

Package ‘HLMdiag’

August 29, 2016

Type Package

Title Diagnostic Tools for Hierarchical (Multilevel) Linear Models

Version 0.3.1

Date 2015-12-7

Author Adam Loy <loyad01@gmail.com>

Maintainer Adam Loy <loyad01@gmail.com>

Description A suite of diagnostic tools for hierarchical (multilevel) linear models. The tools include not only leverage and traditional deletion diagnostics (Cook's distance, covratio, covtrace, and MDFFITS) but also convenience functions and graphics for residual analysis. Models can be fit using either lmer in the 'lme4' package or lme in the 'nlme' package, but only two-level models fit using lme are currently supported.

Depends R (>= 2.15.0)

Imports ggplot2 (>= 0.9.2), stats, methods, plyr, reshape2, MASS, Matrix, RLRsim, mgcv

LinkingTo Rcpp, RcppArmadillo

Suggests mlmRev, WWGbook, lme4 (>= 1.0), nlme

License GPL-2

LazyLoad yes

LazyData yes

Collate 'diagnostic_functions.R' 'group_level_residual_functions.R' 'identification.R' 'plot_functions.R' 'quantile_functions.R' 'adjust_formula_lmList.R' 'case_delete.R' 'LSresids.R' 'HLMresid.R' 'help.R' 'influence_functions.R' 'utility_functions.R' 'rotate_ranefs.R' 'autism.R' 'ahd.R' 'radon.R'

URL <https://github.com/aloj/HLMdiag>

BugReports <https://github.com/aloj/HLMdiag/issues>

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-12-12 09:41:00

R topics documented:

adjust_lmList.formula	2
ahd	3
autism	4
case_delete.default	5
compare_eb_ls	7
covratio.default	7
diagnostics	10
dotplot_diag	11
ggplot_qqnorm	13
group_qqnorm	13
HLMdiag	14
HLMresid.default	15
leverage.default	17
LSresids.default	19
mdffits.default	20
radon	22
rotate_ranef.default	23
rvc.default	24
varcomp.mer	25
wages	26
Index	28

adjust_lmList.formula *Fitting Common Models via lm*

Description

Separate linear models are fit via `lm` similar to `lmList`, however, `adjust_lmList` can handle models where a factor takes only one level within a group. In this case, the `formula` is updated eliminating the offending factors from the formula for that group as the effect is absorbed into the intercept.

Usage

```
## S3 method for class 'formula'
adjust_lmList(object, data, pool)
```

Arguments

<code>object</code>	a linear formula such as that used by <code>lmList</code> , e.g. $y \sim x_1 + \dots + x_n \mid g$, where g is a grouping factor.
<code>data</code>	a data frame containing the variables in the model.
<code>pool</code>	a logical value that indicates whether the pooled standard deviation/error should be used.

References

Douglas Bates, Martin Maechler and Ben Bolker (2012). *lme4: Linear mixed-effects models using Eigen and Eigenpack*. R package version 0.999999-0.

See Also

[lmList](#), [lm](#)

Examples

```
data(Exam, package = 'm1mRev')
sepLM <- adjust_lmList(normexam ~ standLRT + sex + schgend | school, data = Exam)
confint(sepLM)
```

ahd

Methylprednisolone data

Description

Data from a longitudinal study examining the effectiveness of Methylprednisolone as a treatment for patients with severe alcoholic hepatitis. Subjects were randomly assigned to a treatment (31 received a placebo, 35 received the treatment) and serum bilirubin was measured each week for four weeks.

Usage

```
data(ahd)
```

Format

A data frame with 330 observations on the following 5 variables:

treatment The treatment a subject received - a factor. Levels are placebo and treated.

subject Subject ID - a factor.

week Week of the study (0–4) - the time variable.

sbvalue Serum bilirubin level (in $\mu\text{mol/L}$).

baseline A subject's serum bilirubin level at week 0.

Source

Vonesh, E. F. and Chinchilli, V. M. (1997) *Linear and Nonlinear Models for the Analysis of Repeated Measurements*. Marcel Dekker, New York.

References

Carithers, R. L., Herlong, H. F., Diehl, A. M., Shaw, E. W., Combes, B., Fallon, H. J. & Maddrey, W. C. (1989) Methylprednisolone therapy in patients with severe alcoholic hepatitis. *Annals of Internal Medicine*, **110**(9), 685–690.

autism

Autism data

Description

Data from a prospective longitudinal study following 214 children between the ages of 2 and 13 who were diagnosed with either autism spectrum disorder or non-spectrum developmental delays at age 2.

Usage

```
data(autism)
```

Format

A data frame with 604 observation on the following 7 variables:

childid Child ID.

sicdegp Sequenced Inventory of Communication Development group (an assessment of expressive language development) - a factor. Levels are low, med, and high.

age2 Age (in years) centered around age 2 (age at diagnosis).

vsae Vineland Socialization Age Equivalent

gender Child's gender - a factor. Levels are male and female.

race Child's race - a factor. Levels are white and nonwhite.

bestest2 Diagnosis at age 2 - a factor. Levels are autism and pdd (pervasive developmental disorder).

Source

<http://www-personal.umich.edu/~kwelch/>

References

Anderson, D. K., Lord, C., Risi, S., DiLavore, P. S., Shulman, C., Thurm, A., et al. (2007). Patterns of growth in verbal abilities among children with autism spectrum disorder. *Journal of Consulting and Clinical Psychology*, **75**(4), 594–604.

Anderson, D. K., Oti, R. S., Lord, C., & Welch, K. (2009). Patterns of Growth in Adaptive Social Abilities Among Children with Autism Spectrum Disorders. *Journal of Abnormal Child Psychology*, **37**(7), 1019–1034.

case_delete.default *Case Deletion for mer/lmerMod objects*

Description

This function is used to iteratively delete groups corresponding to the levels of a hierarchical linear model. It uses `lmer()` to fit the models for each deleted case (i.e. uses brute force). To investigate numerous levels of the model, the function will need to be called multiple times, specifying the group (level) of interest each time.

Usage

```
## Default S3 method:
case_delete(model, ...)

## S3 method for class 'mer'
case_delete(model, group = NULL, type = c("both", "fixef",
    "varcomp"), delete = NULL, ...)

## S3 method for class 'lmerMod'
case_delete(model, group = NULL, type = c("both", "fixef",
    "varcomp"), delete = NULL, ...)
```

Arguments

model	the original hierarchical model fit using <code>lmer()</code>
...	do not use
group	a variable used to define the group for which cases will be deleted. If this is left NULL (default), then the function will delete individual observations.
type	the part of the model for which you are obtaining deletion diagnostics: the fixed effects (<code>"fixef"</code>), variance components (<code>"varcomp"</code>), or <code>"both"</code> (default).
delete	index of individual cases to be deleted. For higher level units specified in this manner, the group parameter must also be specified. If <code>delete = NULL</code> then all cases are iteratively deleted.

Value

a list with the following components:

fixef.original the original fixed effects estimates

ranef.original the original predicted random effects

vcov.original the original variance-covariance matrix for the fixed effects

varcomp.original the original estimated variance components

fixef.delete a list of the fixed effects estimated after case deletion

ranef.delete a list of the random effects predicted after case deletion

vcov.delete a list of the variance-covariance matrices for the fixed effects obtained after case deletion

fitted.delete a list of the fitted values obtained after case deletion

varcomp.delete a list of the estimated variance components obtained after case deletion

Author(s)

Adam Loy <loyad01@gmail.com>

References

Christensen, R., Pearson, L.M., and Johnson, W. (1992) Case-Deletion Diagnostics for Mixed Models, *Technometrics*, **34**, 38 – 45.

Schabenberger, O. (2004) Mixed Model Influence Diagnostics, in *Proceedings of the Twenty-Ninth SAS Users Group International Conference*, SAS Users Group International.

Examples

```
library(lme4)
data(sleepstudy, package = 'lme4')
fm <- lmer(Reaction ~ Days + (Days|Subject), sleepstudy)

# Deleting every Subject
fmDel <- case_delete(model = fm, group = "Subject", type = "both")

# Deleting only subject 308
del308 <- case_delete(model = fm, group = "Subject", type = "both", delete = 308)

# Deleting a subset of subjects
delSubset <- case_delete(model = fm, group = "Subject", type = "both", delete = 308:310)
```

compare_eb_ls	<i>Visually comparing shrinkage and LS estimates</i>
---------------	--

Description

This function creates a plot (using `qplot()`) where the shrinkage estimate appears on the horizontal axis and the LS estimate appears on the vertical axis.

Usage

```
compare_eb_ls(eb, ols, identify = FALSE, silent = TRUE, ...)
```

Arguments

<code>eb</code>	a matrix of random effects
<code>ols</code>	a matrix of the OLS estimates found using <code>random_ls_coef</code>
<code>identify</code>	the percentage of points to identify as unusual, FALSE if you do not want the points identified.
<code>silent</code>	logical: should the list of data frames used to make the plots be suppressed.
<code>...</code>	other arguments to be passed to <code>qplot()</code>

Author(s)

Adam Loy <loyad01@gmail.com>

Examples

```
library(lme4)
wages.fm1 <- lmer(lnw ~ exper + (exper | id), data = wages)
wages.sepLM <- adjust_lmList(lnw ~ exper | id, data = wages)
rancoef.eb <- coef(wages.fm1)$id
rancoef.ols <- coef(wages.sepLM) # note diff. length than rancoef.eb
compare_eb_ls(eb = rancoef.eb[rownames(rancoef.ols),], ols = rancoef.ols, identify = 0.01)
```

covratio.default	<i>Influence on precision of fixed effects in HLMs</i>
------------------	--

Description

These functions calculate measures of the change in the covariance matrices for the fixed effects based on the deletion of an observation, or group of observations, for a hierarchical linear model fit using `lmer`.

Usage

```
## Default S3 method:
covratio(object, ...)

## Default S3 method:
covtrace(object, ...)

## S3 method for class 'mer'
covratio(object, group = NULL, delete = NULL, ...)

## S3 method for class 'lmerMod'
covratio(object, group = NULL, delete = NULL, ...)

## S3 method for class 'lme'
covratio(object, group = NULL, delete = NULL, ...)

## S3 method for class 'mer'
covtrace(object, group = NULL, delete = NULL, ...)

## S3 method for class 'lmerMod'
covtrace(object, group = NULL, delete = NULL, ...)

## S3 method for class 'lme'
covtrace(object, group = NULL, delete = NULL, ...)
```

Arguments

object	fitted object of class mer or lmerMod
...	do not use
group	variable used to define the group for which cases will be deleted. If group = NULL, then individual cases will be deleted.
delete	index of individual cases to be deleted. To delete specific observations the row number must be specified. To delete higher level units the group ID and group parameter must be specified. If delete = NULL then all cases are iteratively deleted.

Details

Both the covariance ratio (`covratio`) and the covariance trace (`covtrace`) measure the change in the covariance matrix of the fixed effects based on the deletion of a subset of observations. The key difference is how the variance covariance matrices are compared: `covratio` compares the ratio of the determinants while `covtrace` compares the trace of the ratio.

Value

If `delete = NULL` then a vector corresponding to each deleted observation/group is returned.
 If `delete` is specified then a single value is returned corresponding to the deleted subset specified.

Author(s)

Adam Loy <loyad01@gmail.com>

References

- Christensen, R., Pearson, L., & Johnson, W. (1992) Case-deletion diagnostics for mixed models. *Technometrics*, **34**(1), 38–45.
- Schabenberger, O. (2004) Mixed Model Influence Diagnostics, in *Proceedings of the Twenty-Ninth SAS Users Group International Conference*, SAS Users Group International.

See Also

[leverage.mer](#), [cooks.distance.mer](#), [mdffits.mer](#), [rvc.mer](#)

Examples

```
library(lme4)
ss <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)

# covratio for individual observations
ss.cr1 <- covratio(ss)

# covratio for subject-level deletion
ss.cr2 <- covratio(ss, group = "Subject")

## Not run:
## A larger example
data(Exam, package = 'mlmRev')
fm <- lmer(normexam ~ standLRT * schavg + (standLRT | school), Exam)

# covratio for individual observations
cr1 <- covratio(fm)

# covratio for school-level deletion
cr2 <- covratio(fm, group = "school")

## End(Not run)
# covtrace for individual observations
ss.ct1 <- covtrace(ss)

# covtrace for subject-level deletion
ss.ct2 <- covtrace(ss, group = "Subject")

## Not run:
## Returning to the larger example
# covtrace for individual observations
ct1 <- covtrace(fm)

# covtrace for school-level deletion
ct2 <- covtrace(fm, group = "school")
```

```
## End(Not run)
```

```
diagnostics
```

```
Calculating influence diagnostics for HLMs.
```

Description

This group of functions is used to compute deletion diagnostics for a hierarchical linear model based on the building blocks returned by `case_delete`.

Usage

```
diagnostics(object)

## S3 method for class 'case_delete'
cooks.distance(model, ...)

## S3 method for class 'case_delete'
mdffits(object, ...)

## S3 method for class 'case_delete'
covtrace(object, ...)

## S3 method for class 'case_delete'
covratio(object, ...)

## S3 method for class 'case_delete'
rvc(object, ...)
```

Arguments

<code>object</code>	an object containing the output returned by <code>case_delete()</code>
<code>model</code>	an object containing the output returned by <code>case_delete()</code> . This is only named differently to agree with the generic.
<code>...</code>	do not use

Details

The primary function is `diagnostics` which returns either a list or data frame of influence measures depending on whether `type = "both"` (list) or if only one aspect of the model is selected (data.frame). If `type = "both"`, then a list with Cook's distance, MDFFITS, COVTRACE, and COVRATIO are returned for the fixed effects and relative variance change (RVC) is returned for the variance components.

The methods `cooks.distance`, `mdffits`, `covtrace`, `covratio`, and `rvc` can be used for direct computation of the corresponding diagnostic quantities from an object of class `case_delete`.

Note

The results provided by this function will give exact values of the diagnostics; however, these are computationally very slow. Approximate versions of `cooks.distance`, `mdffits`, `covtrace`, `covratio` are implemented in `HLMdiag`, and can be called directly on the `mer` object.

Author(s)

Adam Loy <loyad01@gmail.com>

References

- Christensen, R., Pearson, L.M., and Johnson, W. (1992) "Case-Deletion Diagnostics for Mixed Models, *Technometrics*, **34**, 38 – 45.
- Dillane, D. (2005), Deletion Diagnostics for the Linear Mixed Model," Ph.D. thesis, Trinity College Dublin.
- Schabenberger, O. (2004) Mixed Model Influence Diagnostics, in *Proceedings of the Twenty-Ninth SAS Users Group International Conference*, SAS Users Group International.

See Also

[leverage.mer](#), [cooks.distance.mer](#), [mdffits.mer](#), [covratio.mer](#), [covtrace.mer](#)

Examples

```
library(lme4)
fm <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)

# Subject level deletion and diagnostics
subject.del <- case_delete(model = fm, group = "Subject", type = "both")
subject.diag <- diagnostics(subject.del)
```

dotplot_diag

Dot plots for influence diagnostics

Description

This is a function that can be used to create (modified) dotplots for the diagnostic measures. The plot allows the user to understand the distribution of the diagnostic measure and visually identify unusual cases.

Usage

```
dotplot_diag(x, index, data, cutoff, name = c("cooks.distance", "mdffits",
      "covratio", "covtrace", "rvc", "leverage"), modify = FALSE, ...)
```

Arguments

x	values of the diagnostic of interest
index	index (IDs) of x values
data	data frame to use (optional)
cutoff	value(s) specifying the boundary for unusual values of the diagnostic. The cutoff(s) can either be supplied by the user, or automatically calculated using measures of internal scaling if cutoff = "internal"
name	what diagnostic is being plotted (one of "cooks.distance", "mdffits", "covratio", "covtrace", "rvc", or "leverage"). this is used for the calculation of "internal" cutoffs
modify	specifies the geom to be used to produce a space-saving modification: either "dotplot" or "boxplot"
...	other arguments to be passed to <code>qplot()</code>

Note

The resulting plot uses `coord_flip` to rotate the plot, so when adding customized axis labels you will need to flip the usage of `xlab` and `ylab`.

Author(s)

Adam Loy <loyad01@gmail.com>

Examples

```
library(lme4)
fm <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)

# Subject level deletion and diagnostics
subject.del <- case_delete(model = fm, group = "Subject", type = "both")
subject.diag <- diagnostics(subject.del)

dotplot_diag(x = COOKSD, index = IDS, data = subject.diag[["fixef_diag"]],
             name = "cooks.distance", modify = FALSE,
             xlab = "Subject", ylab = "Cook's Distance")

dotplot_diag(x = sigma2, index = IDS, data = subject.diag[["varcomp_diag"]],
             name = "rvc", modify = "dotplot", cutoff = "internal",
             xlab = "Subject", ylab = "Relative Variance Change")

dotplot_diag(x = sigma2, index = IDS, data = subject.diag[["varcomp_diag"]],
             name = "rvc", modify = "boxplot", cutoff = "internal",
             xlab = "Subject", ylab = "Relative Variance Change")
```

ggplot_qqnorm	<i>Constructing a normal Q-Q plot</i>
---------------	---------------------------------------

Description

This function will construct a normal Q-Q plot within the ggplot2 framework. It combines the functionality of qqnorm and qqline.

Usage

```
ggplot_qqnorm(x, line = NULL, ...)
```

Arguments

x	a numeric vector
line	the method used to fit a reference line. If no reference line is desired, leave the value as NULL. line = "r1m" will use robust regression to fit a reference line. line = "quantile" will fit a line through the first and third quartiles. These options are the same as those given for the qqPlot function in the car package.
...	other arguments to be passed to qplot()

Author(s)

Adam Loy <loyad01@gmail.com>

See Also

[qqnorm](#), [qqline](#)

group_qqnorm	<i>Overlaying normal Q-Q plots</i>
--------------	------------------------------------

Description

This function will overlay multiple normal Q-Q plots on the same plot. This will be particularly useful when comparing the distribution between groups. In this situation, significantly different slopes would indicate the normal distributions for the groups do not share a common standard deviation.

Usage

```
group_qqnorm(x, group, line = NULL, alpha_point = 1, alpha_line = 1, ...)
```

Arguments

<code>x</code>	a numeric vector from which quantiles will be calculated
<code>group</code>	a vector indicating group membership for each value in <code>x</code> .
<code>line</code>	the method used to fit reference lines. If no reference lines are desired, leave the value as <code>NULL</code> . <code>line = "rlm"</code> will use robust regression to fit reference lines. <code>line = "quantile"</code> will fit lines through the first and third quartiles.
<code>alpha_point</code>	alpha value specified for the points
<code>alpha_line</code>	alpha value specified for the lines
<code>...</code>	other arguments to be passed to <code>ggplot</code>

Author(s)

Adam Loy <loyad01@gmail.com>

References

Hilden-Minton, J. A. (1995) Multilevel Diagnostics for Mixed and Hierarchical Linear Models, Ph.D. thesis, University of California Los Angeles.

HLMdiag

Diagnostic tools for hierarchical (multilevel) linear models

Description

HLMdiag provides a suite of diagnostic tools for hierarchical (multilevel) linear models fit using [lmer](#) or [lme](#). These tools are grouped below by purpose. See the help documentation for additional information about each function.

Details**Residual analysis**

HLMdiag's [HLMresid](#) function provides a convenient wrapper to obtain residuals at each level of a hierarchical linear model. In addition to being a wrapper function for functions implemented in the `lme4` package, [HLMresid](#) provides access to the marginal and least squares residuals (through [LSresids](#)) that were not previously implemented.

Influence analysis

Influence on fitted values

HLMdiag provides [leverage](#) that calculates the influence that observations/groups have on the fitted values (leverage). For mixed/hierarchical models leverage can be decomposed into two parts: the fixed part and the random part. We refer the user to the references cited in the help documentation for additional explanation.

Influence on fixed effects estimates

HLMdiag provides [cooks.distance](#) and [mdffits](#) to assess the influence of subsets of observations on the fixed effects.

Influence on precision of fixed effects

HLMdiag provides `covratio` and `covtrace` to assess the influence of subsets of observations on the precision of the fixed effects.

Influence on variance components

HLMdiag's `rvc` calculates the relative variance change to assess the influence of subsets of observations on the variance components.

Graphics

HLMdiag also strives to make graphical assessment easier in the `ggplot2` framework by providing dotplots for influence diagnostics (`dotplot_diag`), grouped Q-Q plots (`group_qqnorm`), and Q-Q plots that combine the functionality of `qqnorm` and `qqline` (`ggplot_qqnorm`).

HLMresid.default *Calculating residuals from HLMs*

Description

HLMresid is a function that extracts residuals from a hierarchical linear model fit using `lmer`. That is, it is a unified framework that extracts/calculates residuals from `mer` or `lmerMod` objects.

Usage

```
## Default S3 method:
HLMresid(object, ...)

## S3 method for class 'mer'
HLMresid(object, level, type = "EB", sim = NULL,
          standardize = FALSE, ...)

## S3 method for class 'lmerMod'
HLMresid(object, level, type = "EB", sim = NULL,
          standardize = FALSE, ...)
```

Arguments

<code>object</code>	an object of class <code>mer</code> or <code>lmerMod</code> .
<code>...</code>	do not use
<code>level</code>	which residuals should be extracted: 1 for within-group (case-level) residuals, the name of a grouping factor (as defined in <code>flist</code> of the <code>mer</code> object) for between-group residuals, or <code>marginal</code> .
<code>type</code>	how are the residuals predicted: either <code>"EB"</code> or <code>"LS"</code> (the default is <code>"EB"</code>).
<code>sim</code>	optional argument giving the data frame used for <code>LS</code> residuals. This is used mainly for dealing with simulations.
<code>standardize</code>	if <code>standardize = TRUE</code> the standardized residuals will be returned; if <code>standardize = "semi"</code> then the semi-standardized level-1 residuals will be returned. Note that for higher-level residuals of <code>type = "LS"</code> , <code>standardize = TRUE</code> does not result in standardized residuals as they have not been implemented.

Details

This function extracts residuals from the model, and can extract residuals estimated using least squares (LS) or Empirical Bayes (EB). This unified framework enables the analyst to more easily conduct an upward residual analysis during model exploration/checking.

The HLMresid function provides a wrapper that will extract residuals from a fitted mer or lmerMod object. The function provides access to residual quantities already made available by the functions resid and ranef, but adds additional functionality. Below is a list of types of residuals that can be extracted.

raw level-1 residuals These are equivalent to the residuals extracted by resid if level = 1, type = "EB", and standardize = FALSE is specified. You can also specify type = "LS" for LS residuals that are not equivalent to those from resid.

standardized level-1 residuals Specify level = 1, and standardize = TRUE. This works with both type = "EB" or "LS".

semi-standardized level-1 residuals Specify level = 1, type = "LS" and standardize = "semi".

raw group level residuals These are equivalent to extracting the predicted random effects for a given group using ranef. Set level to a grouping factor name and type = "EB". type = "LS" can also be specified, though this is less common.

standardized group level residuals Set level to a grouping factor name, type = "EB", and standardized = TRUE. This will not produce standardized residuals for type = "LS".

marginal residuals The marginal residuals can be obtained by setting level = "marginal". Only type = "EB" is implemented.

cholesky residuals These are essentially standardized marginal residuals. To obtain cholesky residuals set level = "marginal", type = "EB", and standardize = TRUE.

Note that standardize = "semi" is only implemented for level-1 LS residuals.

Author(s)

Adam Loy <loyad01@gmail.com>

References

Hilden-Minton, J. (1995) Multilevel diagnostics for mixed and hierarchical linear models. University of California Los Angeles.

Houseman, E. A., Ryan, L. M., & Coull, B. A. (2004) Cholesky Residuals for Assessing Normal Errors in a Linear Model With Correlated Outcomes. *Journal of the American Statistical Association*, **99**(466), 383–394.

See Also

[LSresids](#), [resid](#), [ranef](#)

Examples

```

library(lme4)
fm1 <- lmer(Reaction ~ Days + (Days|Subject), sleepstudy)

# level-1 residuals
all.equal(HLMresid(object = fm1, level = 1, type = "EB"), resid(fm1)) ## EB
r1LS <- HLMresid(object = fm1, level = 1, type = "LS") ## raw LS resid
head(r1LS)
r1LS.std <- HLMresid(object = fm1, level = 1, type = "LS", standardize = TRUE) ## std. LS resid
head(r1LS.std)

# level-2 residuals
all.equal(r2EB <- HLMresid(object = fm1, level = "Subject", type = "EB"), ranef(fm1)[["Subject"]])
r2EB.std <- HLMresid(object = fm1, level = "Subject", type = "EB", standardize = TRUE)
head(r2EB)
head(r2EB.std)

# marginal residuals
mr <- HLMresid(object = fm1, level = "marginal")
cholr <- HLMresid(object = fm1, level = "marginal", standardize = TRUE) # Cholesky residuals

```

leverage.default	<i>Leverage for HLMs</i>
------------------	--------------------------

Description

This function calculates the leverage of a hierarchical linear model fit by lmer.

Usage

```

## Default S3 method:
leverage(object, ...)

## S3 method for class 'mer'
leverage(object, level, ...)

## S3 method for class 'lmerMod'
leverage(object, level, ...)

## S3 method for class 'lme'
leverage(object, level, ...)

```

Arguments

object	fitted object of class mer of lmerMod
...	do not use

level the level at which the leverage should be calculated: either 1 for observation level leverage or the name of the grouping factor (as defined in `flist` of the `mer` object) for group level leverage. Leverage assumes that the grouping factors are unique; thus, if IDs are repeated within each unit, unique IDs must be generated by the user prior to use of leverage.

Details

Demidenko and Stukel (2005) describe leverage for mixed (hierarchical) linear models as being the sum of two components, a leverage associated with the fixed (H_1) and a leverage associated with the random effects (H_2) where

$$H_1 = X(X'V^{-1}X)^{-1}X'V^{-1}$$

and

$$H_2 = ZDZ'V^{-1}(I - H_1)$$

Nobre and Singer (2011) propose using

$$H_2^* = ZDZ'$$

as the random effects leverage as it does not rely on the fixed effects.

For individual observations leverage uses the diagonal elements of the above matrices as the measure of leverage. For higher-level units, leverage uses the mean trace of the above matrices associated with each higher-level unit.

Value

leverage returns a data frame with the following columns:

`overall` The overall leverage, i.e. $H = H_1 + H_2$.

`fixef` The leverage corresponding to the fixed effects.

`ranef` The leverage corresponding to the random effects proposed by Demidenko and Stukel (2005).

`ranef.uc` The (unconfounded) leverage corresponding to the random effects proposed by Nobre and Singer (2011).

Author(s)

Adam Loy <loyad01@gmail.com>

References

Demidenko, E., & Stukel, T. A. (2005) Influence analysis for linear mixed-effects models. *Statistics in Medicine*, **24**(6), 893–909.

Nobre, J. S., & Singer, J. M. (2011) Leverage analysis for linear mixed models. *Journal of Applied Statistics*, **38**(5), 1063–1072.

See Also

[cooks.distance.mer](#), [mdffits.mer](#), [covratio.mer](#), [covtrace.mer](#), [rvc.mer](#)

Examples

```
library(lme4)
fm <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)

# Observation level leverage
lev1 <- leverage(fm, level = 1)
head(lev1)

# Group level leverage
lev2 <- leverage(fm, level = "Subject")
head(lev2)
```

LSresids.default *Calculating least squares residuals*

Description

This function calculates least squares (LS) residuals found by fitting separate LS regression models to each case. For examples see the documentation for HLMresid.

Usage

```
## Default S3 method:
LSresids(object, ...)

## S3 method for class 'mer'
LSresids(object, level, sim = NULL, standardize = FALSE, ...)

## S3 method for class 'lmerMod'
LSresids(object, level, sim = NULL, standardize = FALSE,
         ...)
```

Arguments

object	an object of class mer or lmerMod.
...	do not use
level	which residuals should be extracted: 1 for case-level residuals or the name of a grouping factor (as defined in flist of the mer object) for between-group residuals.
sim	optional argument giving the data frame used for LS residuals. This is used mainly when dealing with simulations.
standardize	if TRUE the standardized level-1 residuals will also be returned (if level = 1); if "semi" then the semi-standardized level-1 residuals will be returned.

Author(s)

Adam Loy <loyad01@gmail.com>

References

Hilden-Minton, J. (1995) Multilevel diagnostics for mixed and hierarchical linear models. University of California Los Angeles.

See Also

[HLMresid](#)

mdffits.default	<i>Influence on fixed effects of HLMs</i>
-----------------	---

Description

These functions calculate measures of the change in the fixed effects estimates based on the deletion of an observation, or group of observations, for a hierarchical linear model fit using `lmer`.

Usage

```
## Default S3 method:
mdffits(object, ...)

## S3 method for class 'mer'
cooks.distance(model, group = NULL, delete = NULL, ...)

## S3 method for class 'lmerMod'
cooks.distance(model, group = NULL, delete = NULL, ...)

## S3 method for class 'lme'
cooks.distance(model, group = NULL, delete = NULL, ...)

## S3 method for class 'mer'
mdffits(object, group = NULL, delete = NULL, ...)

## S3 method for class 'lmerMod'
mdffits(object, group = NULL, delete = NULL, ...)

## S3 method for class 'lme'
mdffits(object, group = NULL, delete = NULL, ...)
```

Arguments

<code>object</code>	fitted object of class <code>mer</code> or <code>lmerMod</code>
<code>...</code>	do not use
<code>model</code>	fitted model of class <code>mer</code> or <code>lmerMod</code>
<code>group</code>	variable used to define the group for which cases will be deleted. If <code>group = NULL</code> , then individual cases will be deleted.

`delete` index of individual cases to be deleted. To delete specific observations the row number must be specified. To delete higher level units the group ID and group parameter must be specified. If `delete = NULL` then all cases are iteratively deleted.

Details

Both Cook's distance and MDFFITS measure the change in the fixed effects estimates based on the deletion of a subset of observations. The key difference between the two diagnostics is that Cook's distance uses the covariance matrix for the fixed effects from the original model while MDFFITS uses the covariance matrix from the deleted model.

Value

Both functions return a numeric vector (or single value if `delete` has been specified) with attribute `beta_cdd` giving the difference between the full and deleted parameter estimates.

Note

Because MDFFITS requires the calculation of the covariance matrix for the fixed effects for every model, it will be slower.

Author(s)

Adam Loy <loyad01@gmail.com>

References

Christensen, R., Pearson, L., & Johnson, W. (1992) Case-deletion diagnostics for mixed models. *Technometrics*, **34**, 38–45.

Schabenberger, O. (2004) Mixed Model Influence Diagnostics, in *Proceedings of the Twenty-Ninth SAS Users Group International Conference*, SAS Users Group International.

See Also

[leverage.mer](#), [covratio.mer](#), [covtrace.mer](#), [rvc.mer](#)

Examples

```
library(lme4)
data(sleepstudy, package = 'lme4')
ss <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)

# Cook's distance for individual observations
ss.cd.lev1 <- cooks.distance(ss)

# Cook's distance for each Subject
ss.cd.subject <- cooks.distance(ss, group = "Subject")

## Not run:
data(Exam, package = 'mlmRev')
```

```

fm <- lmer(normexam ~ standLRT * schavg + (standLRT | school), Exam)

# Cook's distance for individual observations
cd.lev1 <- cooks.distance(fm)

# Cook's distance for each school
cd.school <- cooks.distance(fm, group = "school")

# Cook's distance when school 1 is deleted
cd.school1 <- cooks.distance(fm, group = "school", delete = 1)

## End(Not run)
# MDFFITS for individual observations
ss.m1 <- mdffits(ss)

# MDFFITS for each Subject
ss.m.subject <- mdffits(ss, group = "Subject")

## Not run:

# MDFFITS for individual observations
m1 <- mdffits(fm)

# MDFFITS for each school
m.school <- mdffits(fm, group = "school")

## End(Not run)

```

radon

Radon data

Description

Radon measurements of 919 owner-occupied homes in 85 counties of Minnesota.

Usage

```
data(radon)
```

Format

A data frame with 919 observations on the following 5 variables:

log.radon Radon measurement (in log pCi/L, i.e., log picoCurie per liter)

basement Indicator for the level of the home at which the radon measurement was taken - 0 = basement, 1 = first floor.

uranium Average county-level soil uranium content.

county County ID.

county.name County name - a factor.

Source

<http://www.stat.columbia.edu/~gelman/arm/software/>

References

Price, P. N., Nero, A. V. and Gelman, A. (1996) Bayesian prediction of mean indoor radon concentrations for Minnesota counties. *Health Physics*. **71**(6), 922–936.

Gelman, A. and Hill, J. (2007) *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press.

rotate_ranef.default *Calculate s-dimensional rotated random effects*

Description

This function calculates reduced dimensional rotated random effects. The rotation reduces the influence of the residuals from other levels of the model so that distributional assessment of the resulting random effects is possible.

Usage

```
## Default S3 method:
rotate_ranef(.mod, ...)

## S3 method for class 'mer'
rotate_ranef(.mod, .L, s = NULL, .varimax = FALSE, ...)

## S3 method for class 'lmerMod'
rotate_ranef(.mod, .L, s = NULL, .varimax = FALSE, ...)

## S3 method for class 'lme'
rotate_ranef(.mod, .L, s = NULL, .varimax = FALSE, ...)
```

Arguments

.mod	an object of class mer or lmerMod.
...	do not use
.L	a matrix defining which combination of random effects are of interest.
s	the dimension of the subspace of interest.
.varimax	if .varimax = TRUE than the raw varimax rotation will be applied to the resulting rotation.

Author(s)

Adam Loy <loyad01@gmail.com>

References

Loy, A. & Hofmann, H. (in press). Are you Normal? The Problem of Confounded Residual Structures in Hierarchical Linear Models. *Journal of Computational and Graphical Statistics*.

Examples

```
library(lme4)
fm <- lmer(log.radon ~ basement + uranium + (basement | county), data = radon)

ngrps <- summary(fm)$ngrps
L.b0 <- kronecker(Diagonal(ngrps), c(1, 0))
L.b1 <- kronecker(Diagonal(ngrps), c(0, 1))

b0.rot <- rotate_ranef(.mod = fm, .L = L.b0, s = 65, .varimax = TRUE)
qqnorm(b0.rot)

b1.rot <- rotate_ranef(.mod = fm, .L = L.b1, s = 65, .varimax = TRUE)
qqnorm(b1.rot)
```

rvc.default

Relative variance change for HLMs

Description

This function calculates the relative variance change (RVC) of hierarchical linear models fit via lmer.

Usage

```
## Default S3 method:
rvc(object, ...)

## S3 method for class 'mer'
rvc(object, group = NULL, delete = NULL, ...)

## S3 method for class 'lmerMod'
rvc(object, group = NULL, delete = NULL, ...)

## S3 method for class 'lme'
rvc(object, group = NULL, delete = NULL, ...)
```

Arguments

object	fitted object of class mer or lmerMod
...	do not use
group	variable used to define the group for which cases will be deleted. If group = NULL, then individual cases will be deleted.

`delete` index of individual cases to be deleted. To delete specific observations the row number must be specified. To delete higher level units the group ID and group parameter must be specified. If `delete = NULL` then all cases are iteratively deleted.

Value

If `delete = NULL` a matrix with columns corresponding to the variance components of the model and rows corresponding to the deleted observation/group is returned.

If `delete` is specified then a named vector is returned.

The residual variance is named `sigma2` and the other variance components are named `D**` where the trailing digits give the position in the covariance matrix of the random effects.

Author(s)

Adam Loy <loyad01@gmail.com>

References

Dillane, D. (2005) Deletion Diagnostics for the Linear Mixed Model. Ph.D. thesis, Trinity College Dublin

See Also

[leverage.mer](#), [cooks.distance.mer](#), [mdffits.mer](#), [covratio.mer](#), [covtrace.mer](#)

varcomp.mer

Extracting variance components

Description

This function extracts the variance components from a mixed/hierarchical linear model fit using `lmer`.

Usage

```
varcomp.mer(object)
```

Arguments

`object` a fitted model object of class `mer` or `lmerMod`.

Value

A named vector is returned. `sigma2` denotes the residual variance. The other variance components are names `D**` where the trailing digits specify the of that variance component in the covariance matrix of the random effects.

Author(s)

Adam Loy <loyad01@gmail.com>

Examples

```
library(lme4)
fm1 <- lmer(Reaction ~ Days + (Days|Subject), sleepstudy)
varcomp.mer(fm1)
```

wages

Wages for male high school dropouts

Description

Data on the labor-market experience of male high school dropouts.

Format

A data frame with 6402 observations on the following 15 variables.

id respondent id - a factor with 888 levels.

lnw natural log of wages expressed in 1990 dollars.

exper years of experience in the work force

ged equals 1 if respondent has obtained a GED as of the time of survey, 0 otherwise

postexp labor force participation since obtaining a GED (in years) - before a GED is earned postexp = 0, and on the day a GED is earned postexp = 0

black factor - equals 1 if subject is black, 0 otherwise

hispanic factor - equals 1 if subject is hispanic, 0 otherwise

hgc highest grade completed - takes integers 6 through 12

hgc.9 hgc - 9, a centered version of hgc

uerate local area unemployment rate for that year

ue.7

ue.centert1

ue.mean

ue.person.cen

ue1

Source

These data are originally from the 1979 National Longitudinal Survey on Youth (NLSY79) that can be found here <http://www.bls.gov/nls/nlsdata.htm>.

Singer and Willett (2003) used these data for examples in chapter (insert info. here) and the data sets used can be found on the UCLA Statistical Computing website: <http://www.ats.ucla.edu/stat/examples/alda/>

Additionally the data were discussed by Cook and Swayne (2003) and the data can be found on the GGobi website: <http://www.ggobi.org/book/>.

References

Singer, J. D. and Willett, J. B. (2003), *Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence*, New York: Oxford University Press.

Cook, D. and Swayne, D. F. (2007), *Interactive and Dynamic Graphics for Data Analysis with R and GGobi*, Springer.

Examples

```
str(wages)
summary(wages)

## Not run:
library(lme4)
lmer(lnw ~ exper + (exper | id), data = wages)

## End(Not run)
```

Index

*Topic **datasets**

ahd, [3](#)
autism, [4](#)
radon, [22](#)
wages, [26](#)

*Topic **hplot**

dotplot_diag, [11](#)
ggplot_qqnorm, [13](#)
group_qqnorm, [13](#)

*Topic **models**

adjust_lmList.formula, [2](#)
case_delete.default, [5](#)
covratio.default, [7](#)
diagnostics, [10](#)
HLMresid.default, [15](#)
leverage.default, [17](#)
LSresids.default, [19](#)
mdffits.default, [20](#)
rvc.default, [24](#)
varcomp.mer, [25](#)

*Topic **package**

HLMdiag, [14](#)

*Topic **regression**

adjust_lmList.formula, [2](#)
case_delete.default, [5](#)
covratio.default, [7](#)
diagnostics, [10](#)
HLMresid.default, [15](#)
leverage.default, [17](#)
LSresids.default, [19](#)
mdffits.default, [20](#)
rvc.default, [24](#)
varcomp.mer, [25](#)

adjust_lmList(adjust_lmList.formula), [2](#)
adjust_lmList(formula,data.frame-method
(adjust_lmList.formula), [2](#)
adjust_lmList.formula, [2](#)
ahd, [3](#)
autism, [4](#)

case_delete(case_delete.default), [5](#)
case_delete.default, [5](#)
compare_eb_ls, [7](#)
cooks.distance, [14](#)
cooks.distance(mdffits.default), [20](#)
cooks.distance.case_delete
(diagnostics), [10](#)
cooks.distance.mer, [9, 11, 18, 25](#)
covratio, [15](#)
covratio(covratio.default), [7](#)
covratio.case_delete(diagnostics), [10](#)
covratio.default, [7](#)
covratio.mer, [11, 18, 21, 25](#)
covtrace, [15](#)
covtrace(covratio.default), [7](#)
covtrace.case_delete(diagnostics), [10](#)
covtrace.mer, [11, 18, 21, 25](#)

diagnostics, [10](#)
dotplot_diag, [11, 15](#)

ggplot_qqnorm, [13, 15](#)
group_qqnorm, [13, 15](#)

HLMdiag, [14](#)
HLMdiag-package(HLMdiag), [14](#)
HLMresid, [14, 20](#)
HLMresid(HLMresid.default), [15](#)
HLMresid.default, [15](#)

leverage, [14](#)
leverage(leverage.default), [17](#)
leverage.default, [17](#)
leverage.mer, [9, 11, 21, 25](#)
lm, [3](#)
lme, [14](#)
lmer, [14](#)
lmList, [3](#)
LSresids, [14, 16](#)
LSresids(LSresids.default), [19](#)

LSresids.default, 19

mdffits, 14
mdffits (mdffits.default), 20
mdffits.case_delete (diagnostics), 10
mdffits.default, 20
mdffits.mer, 9, 11, 18, 25

package-HLMdiag (HLMdiag), 14

qqline, 13, 15
qqnorm, 13, 15

radon, 22
ranef, 16
resid, 16
rotate_ranef (rotate_ranef.default), 23
rotate_ranef.default, 23
rvc, 15
rvc (rvc.default), 24
rvc.case_delete (diagnostics), 10
rvc.default, 24
rvc.mer, 9, 18, 21

varcomp.mer, 25

wages, 26