Package 'mlmhelpr'

July 23, 2025

Title Multilevel/Mixed Model Helper Functions

Version 0.1.1

Description A collection of miscellaneous helper function for running multilevel/mixed models in 'lme4'. This package aims to provide functions to compute common tasks when estimating multilevel models such as computing the intraclass correlation and design effect, centering variables, estimating the proportion of variance explained at each level, pseudo-R squared, random intercept and slope reliabilities, tests for homogeneity of variance at level-1, and cluster robust and bootstrap standard errors. The tests and statistics reported in the package are from Raudenbush & Bryk (2002, ISBN:9780761919049), Hox et al. (2018, ISBN:9781138121362), and Snijders & Bosker (2012, ISBN:9781849202015).

License MIT + file LICENSE

URL <https://github.com/lrocconi/mlmhelpr>

BugReports <https://github.com/lrocconi/mlmhelpr/issues>

Depends R (>= 2.10)

Imports lme4, stats, utils, Rdpack, mathjaxr

RdMacros Rdpack, mathjaxr

Suggests clubSandwich, testthat (>= 3.0.0), knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Config/testthat/edition 3

NeedsCompilation no

Author Louis Rocconi [aut, cre] (ORCID: <https://orcid.org/0000-0003-0693-0957>), Anthony Schmidt [aut] (ORCID: <https://orcid.org/0000-0003-4478-0638>)

Maintainer Louis Rocconi <lrocconi@utk.edu>

Repository CRAN

Date/Publication 2024-05-21 20:20:02 UTC

Contents

---

boot_se   *Bootstrap Standard Errors (experimental)*

Description

Computes bootstrapped standard errors for fixed effects. z-test returned using a standard normal reference distribution (interpret with caution)

Usage

```
boot_se(model, nsim = 5, seed = 1234, pct = 95, ...)
```

Arguments

model	a mixed model produced using the lme4 package (lmer or glmer functions). This is an object of class merMod. This function is a wrapper for lme4::bootMer
nsim	number of bootstrap samples to compute. Defaults to 5 but should be closer to 1,000 or 5,000. Note this is time intensive.
seed	random number seed for reproducibility. Defaults to 1234.
pct	percentage level for confidence interval. Defaults to 95.
...	additional parameters to pass to lme4::bootMer. Not currently implemented.

Value

A list containing a data frame with coefficient estimates and number of bootstrapped samples.

Examples

```
# lmer example
fit <- lme4::lmer(mathach ~ 1 + ses + catholic + (1|id),
data=hsb, REML=TRUE)

boot_se(fit)



# run time > 10s
# glmer example: logistic
  # Create binary outcome
hsb$binary_math <- ifelse(hsb$mathach <= 13, 0, 1)

fitb <- lme4::glmer(binary_math ~ 1 + ses + catholic + (1|id),
data=hsb, family = binomial(link="logit"))

boot_se(fitb)
```

---

center | *Automatically grand-mean or group-mean center a fitted object*

Description

This function refits a model using grand-mean centering, group-mean centering (if a grouping variable is specified), or centering at a user-specified value

Usage

```
center(
  x,
  grand_variables = NULL,
  group = NULL,
  group_variables = NULL,
  value = NULL,
  value_variables = NULL
)
```

Arguments

x
: A model produced using the `lme4::lmer()` function. This is an object of class `merMod` and subclass `lmerMod`.

grand_variables
: one or more variables to center at the grand-mean

group
: Grouping variable. If a grouping variable is specified, group-mean centering (also known as centering within cluster) based on that variable will be performed.

group_variables

          Variables to be group-mean centered.

value   Center at a specific value rather than the grand mean

value_variables

          Variables to be centered at user-specified value rather than the grand mean

Value

A newly fitted model with centered variables

Examples

```
fit <- lme4::lmer(mathach ~ 1 + ses + catholic + (1|id),
data=hsb, REML=TRUE)

# Centering a single variable around the grand mean
fit_gmc <- center(fit, grand_variables="ses")

# Centering multiple variables around the grand mean
fit_gmc <- center(fit, grand_variables=c("ses", "catholic"))

# Centering variables around the group means
fit_cwg <- center(fit, group="id", group_variables="ses")

# Centering variables using different strategies
fit_mixed <- center(fit, group = "id", group_variables = "ses", grand_variables = "catholic")
```

---

design_effect   *Design Effect*

Description

The design effect quantifies the degree a sample deviates from a simple random sample. In the multilevel modeling context, this can be used to determine whether clustering will bias standard errors and whether the assumption of independence is held. Thus, it can help determine whether multilevel modeling is appropriate for a given data set. The calculations are based on (Hox et al., 2018) and uses the mlmhelpr:icc function. A rule of thumb is that design effects smaller than 2 may indicate multilevel modeling is not necessary; however, this is dependent on cluster size and other factors (Lai et al., 2015).

Note: For models with random slopes, it is generally advised to interpret with caution. According to Kreft and De Leeuw (1998), "The concept of intra-class correlation is based on a model with a random intercept only. No unique intra-class correlation can be calculated when a random slope is present in the model" (p. 63). Since the intra-class correlation is part of the design effect calculation, caution is advised when interpreting models with random slopes.

Usage

```
design_effect(x, median = FALSE)
```

Arguments

x	model produced using the `lme4::lmer()` function. This is an object of class `merMod` and subclass `lmerMod`.
median	Boolean argument (TRUE/FALSE) to use the median cluster size to compute the design effect. By default, the average cluster size is used.

Value

a data frame containing the cluster variable, number of clusters, average (or median) cluster size, intraclass correlation, and the design effect

References

Hox JJ, Moerbeek M, van de Schoot R (2018). *Multilevel Analysis: Techniques and Applications*. Taylor and Francis. ISBN 9781138121362.

Lai MHC, Kwok O (2015). "Examining the Rule of Thumb of Not Using Multilevel Modeling: The "Design Effect Smaller Than Two" Rule." *The Journal of Experimental Education*, **83**(3), 423–438. ISSN 0022-0973, 1940-0683, doi:10.1080/00220973.2014.907229.

Kreft, Ita, de Leeuw, Jan (1998). *Introducing Multilevel Modeling*. Sage Publications. ISBN 0761951405.

Examples

```
fit <- lme4::lmer(mathach ~ 1 + ses + catholic + (1|id),
data=hsb, REML=TRUE)

design_effect(fit)
```

---

hausman   *Hausman Test (experimental)*

Description

The Hausman test tests whether there are significant differences between fixed effect and random effect models with similar specifications. If the test statistic is *not* statistically significant, a random effects models (i.e. a multilevel model) *may* be more suitable (efficient). This function takes a model estimated with `lme4::lmer`, automatically re-estimates a fixed effects model, applies the Hausman test, and returns the test statistic and p-value.

The Hausman test is based on (Fox, 2016, p. 732, footnote 46). The Hausman test statistic is distributed as chi-square with degrees of freedom equal to the number of coefficients.

Note: The selection of a mixed effect (random effect/multilevel) model should not be solely driven by the Hausman test or any other single statistic. Proper model selection should reflect the research questions and nested nature of the data. In addition, Fox suggests that "the choice between random and fixed effects should reflect our view of the process that generates the data" (p. 732). See also https://stats.stackexchange.com/questions/502811/should-a-hausman-test-be-used-to-decide-between-fixed-vs-random-effects for a discussion of the test and its results.

Usage

```
hausman(re_model)
```

Arguments

re_model   model produced using the `lme4::lmer()` function. This is an object of class
                `merMod` and subclass `lmerMod`.

Value

an object of class "htest"

References

Fox J, Fox J (2016). *Applied Regression Analysis and Generalized Linear Models*, Third Edition
edition. SAGE, Los Angeles. ISBN 978-1-4522-0566-3.

Examples

```
fit <- lme4::lmer(mathach ~ 1 + ses + catholic + (1|id),
data=hsb, REML=TRUE)

hausman(fit)
```

---

hsb   *HSB: High School and Beyond Data*

Description

This data is a modified subsample from the 1982 High School and Beyond Survey and is used
extensively in *Hierarchical Linear Models* by Raudenbush and Bryk. The data file, called hsb,
consists of 7,185 students nested in 160 schools. The outcome variable of interest is the student-
level (level 1) math achievement score (mathach). The variable ses is the socio-economic status of
a student and therefore is at the student level. The variable meanses is the average SES for each
school and therefore is at the school level (level 2). The variable sector is a variable indicating if
a school is public or catholic and is therefore a school-level variable. There are 90 public schools
(sector=0) and 70 catholic schools (sector=1) in the sample.

Usage

```
hsb
```

Format

A data frame with 7185 rows and 11 variables:

id school identification number

minority ethnicity status: other, minority

female gender status: female, male

ses socioeconomic status based on a standardized scale constructed from measures of parental occupation, education, and income

mathach a measure of math achievement

size school enrollment size

catholic school sector: public school or catholic school

pracad proportion of students in the academic track

disclim scale measuring disciplinary climate

himinty proportion of minority enrollment

meanses mean SES for each school

Details

Note: This dataset was imported from an SPSS .sav file using haven and therefore has variable attributes attached.

Source

<https://stats.oarc.ucla.edu/other/hlm/hlm-mlm/introduction-to-multilevel-modeling-using-hlm/>

References

Raudenbush SW, Bryk AS (2002). *Hierarchical Linear Models: Applications and Data Analysis Methods*. SAGE. ISBN 9780761919049.

---

icc   *Intraclass Correlation (ICC)*

Description

The icc function calculates the intraclass correlation (ICC) for multilevel models. The ICC represents the proportion of group-level variance to total variance. The ICC can be calculated for two or more levels in random-intercept models (Hox et al, 2018).

Note: For models with random slopes, it is generally advised to interpret with caution. According to Kreft and De Leeuw (1998, p. 63), "The concept of intra-class correlation is based on a model with a random intercept only. No unique intra-class correlation can be calculated when a random slope is present in the model." However, Snijders and Bosker (2012) offer a calculation to derive this value (equation 7.9). This equation is implemented here.

The `icc` function calculates the intraclass correlation for linear mixed-effects models estimated with the `lme4::lmer` function or generalized linear mixed-effect model estimated with the `lme4::glmer` function with `family = binomial(link="logit")`. For logistic models, the estimation method follows Hox et al. (2018, p. 107) recommendation of setting the level-1 residual variance to $\frac{\pi^2}{3}$. For a discussion different methods for estimating the intraclass correlation for binary responses, see Wu et al. (2012).

Usage

```
icc(model)
```

Arguments

model   A model produced using the `lme4::lmer()` or `lme4::glmer()` functions. This is an object of class `merMod` and subclass `lmerMod` or `glmerMod`.

Value

A data frame with random effects and their intraclass correlations.

References

Hox JJ, Moerbeek M, van de Schoot R (2018). *Multilevel Analysis: Techniques and Applications*. Taylor and Francis. ISBN 9781138121362.

Kreft, Ita, de Leeuw, Jan (1998). *Introducing Multilevel Modeling*. Sage Publications. ISBN 0761951405.

Snijders TAB, Bosker RJ (2012). *Multilevel Analysis*. SAGE. ISBN 9781849202015.

Wu S, Crespi CM, Wong WK (2012). "Comparison of Methods for Estimating the Intraclass Correlation Coefficient for Binary Responses in Cancer Prevention Cluster Randomized Trials." *Contemporary Clinical Trials*, **33**(5), 869–880. ISSN 1559-2030, doi:10.1016/j.cct.2012.05.004.

Examples

```
fit <- lme4::lmer(mathach ~ 1 + ses + catholic + (1|id),
data=hsb, REML=TRUE)

icc(fit)

# Logistic Example
  # Create binary outcome
hsb$binary_math <- ifelse(hsb$mathach <= 13, 0, 1)

fitb <- lme4::glmer(binary_math ~ 1 + ses + catholic + (1|id),
data=hsb, family = binomial(link="logit"))

icc(fitb)
```

---

ncv_tests   *Non-constant Variance Tests at Level-1 (experimental)*

Description

Computes three different Non-constant variance tests: the H test as discussed in Raudenbush and Bryk (2002, pp. 263-265) and Snijders and Bosker (2012, p. 159-160), an approximate Levene's test discussed by Hox et al. (2018, p. 238), and a variation of the Breusch-Pagan test.

For the H test, the user must specify the level-1 formula. This test computes a standardized measure of dispersion for each level-2 group and detects heteroscedasticity in the form of between-group differences in the level-one residuals variances. The standardized measure of dispersion is based on estimated ordinary least squares residuals in each group.

The Levene's test computes a oneway analysis of variance of the level-2 grouping variable on the squared residuals of the model. This test examines whether the variance of the residuals is the same in all groups.

The Breusch-Pagan test regresses the squared residuals on the fitted model. A likelihood ratio test is used to compare this model with a with a null model that regresses the squared residuals on an empty model with the same random effects. This test examines whether the variance of the residuals depends on the predictor variables.

Usage

```
ncv_tests(model, formula = NULL, verbose = FALSE)
```

Arguments

model
: a mixed model produced using the `lme4` package and the `lmer()` function. This is an object of class `merMod` and subclass `lmerMod`. Currently, only supports 2-level models.

formula
: level-1 formula to compute H test. Formula should be of the form $y \sim x_1 + ... + x_n \mid g$ where y is the response, $x_1 + ... + x_n$ are the covariates, and g is the grouping factor, see `lme4::lmList` for details.

verbose
: return additional statistics including d-values and outliers from H test; adjusted R-squared, ANOVA results, and mean residual by cluster for Levene test; and likelihood ratio test for B-P test.

Value

A list containing results from the three non-constant variance tests.

References

Hox JJ, Moerbeek M, van de Schoot R (2018). *Multilevel Analysis: Techniques and Applications*. Taylor and Francis. ISBN 9781138121362.

Raudenbush SW, Bryk AS (2002). *Hierarchical Linear Models: Applications and Data Analysis Methods*. SAGE. ISBN 9780761919049.

Singer JD, Willett JB (2003). *Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence*. Oxford University Press. ISBN 978-0-19-515296-8.

Examples

```
fit <- lme4::lmer(mathach ~ 1 + ses + catholic + (1|id), data=hsb, REML=FALSE)

ncv_tests(fit)

# extract outliers from H test
test <- ncv_tests(fit, formula = mathach ~ 1 + ses | id, verbose = TRUE)
test$H_test$outliers
```

---

plausible_values   *Plausible Values Range / Random Effect Confidence Intervals*

Description

The plausible values range is useful for gauging the magnitude of variation around fixed effects. For more information, see Raudenbush and Bryk (2002, p. 71) and Hoffman (2015, p. 166).

Usage

```
plausible_values(x, pct = 95)
```

Arguments

x   model produced using the `lme4::lmer()` function. This is an object of class `merMod` and subclass `lmerMod`.

pct   Percentile for the plausible value range, similar to a confidence interval. Must be specified as a whole number between 1 and 100 (e.g., 99, 95, 80). The 95% value range is used by default.

Value

A data frame specifying lower and upper bounds for each fixed effect.

References

Hoffman L (2015). *Longitudinal Analysis: Modeling within-Person Fluctuation and Change*. Routledge. ISBN 978-0415876025.

Raudenbush SW, Bryk AS (2002). *Hierarchical Linear Models: Applications and Data Analysis Methods*. SAGE. ISBN 9780761919049.

Examples

```
fit <- lme4::lmer(mathach ~ 1 + ses + catholic + (1|id),
data=hsb, REML=TRUE)

plausible_values(fit) #default is 95% range
plausible_values(fit, 99)
```

---

r2_cor	*Pseudo R-squared: Squared correlation between predicted and observed values*

Description

The r2_cor function estimates a pseudo R-squared by correlating predicted $\hat{Y}$ values and observed Y values. This pseudo R-squared is similar to the R^2 used in OLS regression. It indicates amount of variation in the outcome that is explained by the model (Peugh, 2010; Singer & Willett, 2003, p. 36).

Usage

```
r2_cor(x, verbose = FALSE)
```

Arguments

x	A model produced using the lme4::lmer() function. This is an object of class merMod and subclass lmerMod.
verbose	If true, prints an explanatory message, "The squared correlation between predicted and observed values is...". If false (default), returns a value.

Value

If verbose == TRUE, a console message. If verbose == FALSE (default), a numeric value.

References

Peugh JL (2010). "A Practical Guide to Multilevel Modeling." *Journal of School Psychology*, **48**(1), 85–112. ISSN 00224405, doi:10.1016/j.jsp.2009.09.002.

Singer JD, Willett JB (2003). *Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence*. Oxford University Press. ISBN 978-0-19-515296-8.

Examples

```
fit <- lme4::lmer(mathach ~ 1 + ses + catholic + (1|id),
data=hsb, REML=TRUE)

# returns a numeric value
r2_cor(fit)

# returns a console message with the r2 value
r2_cor(fit, verbose = TRUE)
```

---

r2_pve   *Proportion of variance explained*

Description

r2_pve calculates the proportional reduction in variance explained (PVE) by adding variables to a prior, nested model. The PVE is considered a local effect size estimate (Peugh, 2010; Raudenbush & Bryk, 2002).

Usage

```
r2_pve(model1, model2 = NULL)
```

Arguments

model1	Previous model, produced using the lme4::lmer() function. Usually, this is the null or unconditional model.
model2	Current model, produced using the lme4::lmer() function.

Value

Data frame containing the proportion of variance explained at each level

References

Peugh JL (2010). "A Practical Guide to Multilevel Modeling." *Journal of School Psychology*, **48**(1), 85–112. ISSN 00224405, doi:10.1016/j.jsp.2009.09.002.

Raudenbush SW, Bryk AS (2002). *Hierarchical Linear Models: Applications and Data Analysis Methods*. SAGE. ISBN 9780761919049.

Examples

```
fit1 <- lme4::lmer(mathach ~ 1 + (1|id), data=hsb, REML=FALSE)
fit2 <- lme4::lmer(mathach ~ 1 + ses + (1|id), data=hsb, REML=FALSE)

r2_pve(fit1, fit2)
```

| reliability | *Calculate reliability coefficients for random effects* |

Description

This function computes reliability coefficients for random effects according to Raudenbush and Bryk (2002) and Snijders and Bosker (2012). The reliability coefficient is equal to the proportion of between group variance to total variance: $\frac{\tau^2}{\tau^2 + \frac{\sigma^2}{n_j}}$. The empirical Bayes estimator for the random effect is a weighted combination of the cluster mean and grand mean with the weight given by the reliability of the random effect. We refer to this as a reliability because in classical test theory the ratio of the true score variance, τ^2, relative to the observed score variance of the sample mean is a reliability. A reliability close to 1 puts more weight on the cluster mean while a reliability close to 0 put more weight on the grand mean.

Usage

```
reliability(model)
```

Arguments

model   A model produced using the lme4::lmer() or lme4::glmer() functions. This is an object of class merMod and subclass lmerMod or glmerMod.

Value

A list with reliability estimates for each random effect

References

Snijders TAB, Bosker RJ (2012). *Multilevel Analysis*. SAGE. ISBN 9781849202015.

Raudenbush SW, Bryk AS (2002). *Hierarchical Linear Models: Applications and Data Analysis Methods*. SAGE. ISBN 9780761919049.

Examples

```
# lmer model
fit <- lme4::lmer(mathach ~ 1 + ses + catholic + (1 + ses|id),
data=hsb, REML=TRUE)

reliability(fit)
```

---

robust_se | *Robust Standard Errors*

Description

Implements cluster-robust standard errors from the `clubSandwich` package. The `clubSandwich` package is required to use this function. See `mlmhelpr::boot_se` for an alternative.

Usage

```
robust_se(model, type = "CR2", pct = 95)
```

Arguments

model
: model produced using the `lme4::lmer()` function. This is an object of class `merMod` and subclass `lmerMod`.

type
: character string specifying the estimation type. Options include "CR0", "CR1", "CR1p", "CR1S", "CR2", or "CR3". Defaults to "CR2". See details in `clubSandwich::vcovCR`.

pct
: percentage level for confidence interval. Defaults to 95. Must be specified as a whole number between 1 and 100 (e.g., 99, 95, 80).

Value

Data frame and message indicating type of robust standard error requested.

References

Pustejovsky J (2022). *clubSandwich: Cluster-Robust (Sandwich) Variance Estimators with Small-Sample Corrections*. R package version 0.5.8, <https://CRAN.R-project.org/package=clubSandwich>.

Examples

```
# run time > 5s
fit <- lme4::lmer(mathach ~ 1 + ses + catholic + (1|id),
data=hsb, REML=TRUE)

robust_se(fit)
```

---

taucov   *Tau Covariance*

Description

Quickly get the covariance and correlation between intercepts and slopes. By default, `lme4` only displays the correlation.

Usage

```
taucov(model)
```

Arguments

model   A model fit using the `lme4::lmer` function

Value

A data frame with the intercept, randomly-varying variables, covariance, and correlation.

Examples

```
fit <- lme4::lmer(mathach ~ 1 + ses + (1 + ses|id), data=hsb, REML=TRUE)

taucov(fit)
```

Index