

# Package ‘lmeInfo’

February 15, 2023

**Type** Package

**Title** Information Matrices for 'lmeStruct' and 'glsStruct' Objects

**Version** 0.3.1

**Description** Provides analytic derivatives and information matrices for fitted linear mixed effects (lme) models and generalized least squares (gls) models estimated using `lme()` (from package 'nlme') and `gls()` (from package 'nlme'), respectively. The package includes functions for estimating the sampling variance-covariance of variance component parameters using the inverse Fisher information. The variance components include the parameters of the random effects structure (for lme models), the variance structure, and the correlation structure. The expected and average forms of the Fisher information matrix are used in the calculations, and models estimated by full maximum likelihood or restricted maximum likelihood are supported. The package also includes a function for estimating standardized mean difference effect sizes (Pustejovsky, Hedges, and Shadish (2014) <[DOI:10.3102/1076998614547577](https://doi.org/10.3102/1076998614547577)>) based on fitted lme or gls models.

**URL** <https://jepusto.github.io/lmeInfo/>

**BugReports** <https://github.com/jepusto/lmeInfo/issues>

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 4.0.0)

**Suggests** covr, testthat (>= 2.1.0), knitr, rmarkdown, scdhlms, mlmRev, carData, lme4, Matrix, merDeriv

**Imports** nlme, stats

**VignetteBuilder** knitr

**RoxygenNote** 7.2.3

**Language** en-US

**NeedsCompilation** no

**Author** James Pustejovsky [aut] (<<https://orcid.org/0000-0003-0591-9465>>),  
Man Chen [aut, cre]

**Maintainer** Man Chen <manchen9005@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-02-15 15:00:02 UTC

## R topics documented:

Bryant2016 . . . . .	2
CI_g . . . . .	3
extract_varcomp . . . . .	4
Fisher_info . . . . .	5
g_mlm . . . . .	6
varcomp_vcov . . . . .	8

**Index** **10**

---

Bryant2016	<i>Bryant et al. (2016)</i>
------------	-----------------------------

---

## Description

Data from a multi-level multiple baseline design conducted by Bryant et al. (2016). The study involved collecting repeated measures of math performance on twelve students nested in three schools. The variables are as follows:

- Study\_ID Study identifier.
- school School identifier.
- case Student identifier.
- session Measurement occasion.
- treatment Indicator for treatment phase.
- outcome Texas Early Mathematics Inventory (TEMI-Aim Check) scores.
- trt\_time Measurement occasion times treatment phase.
- session\_c Measurement occasion centered at the follow-up time (Measurement occasion 9).

## Format

A data frame with 299 rows and 8 variables

## Source

Bryant, B. R., Bryant, D. P., Porterfield, J., Dennis, M. S., Falcomata, T., Valentine, C., Brewer, C., & Bell, K. (2016). The effects of a Tier 3 intervention on the mathematics performance of second grade students with severe mathematics difficulties. *Journal of Learning Disabilities, 49*(2), 176-188. doi:[10.1177/0022219414538516](https://doi.org/10.1177/0022219414538516)

---

CI_g	<i>Calculates a confidence interval for a standardized mean difference effect size</i>
------	--

---

### Description

Calculates a confidence interval for a `g_mlm` object, using either a central t distribution (for a symmetric interval) or a non-central t distribution (for an asymmetric interval).

### Usage

```
CI_g(g, cover = 0.95, bound = 35, symmetric = TRUE)
```

### Arguments

<code>g</code>	an estimated effect size object of class <code>g_mlm</code> .
<code>cover</code>	confidence level.
<code>bound</code>	numerical tolerance for non-centrality parameter in <a href="#">qt</a> .
<code>symmetric</code>	If TRUE (the default), use a symmetric confidence interval. If FALSE, use a non-central t approximation to obtain an asymmetric confidence interval.

### Value

A vector of lower and upper confidence bounds.

### Examples

```
library(nlme)
data(Bryant2016, package = "lmeInfo")
Bryant2016_RML1 <- lme(fixed = outcome ~ treatment,
                    random = ~ 1 | school/case,
                    correlation = corAR1(0, ~ session | school/case),
                    data = Bryant2016)
Bryant2016_g1 <- g_mlm(Bryant2016_RML1, p_const = c(0,1), r_const = c(1,1,0,1),
                    infotype = "expected")
CI_g(Bryant2016_g1, symmetric = TRUE)
CI_g(Bryant2016_g1, symmetric = FALSE)
```

---

extract_varcomp	<i>Extract estimated variance components</i>
-----------------	--

---

### Description

Extracts the estimated variance components from a fitted linear mixed effects model (`lmeStruct` object) or generalized least squares model (`glsStruct` object).

### Usage

```
extract_varcomp(mod, separate_variances, vector)
```

### Arguments

<code>mod</code>	Fitted model of class <code>lmeStruct</code> or <code>glsStruct</code> .
<code>separate_variances</code>	Logical indicating whether to return the separate level-1 variance components for each stratum if using <code>varIdent</code> function to allow for different variances per stratum. Default is <code>FALSE</code> .
<code>vector</code>	Logical indicating whether to return the variance components as a numeric vector. Default is <code>FALSE</code> .

### Value

If `vector = FALSE`, an object of class `varcomp` consisting of a list of estimated variance components. Models that do not include correlation structure parameters or variance structure parameters will have empty lists for those components. If `vector = TRUE`, a numeric vector of estimated variance components.

If `separate_variances = TRUE` and if `weights = varIdent(form = ~ 1 | Stratum)` is specified in the model fitting, separate level-1 variance estimates will be returned for each stratum. If `separate_variances = TRUE` but if the weighting structure is not specified with `varIdent`, or if `separate_variances = FALSE`, then no separate level-1 variance estimates will be returned.

### Examples

```
library(nlme)
data(Bryant2016)
Bryant2016_RML <- lme(fixed = outcome ~ treatment,
                    random = ~ 1 | school/case,
                    correlation = corAR1(0, ~ session | school/case),
                    weights = varIdent(form = ~ 1 | treatment),
                    data = Bryant2016)
extract_varcomp(Bryant2016_RML, separate_variances = FALSE)
extract_varcomp(Bryant2016_RML, separate_variances = TRUE)
extract_varcomp(Bryant2016_RML, vector = TRUE)
```

---

Fisher_info	<i>Calculate expected, observed, or average Fisher information matrix</i>
-------------	---

---

**Description**

Calculates the expected, observed, or average Fisher information matrix from a fitted linear mixed effects model (lmeStruct object) or generalized least squares model (glsStruct object).

**Usage**

```
Fisher_info(mod, type = "expected", separate_variances = FALSE)
```

**Arguments**

mod	Fitted model of class lmeStruct or glsStruct.
type	Type of information matrix. One of "expected" (the default), "observed", or "average".
separate_variances	Logical indicating whether to return the Fisher information matrix for separate level-1 variance components if using varIdent function to allow for different variances per stratum. Default is FALSE.

**Value**

Information matrix corresponding to variance component parameters of mod.

If separate\_variances = TRUE and if weights = varIdent(form = ~ 1 | Stratum) is specified in the model fitting, the Fisher information matrix for separate level-1 variance estimates will be returned. If separate\_variances = TRUE but if the weighting structure is not specified with varIdent, or if separate\_variances = FALSE, then the Fisher information matrix for the default variance components will be returned.

**Examples**

```
library(nlme)
data(Bryant2016)
Bryant2016_RML <- lme(fixed = outcome ~ treatment,
                    random = ~ 1 | school/case,
                    correlation = corAR1(0, ~ session | school/case),
                    data = Bryant2016)
Fisher_info(Bryant2016_RML, type = "expected")
Fisher_info(Bryant2016_RML, type = "average")

Bryant2016_RML2 <- lme(fixed = outcome ~ treatment,
                    random = ~ 1 | school/case,
                    correlation = corAR1(0, ~ session | school/case),
                    weights = varIdent(form = ~ 1 | treatment),
                    data = Bryant2016)
```

```
Fisher_info(Bryant2016_RML2, separate_variances = TRUE)
```

---

g\_mlm

*Calculates adjusted mlm effect size*


---

### Description

Estimates a standardized mean difference effect size from a fitted multi-level model, using restricted or full maximum likelihood methods with small-sample correction, as described in Pustejovsky, Hedges, & Shadish (2014).

### Usage

```
g_mlm(
  mod,
  p_const,
  mod_denom = mod,
  r_const = NULL,
  infotype = "expected",
  separate_variances = FALSE,
  ...
)
```

### Arguments

mod	Fitted model of class <code>lmeStruct</code> (estimated using <code>nlme::lme()</code> ) or of class <code>glsStruct</code> (estimated using <code>nlme::gls()</code> ), from which to estimate the numerator of the effect size.
p_const	Vector of constants for calculating numerator of effect size. Must be the same length as fixed effects in <code>mod</code> .
mod_denom	Fitted model of class <code>lmeStruct</code> (estimated using <code>nlme::lme()</code> ) or of class <code>glsStruct</code> (estimated using <code>nlme::gls()</code> ), from which to estimate the denominator of the effect size. If not otherwise specified, the same model will be used for the numerator and the denominator calculations.
r_const	Vector of constants for calculating denominator of effect size. Must be the same length as the number of variance component parameters in <code>mod_denom</code> .
infotype	Type of information matrix. One of "expected" (the default), "observed", or "average".
separate_variances	Logical indicating whether to incorporate separate level-1 variance components in the calculation of the effect size and standard error for models with a 'varIdent()' variance structure. If TRUE, make sure the <code>r_const</code> matches the parameterization of the variance component as returned by <code>extract_varcomp(mod, separate_variances = TRUE)</code> . Default is FALSE.
...	further arguments.

**Value**

A list with the following components

p_beta	Numerator of effect size
r_theta	Squared denominator of effect size
delta_AB	Unadjusted (mlm) effect size estimate
nu	Estimated denominator degrees of freedom
J_nu	Biased correction factor for effect size estimate
kappa	Scaled standard error of numerator
g_AB	Corrected effect size estimate
SE_g_AB	Approximate standard error estimate
theta	Estimated variance component parameters
info_inv	Inversed information matrix

**References**

Pustejovsky, J. E., Hedges, L. V., & Shadish, W. R. (2014). Design-comparable effect sizes in multiple baseline designs: A general modeling framework. *Journal of Educational and Behavioral Statistics*, 39(4), 211-227. doi:10.3102/1076998614547577

**Examples**

```
library(nlme)
data(Bryant2016, package = "lmeInfo")
Bryant2016_RML1 <- lme(fixed = outcome ~ treatment,
                    random = ~ 1 | school/case,
                    correlation = corAR1(0, ~ session | school/case),
                    data = Bryant2016)
Bryant2016_g1 <- g_mlm(Bryant2016_RML1, p_const = c(0,1), r_const = c(1,1,0,1),
                    infotype = "expected")
print(Bryant2016_g1)
summary(Bryant2016_g1)

Bryant2016_RML2 <- lme(fixed = outcome ~ treatment,
                    random = ~ 1 | school/case,
                    correlation = corAR1(0, ~ session | school/case),
                    weights = varIdent(form = ~ 1 | treatment),
                    data = Bryant2016)
Bryant_g <- g_mlm(Bryant2016_RML2, p_const = c(0,1), r_const = c(1,1,0,0,1))
Bryant_g_baseline <- g_mlm(Bryant2016_RML2,
                    p_const = c(0,1),
                    r_const = c(1,1,0,1,0),
                    separate_variances = TRUE)
Bryant_g_treatment <- g_mlm(Bryant2016_RML2,
                    p_const = c(0,1),
                    r_const = c(1,1,0,0,1),
                    separate_variances = TRUE)

print(Bryant_g)
```

```
print(Bryant_g_baseline)
print(Bryant_g_treatment)
```

---

varcomp_vcov	<i>Estimated sampling variance-covariance of variance component parameters.</i>
--------------	---

---

### Description

Estimate the sampling variance-covariance of variance component parameters from a fitted linear mixed effects model (`lmeStruct` object) or generalized least squares model (`glsStruct` object) using the inverse Fisher information.

### Usage

```
varcomp_vcov(mod, type = "expected", separate_variances = FALSE)
```

### Arguments

<code>mod</code>	Fitted model of class <code>lmeStruct</code> or <code>glsStruct</code> .
<code>type</code>	Type of information matrix. One of "expected" (the default), "observed", or "average".
<code>separate_variances</code>	Logical indicating whether to return the Fisher information matrix for separate level-1 variance components if using <code>varIdent</code> function to allow for different variances per stratum. Default is FALSE.

### Value

Sampling variance-covariance matrix corresponding to variance component parameters of `mod`.

### Examples

```
library(nlme)
data(Bryant2016)
Bryant2016_RML <- lme(fixed = outcome ~ treatment,
                    random = ~ 1 | school/case,
                    correlation = corAR1(0, ~ session | school/case),
                    data = Bryant2016)
varcomp_vcov(Bryant2016_RML, type = "expected")

Bryant2016_RML2 <- lme(fixed = outcome ~ treatment,
                     random = ~ 1 | school/case,
                     correlation = corAR1(0, ~ session | school/case),
                     weights = varIdent(form = ~ 1 | treatment),
                     data = Bryant2016)
```



*varcomp\_vcov*

9

```
varcomp_vcov(Bryant2016_RML2, separate_variances = TRUE)
```

# Index

## \* datasets

Bryant2016, 2

Bryant2016, 2

CI\_g, 3

extract\_varcomp, 4

Fisher\_info, 5

g\_mlm, 6

qt, 3

varcomp\_vcov, 8