Package ‘skewlmm’

June 30, 2023

Type     Package
Title    Scale Mixture of Skew-Normal Linear Mixed Models
Version  1.1.0
Maintainer Fernanda L. Schumacher <fernandalschumacher@gmail.com>
Description It fits scale mixture of skew-normal linear mixed models using an expectation–maximization (EM) type algorithm, including some possibilities for modeling the within-subject dependence. Details can be found in Schumacher, Lachos and Matos (2021) <doi:10.1002/sim.8870>.
License  MIT + file LICENSE
Depends  R (>= 4.2), optimParallel
Encoding UTF-8
Imports  dplyr, furrr, future, ggplot2, ggrepel, haven, matrixcalc, methods, moments, MomTrunc, mvtnorm, nlme, numDeriv, purrr, relliptical, TruncatedNormal
URL https://github.com/fernandalschumacher/skewlmm
BugReports https://github.com/fernandalschumacher/skewlmm/issues
NeedsCompilation no
Author  Fernanda L. Schumacher [aut, cre]
        (<https://orcid.org/0000-0002-5724-8918>),
        Larissa A. Matos [aut] (<https://orcid.org/0000-0002-2635-0901>),
        Victor H. Lachos [aut] (<https://orcid.org/0000-0002-7239-2459>),
        Katherine A. L. Valeriano [aut]
        (<https://orcid.org/0000-0001-6388-4753>),
        Nicholas Henderson [ctb],
        Ravi Varadhan [ctb]
Repository CRAN
Date/Publication 2023-06-30 05:00:02 UTC
### R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>acfresid</td>
<td>3</td>
</tr>
<tr>
<td>boot_ci</td>
<td>4</td>
</tr>
<tr>
<td>boot_par</td>
<td>5</td>
</tr>
<tr>
<td>criteria</td>
<td>6</td>
</tr>
<tr>
<td>errorVar</td>
<td>7</td>
</tr>
<tr>
<td>fitted.SMN</td>
<td>8</td>
</tr>
<tr>
<td>fitted.SMNclmm</td>
<td>9</td>
</tr>
<tr>
<td>fitted.SMSN</td>
<td>10</td>
</tr>
<tr>
<td>healy.plot</td>
<td>11</td>
</tr>
<tr>
<td>lmmControl</td>
<td>12</td>
</tr>
<tr>
<td>lr.test</td>
<td>14</td>
</tr>
<tr>
<td>mahalDist</td>
<td>15</td>
</tr>
<tr>
<td>mahalDistCens</td>
<td>16</td>
</tr>
<tr>
<td>plot</td>
<td>17</td>
</tr>
<tr>
<td>plot.acfresid</td>
<td>18</td>
</tr>
<tr>
<td>plot.mahalDist</td>
<td>19</td>
</tr>
<tr>
<td>plot.mahalDistCens</td>
<td>20</td>
</tr>
<tr>
<td>plot.SMNclmm</td>
<td>21</td>
</tr>
<tr>
<td>predict.SMN</td>
<td>22</td>
</tr>
<tr>
<td>predict.SMNclmm</td>
<td>23</td>
</tr>
<tr>
<td>predict.SMSN</td>
<td>24</td>
</tr>
<tr>
<td>print.SMN</td>
<td>25</td>
</tr>
<tr>
<td>print.SMNclmm</td>
<td>26</td>
</tr>
<tr>
<td>print.SMSN</td>
<td>27</td>
</tr>
<tr>
<td>ranef</td>
<td>27</td>
</tr>
<tr>
<td>residuals</td>
<td>28</td>
</tr>
<tr>
<td>residuals.SMNclmm</td>
<td>29</td>
</tr>
<tr>
<td>rsmsn.clmm</td>
<td>31</td>
</tr>
<tr>
<td>rsmsn.lmm</td>
<td>32</td>
</tr>
<tr>
<td>smn.clmm</td>
<td>34</td>
</tr>
<tr>
<td>smn.lmm</td>
<td>37</td>
</tr>
<tr>
<td>smns.lmm</td>
<td>40</td>
</tr>
<tr>
<td>summary.SMN</td>
<td>43</td>
</tr>
<tr>
<td>summary.SMNclmm</td>
<td>44</td>
</tr>
<tr>
<td>summary.SMSN</td>
<td>45</td>
</tr>
<tr>
<td>update</td>
<td>46</td>
</tr>
</tbody>
</table>

**Index**

48
Description

This function calculates the empirical autocorrelation function for the within-subject residuals from a smn.lmm or smsn.lmm fit. The autocorrelation values are calculated using pairs of residuals within-subjects. The autocorrelation function is useful for investigating serial correlation models for discrete-time data, preferably equally spaced.

Usage

acfresid(object, maxLag, resLevel = "marginal", resType = "normalized", calcCI = FALSE, levelCI, MCiter, seed)

Arguments

object An object inheriting from class SMN or SMSN, representing a fitted scale mixture of (skew) normal linear mixed model.
maxLag An optional integer giving the maximum lag for which the autocorrelation should be calculated. Defaults to maximum lag in the within-subject residuals.
resLevel "marginal" (default) or "conditional". An optional character string specifying which residual should be used. For details see residuals.SMN.
resType "response", "normalized" (default), or "modified". An optional character string specifying which type of residual should be used. For details see residuals.SMN.
calcCI TRUE or FALSE (default). A logical value indicating if Monte Carlo confidence intervals should be computed for the conditionally independent model, which can be used for testing if the autocorrelations are zero.
levelCI An optional numeric value in (0, 1) indicating the confidence level that should be used in the Monte Carlo confidence intervals. Default is 0.95.
MCiter An optional discrete value indicating the number of Monte Carlo samples that should be used to compute the confidence intervals. Default is 300.
seed An optional value used to specify seeds inside the function. Default is to use a random seed.

Value

A data frame with columns lag, ACF, and n.used representing, respectively, the lag between residuals within a pair, the corresponding empirical autocorrelation, and the number of pairs used. If calcCI=TRUE, the data frame has two extra columns containing the confidence intervals for the conditionally independent model. The returned value inherits from class acfresid.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos
References


See Also

smn.lmm, smsn.lmm, plot.acfresid

Examples

fm1 = smn.lmm(distance ~ age+Sex, data=nlme::Orthodont, groupVar="Subject")
acf1 = acfresid(fm1)
plot(acf1)

## computing simulated bands
acfCI = acfresid(fm1, calcCI=TRUE)
plot(acfCI)

boot_ci

Extract confidence intervals from lmmBoot object

Description

It extracts confidence intervals from parametric bootstrap results obtained using the boot_par() function.

Usage

boot_ci(object, conf = 0.95)

Arguments

object An object containing the results of the boot_par() function.
conf Confidence level to be considered.

Value

A matrix containing the confidence intervals.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos
See Also

boot_par, smsn.lmm, smn.lmm

Examples

fm1 = smn.lmm(nlme::Orthodont, formFixed=distance ~ age+Sex, groupVar="Subject")
b1 = boot_par(fm1, B=100)
boot_ci(b1)

boot_par

Parametric bootstrap for SMSN/SMN objects

Description

It generates and estimates B Monte Carlo samples identical to the fitted model.

Usage

boot_par(object, B = 100, seed = 123)

Arguments

object  A smsn.lmm or smn.lmm object containing the fitted model to be updated.
B       Number of samples to be used.
seed    Seed to be used.

Details

This function provides an alternative for the asymptotic standard errors and confidence intervals given in summary, which may be helpful for small samples. Nevertheless, the computational cost is higher and it may take several minutes to get the results.

Value

A tibble of class lmmBoot with B lines, where each line contains the estimated parameters from a simulated sample.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

See Also

boot_ci, smsn.lmm, smn.lmm
Examples

\begin{verbatim}
fm1 = smn.lmm(nlme::Orthodont, formFixed=distance ~ age+Sex, groupVar="Subject")
b1 = boot_par(fm1, B=100)
boot_ci(b1)
\end{verbatim}

---

criteria

Extracts criteria for model comparison of SMSN/SMN/SMNclmm objects

Description

It extracts criteria for model comparison of several SMSN-LMM, SMN-LMM and/or SMN-CLMM (for censored responses).

Usage

\begin{verbatim}
criteria(lobjects)
\end{verbatim}

Arguments

lobjects A list containing the smsn.lmm, smn.lmm or smn.clmm objects to be compared.

Value

A \texttt{data.frame} containing for each model the maximum log-likelihood value, the number of parameters, the AIC, and the BIC.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

References


See Also

\texttt{smsn.lmm}, \texttt{smn.lmm}, \texttt{smn.clmm}

Examples

\begin{verbatim}
fm_norm = smn.lmm(nlme::Orthodont, formFixed=distance~age+Sex, groupVar="Subject")
fm_t = update(fm_norm, distr="t")
fm_sn = smsn.lmm(nlme::Orthodont, formFixed=distance~age+Sex, groupVar="Subject")
fm_cens = smn.clmm(nlme::Orthodont, formFixed=distance~age+Sex, groupVar="Subject")
criteria(list(fm_norm=fm_norm, fm_t=fm_t, fm_sn=fm_sn, fm_cens=fm_cens))
\end{verbatim}
Description

It returns a scale matrix associated with the error term at time times. Can be applied to a smn.lmm or smsn.lmm object or to a specific dependence structure with chosen parameter values.

Usage

```
errorVar(times, object = NULL, sigma2 = NULL, depStruct = NULL, phi = NULL)
```

Arguments

times A vector containing the times for which the matrix should be calculated.

object A smn.lmm or smsn.lmm object for which the variance should be extracted.

sigma2 Common variance parameter, such that $\Sigma = \sigma^2 \cdot R$. Only evaluated if object is.null(object).

depStruct Dependence structure. "UNC" for conditionally uncorrelated ("CI" is also accepted), "ARp" for AR(p) – p is length(phi)–, "CS" for compound symmetry, "DEC" for DEC, and "CAR1" for continuous-time AR(1). Only evaluated if object is.null(object).

phi Parameter vector indexing the dependence structure. Only evaluated if object is.null(object).

Value

Matrix of dimension length(times).

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

References


See Also

```
smsn.lmm, smn.lmm
```
Examples

```r
fm1 = smn.lmm(distance ~ age+Sex, data=nlme::Orthodont, groupVar="Subject")
errorVar(times=1:4,fm1)
#
errorVar(times=1:5, sigma2=1, depStruct="ARp", phi=.5)
errorVar(times=1:5, sigma2=1, depStruct="DEC", phi=c(.5,.8))
```

---

**fitted.SMN**

*Extract smn.lmm fitted values*

### Description

The fitted values are obtained by adding together the population fitted values (based only on the fixed effects estimates) and the estimated contributions of the random effects to the fitted values at grouping levels.

### Usage

```r
## S3 method for class 'SMN'
fitted(object, ...)
```

### Arguments

- **object**
  
  An object inheriting from class `SMN`, representing a fitted scale mixture normal linear mixed model.

- **...**
  
  Additional arguments.

### Value

Vector of fitted values with length equal to `nrow(data)`.

### Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

### See Also

- `smn.lmm`
- `predict.SMN`

### Examples

```r
fm1 = smn.lmm(distance ~ age+Sex, data=nlme::Orthodont, groupVar="Subject")
fitted(fm1)
```
Description

The fitted values are obtained by adding together the population fitted values (based only on the fixed effects estimates) and the estimated contributions of the random effects to the fitted values at grouping levels.

Usage

```r
## S3 method for class 'SMNclmm'
fitted(object, ...)
```

Arguments

- `object` An object inheriting from class `SMNclmm`, representing a fitted scale mixture normal linear mixed model with censored responses.
- `...` Additional arguments.

Value

Vector of fitted values with length equal to `nrow(data)`.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos, Victor H. Lachos and Katherine L. Valeriano

See Also

- `smn.clmm`
- `predict.SMNclmm`

Examples

```r
nj1 = 5; m = 30
time = rep(1:nj1, times=m)
groups = as.factor(rep(1:m, each=nj1))
dat1 = rsmsn.clmm(time, groups, cbind(1, time), rep(1,m*nj1), sigma2=0.7,
                 D=0.6*diag(1), beta=c(1,2), depStruct="UNC")

fm1 = smn.clmm(dat1, formFixed=y~x, groupVar="ind", depStruct="UNC",
               ci="ci", lcl="lcl", ucl="ucl", control=lmmControl(max.iter=30))
fitted(fm1)
```
**fitted.SMSN**

*Extract smsn.lmm fitted values*

**Description**

The fitted values are obtained by adding together the population fitted values (based only on the fixed effects estimates) and the estimated contributions of the random effects to the fitted values at grouping levels.

**Usage**

```r
## S3 method for class 'SMSN'
fitted(object, ...)
```

**Arguments**

- `object` An object inheriting from class SMSN, representing a fitted scale mixture skew-normal linear mixed model.
- `...` Additional arguments.

**Value**

Vector of fitted values with length equal to `nrow(data)`.

**Author(s)**

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

**See Also**

`smsn.lmm`, `predict.SMSN`

**Examples**

```r
fm1 = smsn.lmm(distance ~ age+Sex, data=nlme::Orthodont, groupVar="Subject",
               control=lmmControl(tol=.0001))
fitted(fm1)
```
healy.plot

Healy-type plot from a smn.lmm or smsn.lmm object

Description

It creates a Healy-type plot from a smn.lmm or smsn.lmm object, for goodness-of-fit assessment.

Usage

healy.plot(object, dataPlus = NULL, dotsize = 0.4, calcCI = FALSE, levelCI, MCiter, seed, ...)

Arguments

object An object inheriting from class SMN or SMSN, representing a fitted scale mixture of (skew) normal linear mixed model.
dataPlus Optional. Expanded dataset that should be used instead the one used for fitting. This is necessary for unbalanced datasets, since Haley’s plot requires all subject to have the same number of observations.
dotsize Optional. Dotsize used in ggplot.
calcCI TRUE or FALSE (default). A logical value indicating if Monte Carlo confidence intervals should be computed for the conditionally independent model, which can be used for testing if the autocorrelations are zero.
levelCI An optional numeric value in (0, 1) indicating the confidence level that should be used in the Monte Carlo confidence intervals. Default is 0.95.
MCiter An optional discrete value indicating the number of Monte Carlo samples that should be used to compute the confidence intervals. Default is 300.
seed An optional value used to specify seeds inside the function. Default is to use a random seed.
... Additional arguments.

details

It constructs a Healy-type plot (Healy, 1968) by plotting the nominal probability values $1/n, 2/n, ..., n/n$ against the theoretical cumulative probabilities of the ordered observed Mahalanobis distances. If the fitted model is appropriate, the plot should resemble a straight line through the origin with unit slope. If calcCI=TRUE, the plot presents two dashed blue lines containing approximated confidence intervals by considering that the fitted model is correct.

Value

A ggplot object.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos
References


See Also

ggplot, smn.lmm, smsn.lmm, mahalDist, acfresid

Examples

fm1 = smn.lmm(distance ~ age*Sex, data=nlme::Orthodont, groupVar="Subject")
healy.plot(fm1)

## computing simulated bands
healy.plot(fm1, calcCI=TRUE)

lmmControl

Control options for smn.lmm(), smn.lmm() and ssmn.cllmm()

Description

The values supplied in the function call replace the defaults and a list with all possible arguments is returned. The returned list has class "lmmControl" and is used as the control argument to the smn.lmm(), smn.lmm() and ssmn.cllmm() functions.

Usage

lmmControl(tol = 1e-06, max.iter = 200, calc.se = TRUE, lb = NULL, lu = NULL, luDEC = 10,
initialValues = list(beta = NULL, sigma2 = NULL, D = NULL,
lambda = NULL, phi = NULL, nu = NULL),
quiet = FALSE, showCriterium = FALSE, algorithm = "DAAREM",
parallelphi = NULL, parallelnu = NULL, ncores = NULL,
control.daarem = list())

Arguments

tol Tolerance for the convergence criterion. Default = 1e-6.
max.iter Maximum number of iterations for the EM algorithm. Default = 200.
calc.se A logical value indicating if standard errors should be calculated.
lb Optional. Bottom limit for estimating nu.
lu Optional. Upper limit for estimating nu.
**lmmControl**

Optional. Upper limit for estimating the "damping" parameter for DEC covariance. If luDEC<=1, only attenuation of the exponential decay can be obtained.

**initialValues**
Optional. A named list containing initial parameter values, with at most the following elements: beta, sigma2, D, lambda, phi, nu.

**quiet**
A logical value indicating if the iteration message should be suppressed. Useful when calling the function in R Markdown.

**showCriterium**
A logical value indicating if the criterium should be shown at each iteration.

**algorithm**
Algorithm to be used for estimation, either "DAAREM" (default) or "EM". DAAREM is an acceleration method and usually converges with fewer iterations, but it may result in numerical errors (in this case, please use the "EM" option).

**parallelphi**
A logical value indicating if parallel optimization should be used in the numerical update of the parameters related to the within-subject dependence structure. Default is TRUE if the data contains more than 30 subjects, and FALSE otherwise. Meaningless if depStruct = "UNC".

**parallelnu**
A logical value indicating if parallel optimization should be used in the numerical update of nu. Meaningless if distr="norm" or distr="sn".

**ncores**
Number of cores to be used for the parallel optimization. Meaningless if parallelphi = FALSE and parallelnu = FALSE.

**control.daarem**
List of control for the daarem algorithm. See help(daarem, package = "daarem") for details. Meaningless if algorithm = "EM"

**Author(s)**
Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

**References**


**See Also**

`smsn.lmm`, `smn.lmm`, `smn.clmm`, `update`

**Examples**

```r
lmmControl(algorithm = "EM")

fm1 = smn.lmm(nlme::Orthodont, formFixed=distance + age*Sex,
  groupVar="Subject", control=lmmControl(tol=1e-7))
```
lr.test

Likelihood-ratio test for SMSN/SMN objects

Description

It performs a likelihood-ratio test for two nested SMSN-LMM or SMN-LMM.

Usage

lr.test(obj1, obj2, level = 0.05)

Arguments

obj1, obj2 sman.lmm or smn.lmm objects containing the fitted models to be tested.
level The significance level that should be used. If quiet = TRUE, this is ignored.

Value

statistic The test statistic value.
p.value The p-value from the test.
df The degrees of freedom used on the test.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

References


See Also

sman.lmm, smn.lmm

Examples

fm1 = sman.lmm(nlme::Orthodont, formFixed=distance ~ age+Sex, groupVar="Subject")
fm2 = sman.lmm(nlme::Orthodont, formFixed=distance ~ age+Sex, groupVar="Subject",
control=lmmControl(tol=.0001))
lr.test(fm1, fm2)
Description

Returns the squared Mahalanobis distance from a fitted SMN-LMM or SMSN-LMM.

Usage

```r
mahalDist(object, decomposed = FALSE, dataPlus = NULL)
```

Arguments

- `object`: An object inheriting from class `SMN` or `SMSN`, representing a fitted scale mixture of (skew) normal linear mixed model.
- `decomposed`: Logical. If `TRUE`, the Mahalanobis distance is decomposed in an error term and a random effect term. Default is `FALSE`.
- `dataPlus`: Optional. Expanded dataset that should be used instead the one used for fitting, useful for using Healy’s plot with missing data.

Value

A vector containing the Mahalanobis distance, if `decomposed = FALSE`, or a data frame containing the Mahalanobis distance and its decomposition in error term and random effect (b) term, if `decomposed = TRUE`.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

References


See Also

`snn.lmm`, `smsn.lmm`, `plot.mahalDist`

Examples

```r
fm1 = snn.lmm(distance ~ age+Sex, data=nlme::Orthodont, groupVar="Subject")
mahalDist(fm1)
plot(mahalDist(fm1), nlabels=2)
```
mahalDistCens  

Mahalanobis distance from a smn.clmm object

Description

Returns the squared Mahalanobis distance from a fitted SMN-CLMM. Censored values are imputed using their conditional expectation from the fitting algorithm.

Usage

mahalDistCens(object)

Arguments

object  
An object inheriting from class SMNclmm, representing a fitted scale mixture of normal censored linear mixed model.

Value

An object of class mahalDistCens containing the Mahalanobis distance.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos, Victor H. Lachos and Katherine L. Valeriano

References


See Also

smn.clmm, plot.mahalDistCens

Examples

nj1 = 5; m = 30
time = rep(1:nj1, times=m)
groups = as.factor(rep(1:m, each=nj1))
dat1 = rsmsn.clmm(time, groups, cbind(1,time), rep(1,m*nj1), sigma2=0.7, D=0.5*diag(1), beta=c(1,2), depStruct="CS", phi=0.4)
# Estimation
fm1 = smn.clmm(dat1, formFixed=y~x, groupVar="ind", depStruct="CS", ci="ci", lcl="lcl", ucl="ucl", control=lmmControl(max.iter=30))
distance = mahalDistCens(fm1)
plot(distance, level=0.95, nlabels=2)
plot

Plot a smn.lmm or smsn.lmm object

Description

Fitted values versus residuals plot.

Usage

## S3 method for class 'SMN'
plot(x, type = "response", level = "conditional", 
     useweight = TRUE, alpha = 0.3, ...)

## S3 method for class 'SMSN'
plot(x, type = "response", level = "conditional", 
     useweight = TRUE, alpha = 0.3, ...)

Arguments

x  An object inheriting from class SMN or SMSN, representing a fitted scale mixture of (skew) normal linear mixed model.

type Type of residual that should be used. For details see residuals.SMN. Default is "response", indicating raw residuals.

level Level of residual that should be used. For details see residuals.SMN. Default is "conditional".

useweight A logical value indicating if the estimated weights should be used as color in the plot.

alpha Transparency parameter to be used (0<alpha<1). Meaningless if useweight = TRUE.

... Additional arguments.

Value

A ggplot object.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

See Also

ggplot, smn.lmm, smsn.lmm, fitted.SMN, fitted.SMSN, residuals.SMN, residuals.SMSN
Examples

```r
fm1 = smn.lmm(distance ~ age+Sex, data=nlme::Orthodont,
               groupVar="Subject", distr="t")
plot(fm1)
plot(fm1, useweight=FALSE)

library(ggplot2)
plot(fm1) + ggtitle("t-LMM for orthodont data")
```

---

### plot.acfresid

#### Plot ACF for smn.lmm or smsn.lmm residuals

**Description**

Plot method for objects of class "acfresid".

**Usage**

```r
## S3 method for class 'acfresid'
plot(x,...)
```

**Arguments**

- `x` An object inheriting from class acfresid, representing the empirical autocorrelation function from the residuals of a scale mixture of (skew) normal linear mixed model.
- `...` Additional arguments.

**Value**

A ggplot object.

**Author(s)**

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

**See Also**

`ggplot, acfresid, smn.lmm, smsn.lmm, residuals.SMN, residuals.SMSN`

**Examples**

```r
fm1 = smn.lmm(distance ~ age+Sex, data=nlme::Orthodont, groupVar="Subject")
plot(acfresid(fm1))

acfCI = acfresid(fm1, calcCI=TRUE)
plot(acfCI)
```
plot.mahalDist  

Plot Mahalanobis distance for a fitted smn.lmm or smsn.lmm

Description

Plot method for objects of class "mahalDist". For the total Mahalanobis distance, it gives a quantile for outlier detection, based on the Mahalanobis distance theoretical distribution.

Usage

```r
## S3 method for class 'mahalDist'
plot(x, fitobject, type, level = 0.99, nlabels = 3, ...)
```

Arguments

- `x`: An object inheriting from class `mahalDist`, representing the Mahalanobis distance from a fitted scale mixture of (skew) normal linear mixed model.
- `fitobject`: Optional. An object inheriting from class `SMN` or `SMSN`, representing the fitted scale mixture of (skew) normal linear mixed model that was used for calculating the Mahalanobis distance.
- `type`: Optional. Either "total" (default), for the standard Mahalanobis distance, "error", for the error term of the decomposition, or "b" for the random effect term of the decomposition. For details see `mahalDist`.
- `level`: An optional numeric value in (0, 1) indicating the level of the quantile. It only has utility if `type="total"`. Default is 0.99.
- `nlabels`: Number of observations that should be labeled. Default is 3.
- `...`: Additional arguments.

Value

A ggplot object, plotting the index versus the Mahalanobis distance, if all subject have the same number of observations; or plotting the number of observations per subject versus the Mahalanobis, otherwise.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

See Also

`ggplot`, `mahalDist`, `smn.lmm`, `smsn.lmm`
Examples

```r
fm1 = smn.lmm(distance ~ age+Sex, data=nlme::Orthodont, groupVar="Subject")
plot(mahalDist(fm1), nlabels=2)

#the estimated quantile is stored at the attribute "info" of the plot object
plotMD = plot(mahalDist(fm1))
attr(plotMD, "info")
```

Description

Plot method for objects of class "mahalDistCens". It also gives a quantile for outlier detection, based on the Mahalanobis distance theoretical distribution.

Usage

```r
## S3 method for class 'mahalDistCens'
plot(x, fitobject, level = 0.99, nlabels = 3, ...)
```

Arguments

- `x`: An object inheriting from class `mahalDistCens`, representing the Mahalanobis distance from a fitted scale mixture of normal censored linear mixed model.
- `fitobject`: Optional. An object inheriting from class `SMNclmm`, representing the fitted scale mixture of normal linear mixed model that was used for calculating the Mahalanobis distance.
- `level`: An optional numeric value in \((0, 1)\) indicating the level of the quantile. Default is \(0.99\).
- `nlabels`: Number of observations that should be labeled. Default is 3.
- `...`: Additional arguments.

Value

A `ggplot` object, plotting the index versus the Mahalanobis distance, if all subject have the same number of observations; or plotting the number of observations per subject versus the Mahalanobis, otherwise.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos, Victor H. Lachos and Katherine L. Valeriano

See Also

`ggplot, mahalDistCens, smn.clmm`
Examples

nj1 = 5; m = 30
time = rep(1:nj1, times=m)
groups = as.factor(rep(1:m, each=nj1))
dat1 = rsmsn.clmm(time, groups, cbind(1,time), rep(1,m*nj1), sigma2=0.7,
               D=0.5*diag(1), beta=c(1,2), depStruct="CS", phi=0.4)

# Estimation
fm1 = smn.clmm(dat1, formFixed=y~x, groupVar="ind", depStruct="CS", ci="ci",
           lcl="lcl", ucl="ucl", control=lmmControl(max.iter=30))
distance = mahalDistCens(fm1)
plot(distance, level=0.95, nlabels=2)

plot.SMNclmm

Plot a smn.clmm object

Description

Fitted values versus residuals plot. Censored values are imputed using their conditional expectation from the fitting algorithm.

Usage

## S3 method for class 'SMNclmm'
plot(x, level = "conditional", useweight = TRUE,
    alpha = 0.3, ...)

Arguments

x An object inheriting from class SMNclmm, representing a fitted scale mixture of normal censored linear mixed model.
level Level of residual that should be used. For details see residuals.SMNclmm. Default is "conditional".
useweight A logical value indicating if the estimated weights should be used as color in the plot.
alpha Transparency parameter to be used (0<alpha<1). Meaningless if useweight = TRUE.
... Additional arguments.

Value

A ggplot object.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos, Victor H. Lachos and Katherine L. Valeriano
predict.SMN

Prediction of future observations from an smn.lmm object

Description

Predicted values are obtained through conditional expectation. For details, see Schumacher, Lachos, and Matos (2021).

Usage

## S3 method for class 'SMN'
predict(object, newData, ...)

Arguments

object

An object inheriting from class SMN, representing a fitted scale mixture normal linear mixed model.

newData

A data frame for which response variable should be predicted, including covariates, groupVar and possibly timeVar. If missing or NULL, fitted values are returned.

...

Additional arguments.

Value

A data frame with covariates, groupVar and ypred, where ypred contains the predicted values from the response variable.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos
predict.SMNC1mm

References


See Also

smn.lmm, fitted.SMNC1mm

Examples

dat1 = nlme::Orthodont
fm1 = smn.lmm(distance ~ age+Sex, data=subset(dat1, age<14), groupVar="Subject")
predict(fm1, subset(dat1, age==14))

Description

Predicted values are obtained through conditional expectation. For details, see Schumacher, Lachos, and Matos (2021).

Usage

```r
## S3 method for class 'SMNC1mm'
predict(object, newData, ...)
```

Arguments

- **object**
  An object inheriting from class SMNC1mm, representing a fitted scale mixture normal linear mixed model.

- **newData**
  A data frame for which response variable should be predicted, including covariates, groupVar and possibly timeVar. If missing or NULL, fitted values are returned.

- **...**
  Additional arguments.

Value

A data frame with covariates, groupVar and ypred, where ypred contains the predicted values from the response variable.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos, Victor H. Lachos and Katherine L. Valeriano
References


See Also

smn.clmm, fitted.SMNclmm

Examples

# Simulated example: 20 individuals with 5 times for estimation and # 1 time for prediction
set.seed(963)
nj1 = 6; m = 20
time = rep(1:nj1, times=m)
groups = as.factor(rep(1:m, each=nj1))
dat1 = rsmsn.clmm(time, groups, cbind(1,time), rep(1,m*nj1), sigma2=0.4,
D=0.6*diag(1), beta=c(1,2), depStruct="CS", phi=0.25)
# Estimation
fm1 = smn.clmm(subset(dat1,time<6), formFixed=y~x, groupVar="ind",
depStruct="CS", ci="ci", lcl="lcl", ucl="ucl",
control=lmmControl(max.iter=30, tol=1e-4))
# Prediction
pred = predict(fm1, subset(dat1,time==6))
# MSPE
mean((subset(dat1,time==6)$y - pred$ypred)^2)

predict.SMSN

Prediction of future observations from an smsn.lmm object

Description

Predicted values are obtained through conditional expectation. For details, see Schumacher, Lachos, and Matos (2021).

Usage

## S3 method for class 'SMSN'
predict(object, newData, ...)

Arguments

object An object inheriting from class SMSN, representing a fitted scale mixture skew-normal linear mixed model.

newData A data frame for which response variable should be predicted, including covariates, groupVar and possibly timeVar. If missing or NULL, fitted values are returned.

... Additional arguments.
Value

A data frame with covariates, groupVar and ypred, where ypred contains the predicted values from the response variable.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

References


See Also

smsn.lmm, fitted.SMSN

Examples

dat1 = nlme::Orthodont
fm1 = smsn.lmm(distance ~ age+Sex, data=subset(dat1, age<14), groupVar="Subject")
predict(fm1, subset(dat1, age==14))

print.SMN  

Description

Print a smn.lmm object.

Usage

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

Arguments

x An object inheriting from class SMN, representing a fitted scale mixture normal linear mixed model.

... Additional print arguments.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos
See Also

\texttt{smn.1mm}, \texttt{summary.SMN}

Examples

\begin{verbatim}
  fm1 = smn.1mm(distance ~ age+Sex, data=nlme::Orthodont, groupVar="Subject")
  fm1
\end{verbatim}

print.SMNclmm
Print a \texttt{smn.clmm} object

Description

Print a \texttt{smn.clmm} object.

Usage

\begin{verbatim}
## S3 method for class 'SMNclmm'
print(x, confint.level = 0.95, ...)
\end{verbatim}

Arguments

- \texttt{x} An object inheriting from class \texttt{SMNclmm}, representing a fitted scale mixture normal linear mixed model with censored responses.
- \texttt{confint.level} Level of the approximate confidence intervals presented.
- \texttt{...} Additional print arguments.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos, Victor H. Lachos and Katherine L. Valeriano

See Also

\texttt{smn.clmm}, \texttt{summary.SMNclmm}

Examples

\begin{verbatim}
  nj1 = 5; m = 30
  time = rep(1:nj1, times=m)
  groups = as.factor(rep(1:m, each=nj1))
  dat1 = rsmsn.clmm(time, groups, cbind(1,time), rep(1,m*nj1), sigma2=0.7,
                   D=0.6*diag(1), beta=c(1,2), depStruct="UNC")
  fm1 = smn.clmm(dat1, formFixed=y~x, groupVar="ind", formRandom=~1,
                  depStruct="UNC", ci="ci", lcl="lcl", ucl="ucl")
  fm1
\end{verbatim}
print.SMSN  

Print a smsn.lmm object

Description
Print a smsn.lmm object.

Usage
```r
## S3 method for class 'SMSN'
print(x, ...)
```

Arguments
- `x`: An object inheriting from class SMSN, representing a fitted scale mixture skew-normal linear mixed model.
- `...`: Additional print arguments.

Author(s)
Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

See Also
- `smsn.lmm`, `summary.SMSN`

Examples
```r
fm1 = smsn.lmm(distance ~ age+Sex, data=nlme::Orthodont, groupVar="Subject")
fml
```

---

ranef  

Extract random effects from smsn.lmm, smn.lmm and smn.clmm objects

Description
It extracts random effects from smsn.lmm, smn.lmm and smn.clmm objects.

Usage
```r
ranef(object)
```
Arguments

object An object inheriting from class SMSN, representing a fitted scale mixture skew-normal linear mixed model.

Value

Matrix of estimated random effects.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

References


See Also

smsn.lmm, smn.lmm, smn.clmm, fitted.SMSN, fitted.SMN, fitted.SMNclmm

Examples

```r
fm1 = smn.lmm(distance ~ age+Sex, data=nlme::Orthodont, groupVar="Subject")
ranef(fm1)
```

residuals Extract model residuals from smn.lmm and smsn.lmm objects

Description

The conditional residuals are obtained by subtracting the fitted values from the response vector, while the marginal residuals are obtained by subtracting only the fixed effects from the response vector.

Usage

```r
## S3 method for class 'SMN'
residuals(object, level = "conditional", type = "response", ...)

## S3 method for class 'SMSN'
residuals(object, level = "conditional", type = "response", ...)
```
Arguments

- **object**: An object inheriting from class SMN or SMSN, representing a fitted scale mixture of (skew) normal linear mixed model.
- **level**: Either "conditional", for obtaining conditional residuals, or "marginal", for marginal residuals.
- **type**: An optional character string specifying the type of residuals to be used. If "response", as by default, the "raw" residuals (observed - fitted) are used; if "normalized", the standardized residuals (residuals pre-multiplied by the inverse square-root of the estimated variance matrix) are used; else, if "modified", modified residuals (residuals pre-multiplied by the inverse square-root of the estimated scale matrix) are used.
- ... Additional arguments.

Details

Modified residuals are useful when the variance is not finite, such as when \( \nu \leq 2 \) for t or ST distributions, or when \( \nu \leq 1 \) for SL or SSL distributions.

Value

Vector with the residuals of length equal to `nrow(data)`.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

See Also

- `smn.lmm`, `smsn.lmm`, `acfresid`, `mahalDist`, `healy.plot`

Examples

```r
fm1 = smn.lmm(distance ~ age+Sex, data=nlme::Orthodont, groupVar="Subject")
residuals(fm1)
plot(fm1, useweight=FALSE)
```

---

**residuals.SMNclmm**

*Extract model residuals from smn.clmm objects*

Description

The conditional residuals are obtained by subtracting the fitted values from the response vector, while the marginal residuals are obtained by subtracting only the fixed effects from the response vector.

Censored values are imputed using their conditional expectation from the fitting algorithm.
Usage

```r
## S3 method for class 'SMNclmm'
residuals(object, level = "conditional", ...)
```

Arguments

- **object**: An object inheriting from class `SMNclmm`, representing a fitted scale mixture of normal censored linear mixed model.
- **level**: Either "conditional", for obtaining conditional residuals, or "marginal", for marginal residuals.
- **...**: Additional arguments.

Value

A vector with the residuals of length equal to `nrow(data)`.

Note

The residuals are computed after imputing the censored observations.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos, Victor H. Lachos and Katherine L. Valeriano

See Also

- `smn.clmm`, `mahalDistCens`

Examples

```r
nj1 = 5; m = 30
time = rep(1:nj1, times=m)
groups = as.factor(rep(1:m, each=nj1))
dat1 = rsmsn.clmm(time, groups, cbind(1,time), rep(1,m*nj1), sigma2=0.7,
                  D=0.6*diag(1), beta=c(1,2), depStruct="CS", phi=0.4)

fm1 = smn.clmm(dat1, formFixed=y~x, groupVar="ind", depStruct="CS", ci="ci",
               lcl="lcl", ucl="ucl", control=lmmControl(max.iter=30))
residuals(fm1)
```
rsmsn.clmm

Generate data from SMSN-CLMM with censored responses

Description

It creates a simulated data set from SMSN-CLMM with several possible dependence structures, with an established censoring rate or a fixed limit of detection (LOD).

Usage

rsmsn.clmm(time, ind, x, z, sigma2, D, beta, lambda=rep(0, nrow(D)),
  depStruct="UNC", phi=NULL, distr="norm", nu=NULL, type="left",
  pcens=0.10, LOD=NULL)

Arguments

time Vector of length \(N\) containing times that should be used in data generation, where \(N\) indicates the total number of observations.

ind Vector of length \(N\) containing the variable which represents the subjects or groups.

x Design matrix for fixed effects of dimension \(N \times p\).

z Design matrix for random effects of dimension \(N \times q\).

sigma2 Common variance parameter, such that \(\Sigma = \sigma^2 \cdot R\).

D Variance matrix for random effects.

beta Vector of fixed effects parameter.

lambda Skewness parameter of random effects.

depStruct Dependence structure. "UNC" for conditionally uncorrelated ("CI" is also accepted), "ARp" for AR(p) – \(p\) is length(\(\phi\)) –, "CS" for compound symmetry, "DEC" for DEC, "CAR1" for continuous-time AR(1), and "MA1" for moving average of order 1.

phi Parameter vector indexing the dependence structure.

distr Distribution that should be used. "norm" for normal, "t" for Student-t, "sn" for skew-normal, and "st" for skew-t.

nu Degrees of freedom for Student-t and skew-t distributions. It must be greater than 2.

type left for left censoring and right for right censoring.

pcens Desired censoring rate.

LOD Desired limit of detection. If LOD is provided, then pcens will be discarded.

Value

A data frame containing time, the variable indicating groups (\(\text{ind}\)), the generated response variable (\(y\)), the censoring indicator variable (\(\text{ci}\)), the lower censoring limit (\(lcl\)), the upper censoring limit (\(ucl\)), and possible covariates.
Author(s)
Fernanda L. Schumacher, Larissa A. Matos, Victor H. Lachos and Katherine L. Valeriano

References


See Also
smn.clmm

Examples
library(ggplot2)

# Generating a sample for m=25 individuals with 5 times
nj1 = 5
m = 25
time = rep(1:nj1, times=m)
groups = as.factor(rep(1:m, each=nj1))

# Considering 10% of right censoring responses and normal distribution
dat1 = rsmsn.lmm(time, groups, cbind(1,time), rep(1,m*nj1), sigma2=.25,
D=0.5*diag(1), beta=c(1,2), depStruct="ARp", phi=0.5,
type="right")

head(dat1)
ggplot(dat1, aes(x=x, y=y, group=ind)) + geom_line() +
  stat_summary(aes(group=1), geom="line", fun=mean, col="blue", size=1.5) +
geom_line(aes(x=x,y=lcl), color="red", linetype="dashed")

# Considering LOD=4, left censoring, and Student-t distribution
dat2 = rsmsn.lmm(time, groups, cbind(1,time), rep(1,m*nj1), sigma2=.25,
D=0.5*diag(1), beta=c(1,2), depStruct="CS", phi=0.2,
distr="t", nu=4, LOD=4)

mean(dat2$ci) #censoring rate
ggplot(dat2, aes(x=x, y=y, group=ind)) + geom_line() +
  stat_summary(aes(group=1), geom="line", fun=mean, col="blue", size=1.5) +
geom_line(aes(x=x,y=ucl), color="red", linetype="dashed")

rsmsn.lmm

Generate data from SMSN-LMM

Description
It creates a simulated data set from SMSN-LMM (or from SMN-LMM, if lambda = 0) with several possible dependence structures, for one subject.
Usage

rsmsn.lmm(time1, x1, z1, sigma2, D1, beta, lambda, depStruct = "UNC", phi = NULL, distr = "sn", nu = NULL)

Arguments

time1 Vector containing times that should be used in data generation.
x1 Design matrix for fixed effects.
z1 Design matrix for random effects.
sigma2 Common variance parameter, such that \( \Sigma = \sigma^2 \cdot R \).
D1 Variance matrix for random effects.
beta Vector of fixed effects parameter.
lambda Skewness parameter of random effects.
depStruct Dependence structure. "UNC" for conditionally uncorrelated ("CI" is also accepted), "ARp" for AR(p) – p is length(phi) –, "CS" for compound symmetry, "DEC" for DEC, and "CAR1" for continuous-time AR(1).
phi Parameter vector indexing the dependence structure.
distr Distribution that should be used. "sn" for skew-normal, "st" for skew-t, "ss" for skew-slash, and "scn" for skew-contaminated normal.
nu Parameter vector indexing distr. Should be NULL for "sn", be a vector of length 1 for "st" and "ss", and of length 2 for "scn".

Value

A data frame containing time, the generated response variable \((y)\), and possible covariates.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

References


See Also

smsn.lmm
Examples

```r
# Generating a sample for 1 individual at 5 times
nj1 = 5
rsmsn.lmm(1:nj1, cbind(1, 1:nj1), rep(1, nj1), sigma2=.25, D1=diag(1),
          beta=c(1, 2), lambda=2, depStruct="ARp", phi=.5,
          distr="st", nu=5)

# Generating a sample for m=25 individuals with 5 times
library(dplyr)
library(purrr)
library(ggplot2)
nj1 = 5
m = 25
gendatList = map(rep(nj1, m),
          function(nj) rsmsn.lmm(1:nj, cbind(1, 1:nj), rep(1, nj),
                        sigma2=.25, D1=.5*diag(1), beta=c(1, 2),
                        lambda=2, depStruct="ARp", phi=.5))
gendat = bind_rows(gendatList, .id="ind")
ggplot(gendat, aes(x=x, y=y, group=ind)) + geom_line() +
      stat_summary(aes(group=1), geom="line", fun=mean, col="blue", size=2)
#
fm1 = smsn.lmm(y ~ x, data=gendat, groupVar="ind", depStruct="ARp",
               pAR=1)
summary(fm1)
```

```
smn.clmm

ML estimation of scale mixture of normal linear mixed models with
censored responses

Description

It fits left, right, or interval censored scale mixture of normal linear mixed model with possible within-subject dependence structure, using the EM algorithm. It provides estimates and standard errors of parameters.

Usage

```r
smn.clmm(data, formFixed, groupVar, formRandom = ~1, depStruct = "UNC",
          ci, lcl, ucl, timeVar = NULL, distr = "norm",
          nufix = FALSE, pAR = 1, control = lmmControl())
```

Arguments

data A data frame containing the variables named in formFixed, formRandom, groupVar, timeVar, ci, lcl, and ucl.
formFixed A two-sided linear formula object describing the fixed effects part of the model, with the response on the left of a ~ operator and the covariates, separated by + operators, on the right.
```
groupVar  A character containing the name of the variable which represents the subjects or groups in data.

formRandom  A one-sided linear formula object describing the random effects part of the model, with the covariates, separated by + operators, on the right of a ~ operator. By default, a model with random intercept is considered.

depStruct  A character indicating which dependence structure should be used. "UNC" for conditionally uncorrelated ("CI" is also accepted), "ARp" for AR(p) – p is length(phi) – , "CS" for compound symmetry, "DEC" for DEC, "CAR1" for continuous-time AR(1), and "MA1" for moving average of order 1.

ci  A character containing the name of the censoring indicator variable in data, which should be 1 if the respective observation is censored or missing, and 1 otherwise. If missing, it is assumed that none of the observations is censored.

lcl  A character containing the name of the lower censoring limit in data. If missing, it is assumed lcl=-Inf, i.e., no left limit.

ucl  A character containing the name of the upper censoring limit in data. If missing, it is assumed ucl=Inf, i.e., no right limit.

timeVar  A character containing the name of the variable which represents the time in data. Meaningless if depStruct="UNC" or depStruct="CS". For other structures, if is.null(timeVar) the observations are considered equally spaced and ordered. If depStruct="ARp", timeVar must be an index, preferably starting at 1.

distr  A character indicating which distribution should be used. "norm" for normal and "t" for Student-t.

nufix  TRUE or FALSE indicating if nu should be estimated for t distribution. If nufix=TRUE, nu must be specified through lmmControl().

pAR  If depStruct="ARp", pAR indicates the order of the autoregressive process that should be used (1 by default). Otherwise, it is meaningless.

control  An object resulting from the function lmmControl(), containing additional options for the estimation algorithm.

Details

It fits the model \( Y_i = X_i \beta + Z_i b_i + \epsilon_i \), for \( i = 1, \ldots, n \), where \( Y_i \) is a vector with \( n_i \) observed continuous responses, \( b_i \ SIM(0, D; H) \) and \( \epsilon_i \ SIM(0, \Sigma_i; H) \), indexed by the same mixing distribution.

For details see Matos et al. (2013) and Lachos et al. (2019).

Value

An object of class "SMNclmm" representing the SMN-CLMM fit. Generic functions such as print and summary have methods to show the results of the fit. The functions fitted and ranef can be used to extract some of its components.

Specifically, the following components are returned:

theta  Named vector with parameter estimates.
Number of iterations runned.

A named list containing parameter estimates.

Imputed values in the response variable.

Estimated weights.

A vector containing the log-likelihood value from each iteration of the estimation procedure.

Estimated random effects.

A vector with standard errors.

Value of the log-likelihood at last iteration.

Time elapsed in processing, in seconds.

Convergence criterion at last iteration.

A list with AIC, BIC, and SIC criterion.

The `smn.clmm` call that produced the object.

The data frame used on `smn.clmm` call.

A list containing the formulas used on `smn.clmm` call.

A character indicating which dependence structure was used.

A character indicating which structure was used for the random effects scale matrix.

A character indicating which distribution was used.

The number of observations used.

The number of individuals/groups used.

A character indicating the name of the grouping variable.

A character indicating the name of the time variable, if any.

A vector of fitted values.

**Author(s)**

Larissa A. Matos, Victor H. Lachos, Katherine L. Valeriano and Fernanda L. Schumacher

**References**


**See Also**

`lmmControl`, `update`, `predict.SMNclmm`, `residuals.SMNclmm`, `plot.SMNclmm`, `smn.lmm`, `smsn.lmm`
Examples

# Generating a sample for m=30 individuals with 5 times
# Considering 10% of left censoring responses
# AR(1) and normal distribution
nj1 = 5
m = 30
time = rep(1:nj1, times=m)
groups = as.factor(rep(1:m, each=nj1))
dat1 = rsmsn.clmm(time = time, ind = groups, x = cbind(1,time), z = rep(1,m*nj1),
                 sigma2=0.6, D=0.5*diag(1), beta=c(1,2), depStruct="ARp",
                 phi=0.4, pcens = .1, type = "left")

# Estimation using UNC
fm0 = smn.clmm(dat1, formFixed=y~x, groupVar="ind", ci="ci", ucl="ucl")
summary(fm0)

# Estimation using AR(1)
fm1 = update(fm0, depStruct="ARp")

# Estimation using AR(1) and t distribution
fm2 = update(fm1, distr="t")

# Comparing fitted models
criteria(list(fm0= fm0, fm1=fm1, fm2=fm2))

smn.lmm: ML estimation of scale mixture of normal linear mixed models

Description

It fits a scale mixture of normal linear mixed model with possible within-subject dependence structure, using an EM-type algorithm (via the DAAREM method, by default). It provides estimates and standard errors of parameters.

Usage

smn.lmm(data, formFixed, groupVar, formRandom = ~1, depStruct = "UNC",
timeVar = NULL, distr = "norm", covRandom = "pdSymm",
pAR = 1, control = lmmControl())

Arguments

data A data frame containing the variables named in formFixed, formRandom, groupVar, and timeVar.
formFixed A two-sided linear formula object describing the fixed effects part of the model, with the response on the left of a ~ operator and the covariates, separated by + operators, on the right.
groupVar A character containing the name of the variable which represents the subjects or groups in data.

formRandom A one-sided linear formula object describing the random effects part of the model, with the covariates, separated by + operators, on the right of a ~ operator. By default, a model with random intercept is considered.

depStruct A character indicating which dependence structure should be used. "UNC" for conditionally uncorrelated ("CI" is also accepted), "ARp" for AR(p) – p is \text{length}(\phi_1) – , "CS" for compound symmetry, "DEC" for DEC, and "CAR1" for continuous-time AR(1).

timeVar A character containing the name of the variable which represents the time in data. Meaningless if depStruct="UNC" or depStruct="CS". For other structures, if is.null(timeVar) the observations are considered equally spaced and ordered. If depStruct="ARp", timeVar must be an index, preferably starting at 1.

distr A character indicating which distribution should be used. "norm" for normal, "t" for t, "sl" for slash, and "cn" for contaminated normal.

covRandom A character indicating which structure should be used for the random effects scale matrix (either "pdSymm" (default), for a general positive-definite matrix, or "pdDiag", for a diagonal matrix).

pAR If depStruct="ARp", pAR indicates the order of the autoregressive process that should be used. Otherwise, it is meaningless.

control An object resulting from the function \text{lmmControl()}, containing additional options for the estimation algorithm.

Details

It fits the model \( Y_i = X_i \beta + Z_i b_i + \epsilon_i \), for \( i = 1, \ldots, n \), where \( Y_i \) is a vector with \( n_i \) observed continuous responses, \( b_i \sim \text{SMN}(0, D; H) \) and \( \epsilon_i \sim \text{SMN}(0, \Sigma_i; H) \), indexed by the same mixing distribution.

For efficiency, the DAAREM method is used for parameter estimation. In case of numerical errors, please try passing to \text{lmmControl(algorithm = "DAAREM")} to the control argument.

For details see Schumacher, Lachos & Matos (2021).

Value

An object of class "SMN" representing the SMN-LMM fit. Generic functions such as print and summary have methods to show the results of the fit. The functions fitted and ranef can be used to extract some of its components.

Specifically, the following components are returned:

theta Named vector with parameter estimates.

iter Number of iterations runned.

estimates A named list containing parameter estimates.

uhat Estimated weights.
loglik.track  Vector containing the log-likelihood value from each iteration of the estimation procedure.
random.effects  Estimated random effects.
std.error  A vector with standard errors.
loglik  Value of the log-likelihood at last iteration.
elapsedTime  Time elapsed in processing, in seconds.
error  Convergence criterion at last iteration.
call  The smn.lmm call that produced the object.
criteria  A list with AIC and BIC criterion.
data  The data frame used on smn.lmm call.
formula  A list containing the formulas used on smn.lmm call.
depStruct  A character indicating which dependence structure was used.
covRandom  A character indicating which structure was used for the random effects scale matrix.
distr  A character indicating which distribution was used.
N  The number of observations used.
n  The number of individuals/groups used.
groupVar  A character indicating the name of the grouping variable.
control  The control list used in the function’s call.
timeVar  A character indicating the name of the time variable, if any.
fitted  A vector of fitted values, if calc.bi=TRUE.

Author(s)
Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

References

See Also
lmmControl, update, predict.SMN, residuals.SMN, plot.SMN, smsn.lmm, smn.clmm
Examples

# simple example
dat1 = as.data.frame(nlme::Orthodont)
fm1 = smn.lmm(dat1, formFixed=distance ~ age, groupVar="Subject",
             control=lmmControl(max.iter=30))
fm1

# fitting for several distributions / dependence structures
fm1 = smn.lmm(dat1, formFixed=distance ~ age+Sex, groupVar="Subject")
fm2 = smn.lmm(dat1, formFixed=distance ~ age+Sex, groupVar="Subject",
             distr="t")
fm3 = smn.lmm(dat1, formFixed=distance ~ age+Sex, groupVar="Subject",
             distr="sl")
fm4 = smn.lmm(dat1, formFixed=distance ~ age+Sex, groupVar="Subject",
             depStruct="ARp", pAR=1)
criteria(list(fm1=fm1, fm2=fm2, fm3=fm3, fm4=fm4))
summary(fm3)

# some diagnostic tools
plot(fm3)
acf3 = acfresid(fm3, calcCI=TRUE, MCiter=100)
plot(acf3)
plot(mahalDist(fm3), nlabels=2)
healy.plot(fm3)

---

**smsn.lmm**  
*ML estimation of scale mixture of skew-normal linear mixed models*

Description

It fits a scale mixture of skew-normal linear mixed model with possible within-subject dependence structure, using an EM-type algorithm (via the DAAREM method, by default). It provides estimates and standard errors of parameters.

Usage

```r
smrn.lmm(data, formFixed, groupVar, formRandom = ~1, depStruct = "UNC",
         timeVar = NULL, distr = "sn", covRandom = "pdSymm",
         skewInd, pAR = 1, control = lmmControl())
```

Arguments

- **data**  
  A data frame containing the variables named in `formFixed`, `formRandom`, `groupVar`, and `timeVar`.

- **formFixed**  
  A two-sided linear formula object describing the fixed effects part of the model, with the response on the left of a `~` operator and the covariates, separated by `+` operators, on the right.
groupVar A character containing the name of the variable which represents the subjects or groups in data.

formRandom A one-sided linear formula object describing the random effects part of the model, with the covariates, separated by + operators, on the right of a ~ operator. By default, a model with random intercept is considered.

depStruct A character indicating which dependence structure should be used. "UNC" for conditionally uncorrelated ("CI" is also accepted), "ARp" for AR(p) – p is length(phi) – , "CS" for compound symmetry, "DEC" for DEC, and "CAR1" for continuous-time AR(1).

timeVar A character containing the name of the variable which represents the time in data. Meaningless if depStruct="UNC" or depStruct="CS". For other structures, if is.null(timeVar) the observations are considered equally spaced and ordered. If depStruct="ARp", timeVar must be an index, preferably starting at 1.

distr A character indicating which distribution should be used. "sn" for skew-normal, "st" for skew-t, "ssl" for skew-slash, and "scn" for skew-contaminated normal.

covRandom A character indicating which structure should be used for the random effects scale matrix (either "pdSymm" (default), for a general positive-definite matrix, or "pdDiag", for a diagonal matrix).

skewind A vector of length equal to the number of random effects, containing 0's and 1's, indicating which elements of the skewness parameter vector should be estimated (optional, default is an all-ones vector).

pAR If depStruct="ARp", pAR indicates the order of the autoregressive process that should be used. Otherwise, it is meaningless.

control An object resulting from the function lmmControl(), containing additional options for the estimation algorithm.

Details
It fits the model $Y_i = X_i \beta + Z_i b_i + \epsilon_i, \text{ for } i = 1, \ldots, n,$ where $Y_i$ is a vector with $n_i$ observed continuous responses, $b_i \text{ SMSN}(c\Delta, D, \lambda; H)$ and $\epsilon_i \text{ SMN}(0, \Sigma_i; H)$, indexed by the same mixing distribution.

For efficiency, the DAAREM method is used for parameter estimation. In case of numerical errors, please try passing to lmmControl(algorithm = "DAAREM") to the control argument.
For details see Schumacher, Lachos & Matos (2021).

Value
An object of class "SMSN" representing the SMSN-LMM fit. Generic functions such as print and summary have methods to show the results of the fit. The functions fitted and ranef can be used to extract some of its components.

Specifically, the following components are returned:

theta Named vector with parameter estimates.

iter Number of iterations runned.
estimates A named list containing parameter estimates.

uhat Estimated weights.

loglik.track Vector containing the log-likelihood value from each iteration of the estimation procedure.

random.effects Estimated random effects.

std.error A vector with standard errors.

loglik Value of the log-likelihood at last iteration.

elapsedTime Time elapsed in processing, in seconds.

error Convergence criterion at last iteration.

call The smsn.lmm call that produced the object.

criteria A list with AIC and BIC criterion.

data The data frame used on smsn.lmm call.

formula A list containing the formulas used on smsn.lmm call.

depStruct A character indicating which dependence structure was used.

covRandom A character indicating which structure was used for the random effects scale matrix.

distr A character indicating which distribution was used.

N The number of observations used.

n The number of individuals/groups used.

groupVar A character indicating the name of the grouping variable.

control The control list used in the function’s call.

timeVar A character indicating the name of the time variable, if any.

fitted A vector of fitted values, if calc.bi=TRUE.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

References


See Also

lmmControl, update, predict.SMSN, residuals.SMSN, plot.SMSN, smn.lmm, smn.clmm
Examples

# simple example
dat1 = as.data.frame(nlme::Orthodont)
fm1 = smsn.lmm(dat1, formFixed=distance ~ age, groupVar="Subject",
control=lmmControl(max.iter=30))
fm1

# fitting for several distributions / dependence structures
fm1 = smsn.lmm(dat1, formFixed=distance ~ age+Sex, groupVar="Subject")
fm2 = smsn.lmm(dat1, formFixed=distance ~ age+Sex, groupVar="Subject",
  distr="st")
fm3 = smsn.lmm(dat1, formFixed=distance ~ age+Sex, groupVar="Subject",
  distr="ssl")
fm4 = smsn.lmm(dat1, formFixed=distance ~ age+Sex, groupVar="Subject",
  depStruct="ARp", pAR=1)
criteria(list(fm1=fm1, fm2=fm2, fm3=fm3, fm4=fm4))
summary(fm3)

# some diagnostic tools
plot(fm3)
acf3 = acfresid(fm3, calcCI=TRUE, MCiter=100)
plot(acf3)
plot(mahalDist(fm3), nlabels=2)
healy.plot(fm3)

---

### summary.SMN

**Summary of a smn.lmm object**

**Description**

summary method for class "SMN".

**Usage**

```r
## S3 method for class 'SMN'
summary(object, confint.level = 0.95, ...)
```

**Arguments**

- **object**: An object inheriting from class SMN, representing a fitted scale mixture normal linear mixed model.
- **confint.level**: Level of the approximate confidence intervals presented.
- **...**: Additional arguments.
Value

varRandom  Estimated variance matrix from random effects ($D$).
varFixed   Parameter estimates of variance from random errors ($\Sigma$). For recovering the error variance matrix use errorVar function.
tableFixed Estimated fixed effects, their standard errors and approximated confidence intervals.
criteria  Maximum log-likelihood value, AIC and BIC criteria.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos

See Also

boot_par, smn.lmm, errorVar, plot.SMN, residuals.SMN

Examples

fm1 = smn.lmm(distance ~ age+Sex, data=nlme::Orthodont, groupVar="Subject")
summary(fm1)

summary.SMNclmm  Summary of a smn.clmm object

Description

summary method for class "SMNclmm".

Usage

## S3 method for class 'SMNclmm'
summary(object, confint.level = 0.95, ...)

Arguments

object  An object inheriting from class SMNclmm, representing a fitted scale mixture normal linear mixed model with censored responses.
confint.level  Level of the approximate confidence intervals presented.
...  Additional arguments.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos, Victor H. Lachos and Katherine L. Valeriano
**summary.SMSN**

**See Also**

`smn.clmm`, `plot.SMNclmm`, `residuals.SMNclmm`

**Examples**

```r
nj1 = 5; m = 30
   time = rep(1:nj1, times=m)
groups = as.factor(rep(1:m, each=nj1))
dat1 = rsmsn.clmm(time, groups, cbind(1,time), rep(1,m*nj1), sigma2=0.7,
   D=0.5*diag(1), beta=c(1,2), depStruct="UNC")
fml = smn.clmm(dat1, formFixed=y~x, groupVar="ind", formRandom=~1,
   depStruct="UNC", ci="ci", lcl="lcl", ucl="ucl")
summary(fml)
```

---

**Description**

summary method for class "SMSN".

**Usage**

```r
## S3 method for class 'SMSN'
summary(object, confint.level = 0.95, ...)
```

**Arguments**

- `object`: An object inheriting from class SMSN, representing a fitted scale mixture skew-normal linear mixed model.
- `confint.level`: Level of the approximate confidence intervals presented.
- `...`: Additional arguments.

**Value**

- `varRandom`: Estimated variance matrix from random effects ($D$).
- `varFixed`: Parameter estimates of variance from random errors ($\Sigma$). For recovering the error variance matrix use `errorVar` function.
- `tableFixed`: Estimated fixed effects, their standard errors and approximated confidence intervals.
- `criteria`: Maximum log-likelihood value, AIC and BIC criteria.

**Author(s)**

Fernanda L. Schumacher, Larissa A. Matos and Victor H. Lachos
update

See Also

boot_par, smsn.lmm, errorVar, plot.SMN, residuals.SMN

Examples

```r
fm1 = smsn.lmm(distance ~ age+Sex, data=nlme::Orthodont, groupVar="Subject",
               control=lmmControl(tol=.0001))
summary(fm1)
```

Description

It fits a SM(S)N-(C)LMM by updating a fitted object.

Usage

```r
## S3 method for class 'SMN'
update(object, ..., evaluate = TRUE)
## S3 method for class 'SMSN'
update(object, ..., evaluate = TRUE)
## S3 method for class 'SMNclmm'
update(object, ..., evaluate = TRUE)
```

Arguments

- **object**: A `slnm.lmm`, `smn.lmm` or `smn.clmm` object containing the fitted model to be updated.
- **...**: Arguments to be changed.
- **evaluate**: A logical value indicating if the new class should be evaluated. If FALSE, the call is returned.

Value

An object resulting from the `slnm.lmm()`, `smn.lmm()` or `smn.clmm()` function.

Author(s)

Fernanda L. Schumacher, Larissa A. Matos, Victor H. Lachos and Katherine L. Valeriano

See Also

`slnm.lmm`, `smn.lmm`, `smn.clmm`
Examples

```r
fm1 = smm.lmm(nlme::Orthodont, formFixed=distance ~ age*Sex, groupVar="Subject")
fm2 = update(fm1, distr="t")
fm2
```
Index

* datagen
  rsmsn.clmm, 31
  rsmn.lmm, 32

* hplot
  plot, 17
  plot.acfresid, 18
  plot.mahalDist, 19
  plot.mahalDistCens, 20
  plot.SMNclmm, 21

* htest
  boot_ci, 4
  boot_par, 5
  lr.test, 14

* models
  acfresid, 3
  criteria, 6
  errorVar, 7
  fitted.SMN, 8
  fitted.SMNclmm, 9
  fitted.SMSN, 10
  healy.plot, 11
  lmmControl, 12
  mahalDist, 15
  mahalDistCens, 16
  predict.SMN, 22
  predict.SMNclmm, 23
  predict.SMSN, 24
  print.SMN, 25
  print.SMNclmm, 26
  print.SMSN, 27
  ranef, 27
  residuals, 28
  residuals.SMNclmm, 29
  smn.clmm, 34
  smn.lmm, 37
  smsn.lmm, 40
  summary.SMN, 43
  summary.SMNclmm, 44
  summary.SMSN, 45
  update, 46

* robust
  smn.clmm, 34
  smn.lmm, 37
  smsn.lmm, 40
  summary.SMSN, 45
  update, 46

acfresid, 3, 12, 18, 29
boot_ci, 4, 5
boot_par, 5, 5, 44, 46
criteria, 6
errorVar, 7, 44–46
fitted.SMN, 8, 17, 23, 28
fitted.SMNclmm, 9, 22, 24, 28
fitted.SMSN, 10, 17, 25, 28
ggplot, 12, 17–20, 22
healy.plot, 11, 29
lmmControl, 12, 36, 38, 39, 41, 42
lr.test, 14
mahalDist, 12, 15, 19, 29
mahalDistCens, 16, 20, 30
plot, 17
plot.acfresid, 4, 18
plot.mahalDist, 15, 19
plot.mahalDistCens, 16, 20
plot.SMN, 39, 44
plot.SMNclmm, 21, 36, 45
plot.SMSN, 42, 46
predict.SMN, 8, 22, 39
predict.SMNclmm, 9, 23, 36
predict.SMSN, 10, 24, 42
INDEX

print.SM, 25  
print.SMclmm, 26  
print.SMN, 27  

ranef, 27  
residuals, 28  
residuals.SM, 3, 17, 18, 39, 44  
residuals.SMclmm, 21, 22, 29, 36, 45  
residuals.SMN, 17, 18, 42, 46  
rsmsn.clmm, 31  
rsmsn.lmm, 32  

smn.clmm, 6, 9, 13, 16, 20, 22, 24, 26, 28, 30, 32, 34, 39, 42, 45, 46  
smn.lmm, 4–8, 12–15, 17–19, 23, 26, 28, 29, 36, 37, 42, 44, 46  
smsn.lmm, 4–7, 10, 12–15, 17–19, 25, 27–29, 33, 36, 39, 40, 46  
summary.SM, 26, 43  
summary.SMclmm, 26, 44  
summary.SMN, 27, 45  

update, 13, 36, 39, 42, 46