),
                data = crossed_data)
res <- eta2p(model_c, "condition", crossed_data,
             design   = "crossed",
             subj_var = "subject",
             item_var = "item")
res                 # concise: just the estimate
summary(res)        # full breakdown: coefficient, var(X), error components

#  Sensitivity analysis: the effect size a hypothetical slope would imply,
#  holding the error structure fixed (useful alongside simr)
eta2p(model_c, "condition", crossed_data,
      design   = "crossed",
      subj_var = "subject",
      item_var = "item",
      slope    = 0.03)

#  Three crossed factors using cross_vars
set.seed(42)
three_way_data <- data.frame(
  y         = rnorm(180),
  condition = rep(c(-0.5, 0.5), 90),
  subject   = factor(rep(1:20, each = 9)),
  item      = factor(rep(rep(1:6, each = 3), 10)),
```

```
  rater     = factor(rep(1:3, 60))
)
model_3 <- lmer(y ~ condition +
                  (1 | subject) + (1 | item) + (1 | rater),
                data = three_way_data)
eta2p(model_3, "condition", three_way_data,
      design     = "crossed",
      cross_vars = c("subject", "item", "rater"))

#  Mixed design: photos nested within models, crossed with participants
# Participants each view multiple photos of multiple models.
# Photos are nested within models (each photo belongs to one model),
# but both levels are crossed with participants.
set.seed(42)
n_subj  <- 30; n_model <- 10; n_photo <- 3
mixed_data <- expand.grid(
  participant = factor(seq_len(n_subj)),
  photo_id    = factor(seq_len(n_model * n_photo))
)
mixed_data$model_id <- factor(
  rep(seq_len(n_model), each = n_photo)[as.integer(mixed_data$photo_id)]
)
mixed_data$x <- rnorm(nrow(mixed_data))
mixed_data$y <- rnorm(nrow(mixed_data))

model_m <- lmer(y ~ x +
                  (1 | participant) +
                  (1 | model_id) +
                  (1 | photo_id:model_id),
                data = mixed_data)
eta2p(model_m, "x", mixed_data,
      design     = "mixed",
      cross_vars = "participant",
      nest_vars  = c("photo_id", "model_id"))

#  Supply predictor variance directly (var_x)
eta2p(model_c, "condition", crossed_data,
      design     = "crossed",
      cross_vars = c("subject", "item"),
      var_x      = 1)   # +/-1 binary predictor: var = 1 by design

#  Factor predictor (no manual recoding needed)
crossed_data$grp <- factor(rep(c("A", "B"), 60))
model_f <- lmer(y ~ grp + (1 | subject) + (1 | item), data = crossed_data)
# accepts the variable name "grp"; resolves to the coefficient automatically
eta2p(model_f, "grp", crossed_data,
      design = "crossed", cross_vars = c("subject", "item"))

#  Operative effect size
eta2p(model_c, "condition", crossed_data,
      design = "crossed", cross_vars = c("subject", "item"),
      operative = TRUE)
```

```
#  Total-variance (raw) numerator instead of the default unique variance
eta2p(model_c, "condition", crossed_data,
      design = "crossed", cross_vars = c("subject", "item"),
      partial_predictors = FALSE)

#  Confidence interval by parametric bootstrap
eta2p(model_c, "condition", crossed_data,
      design = "crossed", cross_vars = c("subject", "item"),
      ci = TRUE, n_boot = 200, seed = 1)

#  All fixed effects at once (per-coefficient)
batch_eta2p(model_c, crossed_data,
            design = "crossed", cross_vars = c("subject", "item"))
```

Index

batch_eta2p, 2, *9*

eta2p, 3

print.eta2p_lmm *(*eta2p*)*, 3

summary.eta2p_lmm *(*eta2p*)*, 3