Package 'influence.ME'

October 13, 2022

Type Package

Title Tools for Detecting Influential Data in Mixed Effects Models

Version 0.9-9

Date 2017-06-07

Author Rense Nieuwenhuis, Ben Pelzer, Manfred te Grotenhuis

Maintainer Rense Nieuwenhuis <rense.nieuwenhuis@sofi.su.se>

Description Provides a collection of tools for
detecting influential cases in generalized mixed effects
models. It analyses models that were estimated using 'lme4'. The
basic rationale behind identifying influential data is that
when single units are omitted from the data, models
based on these data should not produce substantially different
estimates. To standardize the assessment of how influential a
(single group of) observation(s) is, several measures of
influence are common practice, such as Cook's Distance.
In addition, we provide a measure of percentage change of the fixed point
estimates and a simple procedure to detect changing levels of significance.

License GPL-3

URL http://www.rensenieuwenhuis.nl/r-project/influenceme/

Depends R(>= 2.15.0), lme4(>= 1.0)

Imports Matrix(>= 1.0), lattice

LazyLoad yes

LazyData yes

NeedsCompilation no

Repository CRAN

Date/Publication 2017-06-07 07:05:02 UTC

R topics documented:

1

---

influence.ME-package   *Influence.ME: Tools for detecting influential data in mixed effects models*

Description

influence.ME calculates measures of influence for mixed effects models estimated with lme4. The basic rationale behind measuring influential cases is that when iteratively single units are omitted from the data, models based on these data should not produce substantially different estimates. To standardize the assessment of how influential a (single group of) observation(s) is, several measures of influence are common practice. First, DFBETAS is a standardized measure of the absolute difference between the estimate with a particular case included and the estimate without that particular case. Second, Cook's distance provides an overall measurement of the change in all parameter estimates, or a selection thereof.

Details

Package:	influence.ME
Type:	Package
Version:	0.9.2
Date:	2013-01-15
License:	GPL-3
LazyLoad:	yes

Calculating measures of influential data on a mixed effects regression model entails the re-estimation of this model for each set of potentially influential data separately. The influence() function does this, and returns the altered estimates resulting from each re-estimation. These altered estimates can subsequently be entered to the cooks.distance and dfbetas methods, to calculate Cook's Distance and the DFBETAS (standardized difference of the beta) measures.

Author(s)

Rense Nieuwenhuis, Ben Pelzer, Manfred te Grotenhuis

Maintainer: Rense Nieuwenhuis <r.nieuwenhuis@utwente.nl>

References

Belsley, D.A., Kuh, E. & Welsch, R.E. (1980). *Regression Diagnostics. Identifying Influential Data and Source of Collinearity*. Wiley.

Snijders, T.A. & Bosker, R.J. (1999). *Multilevel Analysis, an introduction to basic and advanced multilevel modeling*. Sage.

Van der Meer, T., Te Grotenhuis, M., & Pelzer, B. (2010). *Influential Cases in Multilevel Modeling: A Methodological Comment*. American Sociological Review, 75(1), 173-178.

See Also

influence, cooks.distance.estex, dfbetas.estex, pchange, sigtest

Examples

```
## Not run:
data(school23)

model.a <- lmer(math ~ structure + SES  + (1 | school.ID), data=school23)
alt.est.a <- influence(model.a, "school.ID")

model.b <- exclude.influence(model.a, "school.ID", "7472")
alt.est.b <- influence(model.b, "school.ID")

cooks.distance(alt.est.b)

model.c <- exclude.influence(model.b, "school.ID", "54344")
alt.est.c <- influence(model.c, "school.ID")

cooks.distance(alt.est.c)

## End(Not run)
```

---

cooks.distance.estex   *Compute the Cook's distance measure of influential data on mixed effects models*

Description

Cook's Distance is a measure indicating to what extent model parameters are influenced by (a set of) influential data on which the model is based. This function computes the Cook's distance based on the information returned by the influence() function.

Usage

```
## S3 method for class 'estex'
cooks.distance(model, parameters=0, sort=FALSE, ...)
```

Arguments

model	An object as returned by the influence() function, containing the altered estimates of a mixed effects regression model
parameters	Used to define a selection of parameters. If parameters=0 (default), Cook's Distance is calculated based on all parameters in the model
sort	If sort=TRUE the values of Cook's Distance are ordered based on magnitude. If sort=FALSE (default) no sorting takes place.
...	Currently not used

Value

A one-column matrix is returned containing values for the Cook's Distance based on the selected (fixed) parameters of the model. Each row shows the Cook's Distance associated with each evaluated set of influential data (data nested within each evaluated level of the grouping factor).

Author(s)

Rense Nieuwenhuis, Ben Pelzer, Manfred te Grotenhuis

References

Nieuwenhuis, R., Te Grotenhuis, M., & Pelzer, B. (2012). Influence.ME: tools for detecting influential data in mixed effects models. *R Journal*, 4(2), 38???47.

Belsley, D.A., Kuh, E. & Welsch, R.E. (1980). *Regression Diagnostics. Identifying Influential Data and Source of Collinearity*. Wiley.

Snijders, T.A. & Bosker, R.J. (1999). *Multilevel Analysis, an introduction to basic and advanced multilevel modeling*. Sage.

Van der Meer, T., Te Grotenhuis, M., & Pelzer, B. (2010). *Influential Cases in Multilevel Modeling: A Methodological Comment*. American Sociological Review, 75(1), 173-178.

See Also

influence, dfbetas

Examples

```
## Not run:
data(school23)
model <- lmer(math ~ structure + SES  + (1 | school.ID), data=school23)

alt.est <- influence(model, group="school.ID")
cooks.distance(alt.est)

## End(Not run)
```

---

dfbetas.estex   *Compute the DFBETAS measure of influential data*

Description

DFBETAS (standardized difference of the beta) is a measure that standardizes the absolute difference in parameter estimates between a (mixed effects) regression model based on a full set of data, and a model from which a (potentially influential) subset of data is removed. A value for DFBETAS is calculated for each parameter in the model separately. This function computes the DFBETAS based on the information returned by the influence() function.

Usage

```
## S3 method for class 'estex'
dfbetas(model, parameters = 0, sort=FALSE, to.sort=NA, abs=FALSE, ...)
```

Arguments

model
: An object as returned by the influence() function, containing the altered estimates of a mixed effects regression model

parameters
: Used to define a selection of parameters. If parameters=0 (default), DFBETAS is calculated for all parameters in the model

sort
: If sort=TRUE the values of DFBETAS are ordered based on magnitude. If sort=FALSE (default) no sorting takes place.

to.sort
: Specify on which variable the DFBETAS must be sorted. If only one variable present (either in the model, or due to the selection specified in parameters), this parameter can be omitted. If DFBETAS is calculated for multiple variables, and sort=TRUE, specification of to.sort is required, or an error is returned.

abs
: If abs=TRUE, the absolute values of DFBETAS are returned, while if abs=FALSE (default), both positive and negative values are possible. If both abs=TRUE and sort=TRUE, the abs parameters precedes the sort parameter, and thus the absolute values of DFBETAS are sorted.

...
: Currently not used

Value

A matrix is returned, containing DFBETAS-values for each (selected) fixed parameter of the model, and separately for each evaluated set of influential data.

Author(s)

Rense Nieuwenhuis, Ben Pelzer, Manfred te Grotenhuis

References

Nieuwenhuis, R., Te Grotenhuis, M., & Pelzer, B. (2012). Influence.ME: tools for detecting influential data in mixed effects models. *R Journal*, 4(2), 38???47.

Belsley, D.A., Kuh, E. & Welsch, R.E. (1980). *Regression Diagnostics. Identifying Influential Data and Source of Collinearity*. Wiley.

Snijders, T.A. & Bosker, R.J. (1999). *Multilevel Analysis, an introduction to basic and advanced multilevel modeling*. Sage.

Van der Meer, T., Te Grotenhuis, M., & Pelzer, B. (2010). *Influential Cases in Multilevel Modeling: A Methodological Comment*. American Sociological Review, 75(1), 173-178.

See Also

[influence.mer](#), [cooks.distance.estex](#)

Examples

```
## Not run:
 data(school23)
 model <- lmer(math ~ structure + SES  + (1 | school.ID), data=school23)

 alt.est <- influence(model, group="school.ID")
 dfbetas(alt.est)

## End(Not run)
```

---

exclude.influence	*Exclude the influence of a grouped set of observations in mixed effects models.*

Description

Using mixed effects regression models, exclude.influence excludes the influence of a group of cases grouped within a single grouping factor, or a set of grouping factors. The function returns a model in which the influence a grouped set of observations has on both the variance and point-estimate of the (random) intercept.

Usage

```
exclude.influence(model, grouping=NULL, level=NULL, obs=NULL, gf="single", delete=TRUE)
```

Arguments

model	A mixed effects regression model
grouping	The grouping factor of which one or more groupings levels are to be 'neutralized'
level	Vector of character strings, indicating either a single level or a set of grouping levels the influence of which is to be neutralized

obs	Specifies which individual observation(s) (rather than groups) to be deleted from the data/
gf	Indicates from which of the model's grouping factors the influence of the specified grouping factor is to be neutralized. If gf="single" (default), the levels of the specified grouping factor are only neutralized from the grouping factor specified in group. In its present form, gf="single" only works on mixed models with a maximum of 2 grouping factors. If gf="all", the influence from the levels of group is neutralized regarding all grouping factors in the model. This option only applies to models with more than a single grouping factor.
delete	If delete=TRUE (default), the influence is excluded by simply deleting the observations nested within the higher level group. If delete=FALSE, the influence of higher level groups is excluded from the model by setting the intercept-vector for the observations nested within these groups to 0, and by adding a dummy-variable indicating these observations (Langford and Lewis, 1998). This latter option currently does not work with models that include factor variables.

Details

To apply the basic logic of influential cases to mixed effects models one has to measure the influence of a particular higher level unit on the estimates of a higher level predictor. This means that the mixed effects model has to be adjusted to neutralize the unit's influence on that estimate, while at the same time allowing the unit's lower-level cases to help estimate the effects of the lower-level predictors in the model. This procedure is based on a modification of the intercept and the addition of a dummy variable for the cases that might be influential.

The model that is returned by exclude.influence thus contains a modified intercept, and one or more additional dummy variables. To help identify this model as modified (which is required when in a later stage the influence of additional grouping levels is excluded), the intercept is renamed to 'intercept.alt'. The additional dummy variables, indicating the observations associated with the grouping factor levels of which the influence was neutralized, are labeled starting with 'estex.', combined with the label of the neutralized grouping level.

Value

Mixed effects regression model of class 'mer', with a modified random intercept and dummy variables indicating the estimates of the neutralized influence of selected grouping levels.

Note

Please note that in its present form, the exclude.influence function only works on mixed effects regression models of class mer that have been estimated using the functions in the lme4 package.

Also, it is required that the mer model was estimated using a factor variable to indicate group levels. When using something similar to + (1 | as.factor(variable)), the function is not able of identifying the correct grouping factors, and returns an error.

Author(s)

Rense Nieuwenhuis, Ben Pelzer, Manfred te Grotenhuis

References

Nieuwenhuis, R., Te Grotenhuis, M., & Pelzer, B. (2012). Influence.ME: tools for detecting influential data in mixed effects models. *R Journal*, 4(2), 38???47.

Belsley, D.A., Kuh, E. & Welsch, R.E. (1980). *Regression Diagnostics. Identifying Influential Data and Source of Collinearity*. Wiley.

Langford, I. H. and Lewis, T. (1998). Outliers in multilevel data. Journal of the Royal Statistical Society: Series A (Statistics in Society), 161:121-160.

Snijders, T.A. & Bosker, R.J. (1999). *Multilevel Analysis, an introduction to basic and advanced multilevel modeling*. Sage.

Van der Meer, T., Te Grotenhuis, M., & Pelzer, B. (2010). *Influential Cases in Multilevel Modeling: A Methodological Comment*. American Sociological Review, 75(1), 173-178.

See Also

influence

Examples

```
## Not run:
data(school23)
model.a <- lmer(math ~ structure + SES  + (1 | school.ID), data=school23)
summary(model.a)
model.b <- exclude.influence(model.a, grouping="school.ID", level="7472")
summary(model.b)
model.c <- exclude.influence(model.a, grouping="school.ID", level=c("7472", "62821"))
summary(model.c)
model.d <- exclude.influence(model.a, obs=1:10)
summary(model.d)

data(Penicillin, package="lme4")
model.d <- lmer(diameter ~ (1|plate) + (1|sample), Penicillin)
summary(model.d)
model.e <- exclude.influence(model.d, grouping="sample", level="A", gf="all")
summary(model.e)

## End(Not run)
```

---

grouping.levels	*Returns the levels of a grouping factor in a mixed effects regression model*

Description

Helper function returning all the levels of a grouping factor in a mixed effects regression model.

Usage

```
grouping.levels(model, group)
```

Arguments

`model`	Mixed effects model of class 'mer'
`group`	Grouping factor of 'model' of which the levels are returned

Details

Please note that at times different results may be obtained by using nesting.levels(), compared with deriving the levels of the grouping factor directly from the (original) data. This is because nesting.levels() only extracts the nesting levels that were de facto used in the model. Due to missing values, this may diverge from those present in the actual data.

Value

Returns a character vector containing all the names / labels of levels of the grouping factor.

Author(s)

Rense Nieuwenhuis, Ben Pelzer, Manfred te Grotenhuis

Examples

```
## Not run:
# Penicillin data originates from the lme4 package.
model <- lmer(diameter ~ (1|plate) + (1|sample), Penicillin)

grouping.levels(model, "plate")
grouping.levels(model, "sample")

## End(Not run)
```

---

influence.mer	*influence returns mixed model estimates, iteratively excluding the influence of data nested within single grouping factors.*

Description

influence() is the workhorse function of the influence.ME package. Based on a priorly estimated mixed effects regression model (estimated using lme4), the influence() function iteratively modifies the mixed effects model to neutralize the effect a grouped set of data has on the parameters, and which returns returns the fixed parameters of these iteratively modified models. These are used to compute measures of influential data.

Usage

```
influence(model, group=NULL, select=NULL, obs=FALSE,
gf="single", count = FALSE, delete=TRUE, ...)
```

Arguments

`model`	Mixed effects model of class 'mer'.
`group`	Grouping factor in model of which iteratively levels are neutralized
`select`	Defines the selection of grouping factors that should be omitted. Defaults to 0, resulting in each level of the grouping factor being omitted iteratively. When a selection is defined, model parameters for the full model, and the altered model are returned. The selection can be a vector of multiple levels of the grouping factor.
`obs`	If obs=TRUE, single observations - rather than groups - are deleted from the model.
`gf`	Indicates from which of the model's grouping factors the influence of the specified grouping factor is to be neutralized. If `gf="single"` (default), the levels of the specified grouping factor are only neutralized regarding the grouping factor specified in `group`. In its present form, `gf="single"` only works on mixed models with a maximum of 2 grouping factors. If `gf="all"`, the influence from the levels of `group` is neutralized regarding all grouping factors in the model. This option only applies to models with more than a single grouping factor.
`count`	If count=TRUE, the remaining number of grouping factors that still need to be omitted are printed.
`delete`	If delete=TRUE (default), the influence is excluded by simply deleting the observations nested within the higher level group. If delete=FALSE, the influence of higher level groups is excluded from the model by setting the intercept-vector for the observations nested within these groups to 0, and by adding a dummy-variable indicating these observations (Langford and Lewis, 1998). This latter option currently does not work with models that include factor variables.
`...`	Optional arguments that are passed on to the lmer/glmer function

Details

The basic rationale behind measuring influential cases is that when iteratively single units are omitted from the data, models based on these data should not produce substantially different estimates. To apply this logic to mixed effects models one has to measure the influence of a particular higher level unit on the estimates of a higher level predictor. This means that the mixed effects model has to be adjusted to neutralize the unit's influence on that estimate, while at the same time allowing the unit's lower-level cases to help estimate the effects of the lower-level predictors in the model. This procedure is based on a modification of the intercept and the addition of a dummy variable for the cases that might be influential.

influence() is the workhorse function of this likewise called package. Based on a priorly estimated mixed effects regression model (of the 'mer' class), the influence() function iteratively modifies the mixed effects model by neutralizing the effect a grouped set of data has on the parameters, and which returns returns the fixed parameters of these iteratively modified models.

The returned object (see 'value') contains information which is required for functions computing various measures of influential data.

Value

The object returned by influence() of class "estex" contains the estimates (excluding the influence of specific (groups of) observations) required by several other functions to calculate measures of influential data. A list containing six elements is returned:

`or.fixed`	Fixed estimates of the original model (based on the full data)
`or.se`	Standard Error of the estimates of the original model
`or.vcov`	Variance / Covariance matrix of the original model
`alt.fixed`	Matrix of the fixed parameters estimate, after iteratively subsets of data are removed. Altered estimates associated with the deletion of data nested within each grouping factor are provided.
`alt.se`	Matrix of the standard errors of the fixed parameter estimates, after iteratively subsets of data are removed. Altered estimates associated with the deletion of data nested within each grouping factor are provided.
`alt.vcov`	Variance / Covariance matrix of the altered models, after iteratively subsets of data are removed. Altered estimates associated with the deletion of data nested within each grouping factor are provided.

Note

Please note that in its present form, the `influence` function only works on mixed effects regression models that have been estimated using the functions in the `lme4` package.

Also, it is required that the `mer` model was estimated using a factor variable to indicate group levels. When using something similar to `+ (1 | as.factor(variable))`, the function is not able of identifying the correct grouping factors, and returns an error.

Since influence() entails the re-estimation of the provided mixed effects model for each level of the specified grouping factor (after alteration of the data), executing this procedure can be computationally highly demanding.

Author(s)

Rense Nieuwenhuis, Ben Pelzer, Manfred te Grotenhuis

References

Nieuwenhuis, R., Te Grotenhuis, M., & Pelzer, B. (2012). Influence.ME: tools for detecting influential data in mixed effects models. *R Journal*, 4(2), 38???47.

Belsley, D.A., Kuh, E. & Welsch, R.E. (1980). *Regression Diagnostics. Identifying Influential Data and Source of Collinearity*. Wiley.

Langford, I. H. and Lewis, T. (1998). Outliers in multilevel data. Journal of the Royal Statistical Society: Series A (Statistics in Society), 161:121-160.

Snijders, T.A. & Bosker, R.J. (1999). *Multilevel Analysis, an introduction to basic and advanced multilevel modeling*. Sage.

Van der Meer, T., Te Grotenhuis, M., & Pelzer, B. (2010). *Influential Cases in Multilevel Modeling: A Methodological Comment*. American Sociological Review, 75(1), 173-178.

See Also

[cooks.distance.estex](), [dfbetas.estex]()

Examples

```
## Not run:
data(school23)
model.a <- lmer(math ~ structure + SES  + (1 | school.ID), data=school23)
alt.est.a <- influence(model=model.a, group="school.ID")
alt.est.b <- influence(model=model.a, group="school.ID", select="7472")
alt.est.c <- influence(model=model.a, group="school.ID", select=c("7472", "62821"))

#Note: does not work on models produced by exclude.influence()
model.b <- lmer(math ~ structure + scale(SES)  + (1 | school.ID), data=school23)
alt.est.d <- influence(model=model.b, group="school.ID", select=c("7472", "62821"))

data(Penicillin, package="lme4")
model.c <- lmer(diameter ~ (1|plate) + (1|sample), Penicillin)
alt.est.e <- influence(model=model.c, group="plate")
alt.est.f <- influence(model=model.c, group="sample")
alt.est.g <- influence(model=model.c, group="sample", gf="all")


## End(Not run)
```

---

pchange   *Compute the percentage change, as measure of influential data*

Description

Computes the percentile change, as a measure of influential data. This unstandardized measure can serve to help interpret the magnitude of the influence single or combined grouping levels exert on mixed effects models. The percentage change in parameter estimates between a (mixed effects) regression model based on a full set of data, and a model from which a (potentially influential) subset of data is removed. A value of percentage change is calculated for each parameter in the model separately, based on the information returned by the influence() function.

Usage

```
pchange(estex, parameters = 0, sort=FALSE, to.sort=NA, abs=FALSE)
```

Arguments

estex	An object as returned by the influence() function, containing the altered estimates of a mixed effects regression model
parameters	Used to define a selection of parameters. If parameters=0 (default), percentage change are calculated for all parameters in the model

sort	If sort=TRUE the values of percentage change are ordered based on magnitude. If sort=FALSE (default) no sorting takes place.
to.sort	Specify on which variable the percentage changes must be sorted. If only one variable present (either in the model, or due to the selection specified in parameters), this parameter can be omitted. If percentage changes are calculated for multiple variables, and sort=TRUE, specification of to.sort is required, or an error is returned.
abs	If abs=TRUE, the absolute values of percentage change are returned, while if abs=FALSE (default), both positive and negative values are possible. If both abs=TRUE and sort=TRUE, the abs parameters precedes the sort parameter, and thus the absolute values of percentage change are sorted.

Value

A matrix is returned, containing values of percentage change for each (selected) fixed parameter estimate of the model, and separately for each evaluated set of influential data.

Author(s)

Rense Nieuwenhuis, Ben Pelzer, Manfred te Grotenhuis

References

Belsley, D.A., Kuh, E. & Welsch, R.E. (1980). *Regression Diagnostics. Identifying Influential Data and Source of Collinearity*. Wiley.

Snijders, T.A. & Bosker, R.J. (1999). *Multilevel Analysis, an introduction to basic and advanced multilevel modeling*. Sage.

Van der Meer, T., Te Grotenhuis, M., & Pelzer, B. (2010). *Influential Cases in Multilevel Modeling: A Methodological Comment*. American Sociological Review, 75(1), 173-178.

See Also

influence, cooks.distance.estex, dfbetas.estex

Examples

```
## Not run:
 data(school23)
 model <- lmer(math ~ structure + SES  + (1 | school.ID), data=school23)

 alt.est <- influence(model, group="school.ID")
 pchange(alt.est)

## End(Not run)
```

plot.estex	*Dotplot visualization of measures of influence*

Description

This is a wrapper function to the dotplot() function in the lattice-package.

Usage

```
## S3 method for class 'estex'
plot(x, which="dfbetas", sort=FALSE, to.sort=NA, abs=FALSE, cutoff=0,
parameters=seq_len(ncol(estex$alt.fixed)),
                        groups=seq_len(nrow(estex$alt.fixed)), ...)
```

Arguments

x	An object as returned by the influence() function, containing the altered estimates of a mixed effects regression model.
which	Select which measure of influence is to be plotted. Available options are: "dfbetas" to visualize dfbetas, "cook" to plot the cook's distances, "pchange" to plot the percentage change, and "sigtest" to plot the test statistic of a parameter estimate after deletion of specific cases.
sort	If sort=TRUE The values of the selected measure of influence are ordered based on magnitude before visualization. If sort=FALSE (default) no sorting takes place.
to.sort	Specify on which variable the values of the selected measure of influence must be sorted. If only one variable present (either in the model, or due to the selection specified in parameters), this parameter can be omitted. If multiple variables are visualized, and sort=TRUE, specification of to.sort is required, or an error is returned.
abs	If abs=TRUE, the absolute values of the values of the selected measure of influence are visualized, while if abs=FALSE (default), both positive and negative values are possible. If both abs=TRUE and sort=TRUE, the abs parameters precedes the sort parameter, and thus the absolute values of the selected measure of influence are sorted.
cutoff	Values of the selected measure of influence exceeding the specified (cutoff) value are plotted visually different from values not exceeding the cutoff. If cutoff=0 (default), no such differentiation is made in the way values are plotted.
parameters	Used to define a selection of parameters. If left unspecified (default), values for the selected measure of influence are visualized for parameters in the model.
groups	Used to define a selection of nesting groups that should be visualized. If left unspecified (default), the values of the selected measure of influence for all nesting groups are shown.

... Further arguments passed on to the dotplot() function.

Author(s)

Rense Nieuwenhuis, Ben Pelzer, Manfred te Grotenhuis

See Also

[influence](), [dfbetas.estex](), [cooks.distance.estex](), [pchange](), [sigtest]()

Examples

```
## Not run:
data(school23)
model <- lmer(math ~ structure + SES  + (1 | school.ID), data=school23)

alt.est <- influence(model, "school.ID")
plot(alt.est, which="dfbetas")
plot(alt.est, which="cook", sort=TRUE)

## End(Not run)
```

---

school23 *Math test performance in 23 schools*

Description

The school23 data contains information on students' performance on a math test, as well as several explanatory variables. These data are subset of the NELS-88 data (National Education Longitudinal Study of 1988). Both a selected number of variables and a selected number of observations are given here.

Format

A data frame with 519 observations on the following 15 variables.

school.ID a factor with 23 levels, representing the 23 schools within which students are nested.

SES a numeric vector, representing the socio-economic status

mean.SES a numeric vector, representing the mean socio-economic status per school

homework a factor representing the time spent on math homework each week, with levels None, Less than 1 hour, 1 hour, 2 hours, 3 hours, 4-6 hours, 7-9 hours, and 10 or more

parented a factor representing the parents' highest education level, with levels Dod not finish H.S., H.S. grad or GED, GT H.S. and LT 4yr degree, College graduate, M.A. or equivalent, and Ph.D., M.D., other

ratio a numeric vector, representing the student-teacher ratio

perc.minor a factor representing the percent minority in school, with levels None, 1-5, 6-10, 11-20, 21-40, 41-60, 61-90, and 91-100

math a numeric vector, representing the number of correct answers on a mathematics test

sex a factor with levels `Male` and `Female`

race a factor with levels `Asian`, `Hispanic`, `Black`, `White`, and `American Indian`

school.type a factor representing the school type, with levels `Public school`, `Catholic school`, `Private, other religious affiliation`, and `Private, no religious affiliation`

structure a numeric vector representing the degree to which the classroom environment is structured. High values represent higher levels of (accurate) classroom environment structure

school.size a factor representing the total school enrollment, with levels `1-199 Students`, `200-399`, `400-599`, `600-799`, `800-999`, `1000-1199`, and `1200+`

urban a factor with levels `Urban`, `Suburban`, and `Rural`

region a factor with levels `Northeast`, `North Central`, `South`, and `West`

Details

Labels for the factors were found in an appendix in Kreft \& De Leeuw (1998). All labels were designated, although in some cases not all possible values are represented in the variable (i.e. `region`). This is probably due to the fact that this is only a subsample from the full NELS-88 data.

Also, some of the variable names were changed.

Source

These data are used in the examples given in Kreft \& De Leeuw (1998). Both the examples and the data are publicly available from the internet: http://www.ats.ucla.edu/stat/examples/imm/. Data reproduced with permission from Jan de Leeuw.

References

Kreft, I. and De Leeuw, J. (1998). *Introducing Multilevel Modeling*. Sage Publications.

Examples

```
## Not run:
data(school23)
model <- lmer(math ~ structure + (1 | school.ID), data=school23)
summary(model)

## End(Not run)
```

Description

Returns the standard errors of the fixed estimates in a mixed effects model.

Usage

```
se.fixef(model)
```

Arguments

model   Mixed effects regression model of class 'mer'

Value

A vector with the standard errors of the fixed parameters of the model.

Note

This is a small helper-function to the influence.ME package. For more elaborate functionality, refer to the se.fixef function in the 'car' package.

Author(s)

Rense Nieuwenhuis, Ben Pelzer, Manfred te Grotenhuis

Examples

```
## Not run:
data(school23)
model <- lmer(math ~  homework + structure + (1 | school.ID), data=school23)
summary(model)
se.fixef(model)

## End(Not run)
```

---

sigtest	*Test for changes in the level of statistical significance resulting from the deletion of potentially influential observations*

Description

Test for changes in the level of statistical significance resulting from the deletion of potentially influential observations

Usage

```
sigtest(estex, test = 1.96, parameters = 0, sort = FALSE, to.sort = NA)
```

Arguments

estex	Object of class 'estex', as returned from the `influence` function.
test	Value of the test statistic against which statistical significance is to be evaluated
parameters	Vector specifying the parameter(s) of which the significance is to be evaluated. If left unspecified, all parameters of the model are evaluated
sort	Specify whether the output should be sorted on the (absolute) magnitude of the test statistic after deletion of potentially influential cases
to.sort	If `sort==true`, the variable on which to sort the output needs to be be specified

Details

The `"sigtest"` function tests whether excluding the influence of a single case changes the statistical significance of any or more variables in the model. This test of significance is based on the test statistic provided by the **lme4** package. The nature of this statistic varies between different distributional families in the generalized mixed effects models. For instance, the t-statistic is related to a normal distribution while the z-statistic is related to binomial distributions.

For each of the cases that are evaluated, the test statistic of each variable is compared to a test-value specified by the user. For the purpose of this test, the parameter is regarded to statistically significant if the test statistic of the model exceeds the specified value. The `"sigtest"` function reports for each variable the test statistic after deletion of each evaluated case, whether or not this updated test statistic results in statistical significance based on the user-specified value, and whether or not this new statistical significance differs from the significance in the original model. So, in other words, if a parameter was statistically significant in the original model, but is not longer significant after the deletion of a specific case from the model, this is indicated by the output of the `"sigtest"` function. It is also indicated when an estimate was not significant originally, but reached statistical significance after deletion of a specific case.

Value

Returns a list. For each variable in the original model that was evaluated, this list contains a matrix showing the test statistic from the original model (column 1), the test statistic after a potentially influential case was excluded from the model (column 2) and the result (TRUE / FALSE) of the test whether statistical significance changed as a result from deletion of (potentially) influential cases.

Author(s)

Rense Nieuwenhuis, Manfred te Grotenhuis, Ben Pelzer

Examples

```
## Not run:
data(school23)
m23 <- lmer(math ~ homework + structure
   + (1 | school.ID),
   data=school23)

estex.m23  <- influence(m23, group="school.ID")

sigtest(estex.m23, test=-1.96)$structure

## End(Not run)
```

Index