

Package ‘mixedbiastest’

October 23, 2026

Type Package

Title Bias Diagnostic for Linear Mixed Models

Version 1.1.0

Date 2026-10-22

Maintainer Andrew T. Karl <akar1@asu.edu>

Description Provides a function to perform bias diagnostics on linear mixed models fitted with `lmer()` from the 'lme4' package. Compares an internal random-effects alignment diagnostic with a fitted permutation reference distribution, following Karl and Zimmerman (2021) <[doi:10.1016/j.jspi.2020.06.004](https://doi.org/10.1016/j.jspi.2020.06.004)>. Conditional response-refit simulations separately record random-effects alignment and residual-weight contributions to estimation error. Karl and Zimmerman (2020) <[doi:10.17632/tmynggddfm.1](https://doi.org/10.17632/tmynggddfm.1)> provide R code for implementing the test using 'mvglmmRank' output. Development and review were assisted by 'GPT o1-preview', 'GPT 5 Pro', and 'GPT6 Astra'; the author is responsible for the code and its interpretation.

Depends R (>= 3.5.0)

Imports lme4, Matrix, ggplot2, rlang

Suggests plm, lmerTest, testthat (>= 3.1.7)

License GPL-3

Encoding UTF-8

LazyData true

Config/testthat/edition 3

NeedsCompilation no

Config/roxygen2/version 8.0.0

Author Andrew T. Karl [cre, aut] (ORCID:
<<https://orcid.org/0000-0002-5933-8706>>)

Repository CRAN

Date/Publication 2026-10-22 23:40:08 UTC

Contents

mixedbiastest-package	2
example_data	3
list_fixed_effects	4
mixedbiastest	5
plot.mixedbiastest	7
print.mixedbiastest	8
print.mixedbiastest_simulation	8
simulate_bias	9

Index	12
--------------	-----------

mixedbiastest-package *mixedbiastest: Bias Diagnostics for Linear Mixed Models*

Description

The ‘mixedbiastest’ package computes an internal random-effects alignment diagnostic for Gaussian models fitted with ‘lmer’ from ‘lme4’, following Karl and Zimmerman (2021). Fitted permutation reference distributions describe the observed alignment. They are not generally exact tests of total bias. Conditional response-refit simulations retain both the random-effects and residual contributions to estimation error under a specified generating scenario.

Details

The methods in this package are designed for Gaussian linear mixed models with diagonal random-effects covariance matrices and homoskedastic residual errors with covariance $\sigma^2 I_n$. Models with more general random-effects structures or residual covariance patterns are currently not supported.

While the bias diagnostic of Karl and Zimmerman (2021) is derived for general linear mixed models with arbitrary random-effects and residual covariance matrices G and R , the current **mixedbiastest** implementation focuses on the practically important case described above. Extending the package to handle correlated random effects or more general residual covariance structures would require additional work on both the underlying linear algebra and the permutation scheme, and is therefore left for future research.

Functions

- `mixedbiastest` Computes the internal diagnostic and fitted permutation reference.
- `simulate_bias` Simulates conditional estimation error with covariance refitting.
- `print.mixedbiastest` Prints the results of the bias diagnostic.
- `plot.mixedbiastest` Plots fitted permutation reference distributions and observed internal diagnostics.
- `list_fixed_effects` List Fixed Effects from an merMod Object.

Acknowledgments

Development of this package was assisted by GPT o1-preview and GPT 5 Pro, which helped in constructing the structure of much of the code and the roxygen documentation. The code is based on the R code provided by Karl and Zimmerman (2020). GPT6 Astra assisted the 2026 review and revision; the author is responsible for the implementation and interpretation.

Author(s)

Maintainer: Andrew T. Karl <akar1@asu.edu> ([ORCID](#))

Authors:

- Andrew T. Karl <akar1@asu.edu> ([ORCID](#))

References

Karl, A. T., & Zimmerman, D. L. (2021). A diagnostic for bias in linear mixed model estimators induced by dependence between the random effects and the corresponding model matrix. *Journal of Statistical Planning and Inference*, 211, 107-118. doi:10.1016/j.jspi.2020.06.004 Author manuscript and revisions: <https://arxiv.org/abs/2003.08087>.

Karl, A., & Zimmerman, D. (2020). Data and Code Supplement for 'A Diagnostic for Bias in Linear Mixed Model Estimators Induced by Dependence Between the Random Effects and the Corresponding Model Matrix'. Mendeley Data, V1. doi:10.17632/tmynggddfm.1

example_data

Example Dataset for the mixedbiastest Package

Description

A dataset containing 97 observations of three variables: y, x, and group.

Usage

example_data

Format

A data frame with 97 rows and 3 variables:

y Numeric response variable.

x Numeric predictor variable.

group Integer indicating group membership.

list_fixed_effects	<i>List Fixed Effects from a merMod Object</i>
--------------------	--

Description

This function lists the fixed effects coefficients from an ‘lmerMod’/‘merMod’ object, providing the index and name of each coefficient. This can help users construct contrast vectors (‘k_list’) for use with the ‘mixedbiastest’ function. Only retained model-matrix columns are listed when the fit is rank deficient.

Usage

```
list_fixed_effects(model)
```

Arguments

model	An object that inherits from class merMod, such as an lmerMod or glmerMod object fitted using lmer or glmer from the lme4 package.
-------	---

Value

A data frame with two columns:

Index The index of each fixed effect coefficient.

Coefficient The name of each fixed effect coefficient.

Acknowledgments

Development of this package was assisted by GPT o1-preview, which helped in constructing the structure of much of the code and the roxygen documentation. The code is based on the R code provided by Karl and Zimmerman (2020).

Examples

```
if (requireNamespace("plm", quietly = TRUE) && requireNamespace("lme4", quietly = TRUE)) {
  library(lme4)
  data("Gasoline", package = "plm")
  # Fit a random effects model using lme4
  mixed_model <- lmer(lgaspcar ~ lincomep + lrpmpg + lcarpcap + (1 | country),
    data = Gasoline, REML = FALSE)

  # List fixed effects
  fixed_effects <- list_fixed_effects(mixed_model)
  print(fixed_effects)
}
```

mixedbiastest	<i>Bias Diagnostic for Linear Mixed Models</i>
---------------	--

Description

Computes an internal random-effects alignment diagnostic for fixed-effect contrasts in a model fitted with [lmer](#), and compares it with a permutation reference distribution.

Usage

```
mixedbiastest(model, n_permutations = 10000, k_list = NULL, verbose = FALSE)
```

Arguments

model	An object of class <code>lmerMod</code> (or a subclass, such as <code>lmerModLmerTest</code>) fitted using lmer from the lme4 package. Models fitted with glmer or with non-identity residual covariance structures (for example, non-unit weights) are not currently supported.
n_permutations	Integer. Number of permutations to perform (default is 10000). Must be a single positive integer no larger than <code>.Machine\$integer.max</code> .
k_list	Optional list of numeric vectors. Each vector specifies a linear combination of fixed effects to test. If <code>NULL</code> , each fixed effect is tested individually. Vectors must be finite and follow the order returned by <code>list_fixed_effects(model)</code> , after any rank-deficient columns are dropped. The list must be nonempty; supplied list names must be nonempty and unique.
verbose	Logical. If <code>TRUE</code> , prints detailed messages during execution.

Details

Let $W = (X'V^{-1}X)^{-1}X'V^{-1}$ use the fitted covariance parameters. The returned diagnostic is $k'WZ\hat{b}$. For a response model $y = X\beta + Zb + e$ (after subtracting any known offsets), the estimation error is $k'WZb + k'We$. Covariance estimates depend on the response, so the expectation of the second term need not vanish. The internal diagnostic substitutes predicted random effects for b ; it is not an unbiased estimate of total bias and should not be automatically subtracted from a coefficient estimate. See [simulate_bias](#) for conditional response-refit simulations that retain both terms.

The permutations hold fitted weights and predicted random effects fixed. Their p-values describe this fitted reference distribution. Exchangeability of latent random effects alone does not make their shrunken predictions exchangeable, particularly with unequal information across groups. Thus these p-values are diagnostic reference values, not generally exact tests of zero total bias or of independence under a stochastic assignment mechanism.

The implementation follows Karl and Zimmerman (2021) and is currently restricted to:

- Gaussian linear mixed models fitted by [lmer](#) (no GLMMs).
- Diagonal random-effects covariance matrices (i.e., a block-diagonal G with scalar blocks for each random-effect coefficient), including diagonal terms specified with `diag()` in recent versions of **lme4**.

- Homoskedastic residual errors with covariance $\sigma^2 I_n$ (no observation weights or residual correlation structures).

In particular, models fitted with `glmer` or with non-identity residual covariance structures (for example, non-unit observation weights) are beyond the scope of the current implementation.

Random effects are permuted separately within each coefficient of each random-effects term. Automatically scaled fixed-effect predictors are handled on the original coefficient scale, including the intercept adjustment. A singular fit produces a warning; zero variance components are retained as zero. Results retain full numerical precision. The two-sided permutation p-value uses the correction $(1 + B)/(1 + M)$, where B is the number of sampled absolute statistics at least as large as the observed absolute statistic, and M is `n_permutations`. Use `set.seed()` before calling this function for reproducible permutations.

While the diagnostic of Karl and Zimmerman (2021) is formulated for general linear mixed models with arbitrary covariance matrices G and R , this function implements the special case of Gaussian lmer models with diagonal G and homoskedastic residual errors. Extending `mixedbiastest()` to correlated random effects or more general residual covariance structures would require substantial additional work on the underlying linear algebra and permutation scheme, and is left for future research.

See the `list_fixed_effects` function if you would like to construct contrasts of fixed effects to be used as `k_list`.

Value

An object of class "mixedbiastest" containing:

`results_table` A data frame with the test results for each fixed effect or contrast, including the internal diagnostic (the legacy column name is `Bias_Estimate`) and permutation-reference p-values.

`permutation_values` A list of numeric vectors containing permutation values for each fixed effect or contrast.

`model` The original lmerMod model object provided as input.

Acknowledgments

Development of this package was assisted by GPT o1-preview and GPT 5 Pro, which helped in constructing the structure of much of the code and the roxygen documentation. The code is based on the R code provided by Karl and Zimmerman (2020). GPT6 Astra assisted the 2026 review and revision; the author is responsible for the implementation and its interpretation.

References

- Karl, A. T., & Zimmerman, D. L. (2021). A diagnostic for bias in linear mixed model estimators induced by dependence between the random effects and the corresponding model matrix. *Journal of Statistical Planning and Inference*, 211, 107-118. doi:10.1016/j.jspi.2020.06.004 Author manuscript and revisions: <https://arxiv.org/abs/2003.08087>.
- Karl, A., & Zimmerman, D. (2020). Data and Code Supplement for "A Diagnostic for Bias in Linear Mixed Model Estimators Induced by Dependence Between the Random Effects and the Corresponding Model Matrix". Mendeley Data, V1. doi:10.17632/tmynggddfm.1

Examples

```

if (requireNamespace("plm", quietly = TRUE) && requireNamespace("lme4", quietly = TRUE)) {
  library(lme4)
  data("Gasoline", package = "plm")
  mixed_model <- lmer(lgaspcar ~ lincomep + lrpmpg + lcarpcap + (1 | country),
                     data = Gasoline)
  result <- mixedbiastest(mixed_model)
  print(result); plot(result)
}
if (requireNamespace("lme4", quietly = TRUE)) {
  library(lme4)
  example_model <- lmer(y ~ x + (1 | group), data = example_data)
  result2 <- mixedbiastest(example_model)
  print(result2); plot(result2)

  # Simulate data
  set.seed(123)
  n_groups <- 30
  n_obs_per_group <- 10
  group <- rep(1:n_groups, each = n_obs_per_group)
  x <- runif(n_groups * n_obs_per_group)
  beta0 <- 2; beta1 <- 5
  sigma_u <- 1; sigma_e <- 0.5
  u <- rnorm(n_groups, 0, sigma_u)
  e <- rnorm(n_groups * n_obs_per_group, 0, sigma_e)
  y <- beta0 + beta1 * x + u[group] + e

  data_sim <- data.frame(y = y, x = x, group = factor(group))
  model3 <- lmer(y ~ x + (1 | group), data = data_sim)
  result3 <- mixedbiastest(model3, verbose = TRUE)
  plot(result3)
}

```

plot.mixedbiastest *Plot Method for Bias Diagnostic Results*

Description

Plots fitted permutation reference distributions and observed internal diagnostics.

Usage

```

## S3 method for class 'mixedbiastest'
plot(x, bins = 30, ...)

```

Arguments

x	An object of class "mixedbiastest".
bins	Integer, number of bins for the histograms (default 30).
...	Additional arguments (currently not used).

Value

A ggplot object (returned invisibly) showing permutation distributions for all fixed effects.

```
print.mixedbiastest
```

Print Method for Bias Diagnostic Results

Description

Prints the results of the bias diagnostic in a formatted table.

Usage

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

Arguments

```
x
```

An object of class "mixedbiastest".

```
...
```

Additional arguments (currently not used).

Value

The input object, returned invisibly.

```
print.mixedbiastest_simulation
```

Print conditional response-refit simulation summaries

Description

Shows the generating scenario, success counts and component means/MCSEs without printing every replicate or the original fitted model.

Usage

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

Arguments

```
x
```

An object of class "mixedbiastest_simulation".

```
digits
```

Significant digits displayed for means and Monte Carlo errors.

```
...
```

Additional arguments (currently unused).

Value

The input object, returned invisibly.

simulate_bias

*Fixed-design response-refit simulation***Description**

Examines fixed-effect estimation error under a specified conditional generating model, refitting covariance parameters for every simulated response.

Usage

```
simulate_bias(
  model,
  nsim = 1000,
  beta = NULL,
  k_list = NULL,
  generator = NULL,
  centered_errors = is.null(generator)
)
```

Arguments

- | | |
|-----------------|---|
| model | An object of class <code>lmerMod</code> (or a subclass, such as <code>lmerModLmerTest</code>) fitted using <code>lmer</code> from the lme4 package. Models fitted with <code>glmer</code> or with non-identity residual covariance structures (for example, non-unit weights) are not currently supported. |
| nsim | Number of independent response simulations, an integer at least 2. |
| beta | Generating fixed-effect coefficients on the original scale, in the retained order returned by <code>list_fixed_effects(model)</code> . Defaults to the fitted coefficients (<code>NULL</code>), with original-scale reconstruction for autoscaled rank-deficient designs. Named coefficients must have that exact order. |
| k_list | Optional list of numeric vectors. Each vector specifies a linear combination of fixed effects to test. If <code>NULL</code> , each fixed effect is tested individually. Vectors must be finite and follow the order returned by <code>list_fixed_effects(model)</code> , after any rank-deficient columns are dropped. The list must be nonempty; supplied list names must be nonempty and unique. |
| generator | Optional function with arguments <code>model</code> and <code>replicate</code> . It must return a list with finite numeric vectors <code>random_effects</code> and <code>residual_errors</code> , ordered as <code>lme4::getME(model, "b")</code> and the rows used in the fit, respectively. The default holds the original predicted random effects fixed and draws independent Gaussian errors with the original fitted residual variance. |
| centered_errors | Logical declaration that the generating residual errors have mean zero at the fixed design. Defaults to <code>TRUE</code> for the default generator and <code>FALSE</code> for a custom generator. If true, also summarize a paired control-variate estimate of the residual contribution. |

Details

The design matrices, offsets, fitting method (ML or REML), and original model structure are held fixed. The default response is $y_s = X\beta + Z\hat{b} + o + e_s$, where o is the known offset vector. A custom generator replaces $\hat{b}$ and e_s with its own draws. `lme4::refit` estimates covariance parameters again for each response. If W_s is the resulting fitted GLS map from offset-adjusted responses to coefficients, the recorded terms are $k'W_sZb_s$ (Alignment), $k'W_se_s$ (Residual), their sum (Total), the actual coefficient error (Estimation_Error), and the internal diagnostic $k'W_sZ\hat{b}_s$ (Internal_Diagnostic). A numerical check verifies the per-replicate decomposition. Total and Estimation_Error agree up to numerical precision; they are both retained for inspection.

When `centered_errors = TRUE`, `Residual_CV` is $k'(W_s - W_0)e_s$, where W_0 uses the original fitted covariance. Its expectation equals the residual contribution because W_0 is fixed and the declared residual mean is zero. `Total_CV` is `Alignment + Residual_CV`. These paired estimates can reduce Monte Carlo error, but need not do so in every scenario. Raw errors remain available.

This is a sensitivity analysis of the supplied conditional generating model. Holding predicted random effects and the observed design fixed does not reproduce a population assignment mechanism or uncertainty about that mechanism. Results condition on the plug-in generating coefficients, predicted effects and residual variance; uncertainty in these estimates is not propagated. Predicted effects are shrunk, so refitted covariance parameters and mean Alignment can differ from the original fit and its internal diagnostic. A custom generator specifies a scenario; it does not identify the true bias from the fitted data. Stochastic or endogenous X or Z requires an external joint generation and refitting workflow. No automatic coefficient correction, bias test, or confidence interval is produced.

Use `set.seed()` for reproducibility. Simulations run sequentially. Monte Carlo standard errors are the empirical standard deviation divided by the square root of the successful replicate count, requiring independent replicates with finite variances. Custom generators must meet those conditions for that interpretation. Failed or nonconverged refits remain in replicates and `fit_status`; their numeric values are missing. Summaries explicitly report successful and failed counts and, after a failure, describe successful refits only. If any refit fails or does not converge, control-variate summary means and MCSEs are withheld: selection of successful refits can make the baseline residual mean nonzero, so target equivalence no longer follows. Per-replicate control-variate values remain inspectable. Singular fits are retained and counted. Numerical validation failures have status `identity_check_failed` or `validation_failed`, separately from optimizer errors. The identity tolerance is $1e-6$ times the maximum of one and the absolute recorded error contributions. Extreme coefficient scales combined with ill-conditioned designs can fail this guard; centering predictors or representing a known large response baseline as an offset can help.

With `lme4` autoscaling and no fixed intercept, the fitted columns can be centered even though the original formula has no intercept. This function preserves that fitted centered design and reports coefficients in their original units, matching `lme4::fixef`. It does not reinterpret the fit as a regression on raw uncentered columns. Refit with autoscaling disabled when the raw no-intercept design is intended.

Value

A list of class `"mixedbiastest_simulation"`, with `summary` (component means, MCSEs and success/failure counts), `replicates` (one row per contrast and replicate), `fit_status` (errors, convergence, singularity and messages), `counts`, `beta`, `centered_errors`, `generator_type`, and `model`.

See Also[mixedbiastest](#)**Examples**

```
unbalanced <- lme4::sleepstudy[-c(1:3, 25:31), ]
fit <- lme4::lmer(Reaction ~ Days + (1 | Subject), unbalanced)
set.seed(20261022)
# Small example only; use more replications to obtain useful MCSEs.
sim <- simulate_bias(fit, nsim = 10, k_list = list(Days = c(0, 1)))
sim$summary
```

Index

* **datasets**

example_data, [3](#)

example_data, [3](#)

glmer, [4-6](#), [9](#)

list_fixed_effects, [2](#), [4](#), [6](#)

lmer, [4](#), [5](#), [9](#)

mixedbiastest, [2](#), [5](#), [11](#)

mixedbiastest-package, [2](#)

plot.mixedbiastest, [2](#), [7](#)

print.mixedbiastest, [2](#), [8](#)

print.mixedbiastest_simulation, [8](#)

simulate_bias, [2](#), [5](#), [9](#)