Package ‘nlive’

July 22, 2025

Type Package

Title Automated Estimation of Sigmoidal and Piecewise Linear Mixed
Models

Version 0.7.0

Date 2025-04-14

Maintainer Maude Wagner <maude_wagner@rush.edu>

Description Estimation of relatively complex nonlinear mixed-effects models, including the Sigmoidal Mixed Model and the Piecewise Linear Mixed Model with abrupt or smooth transition, through a single intuitive line of code and with automated generation of starting values.

License MIT + file LICENSE

URL https://github.com/MaudeWagner/nlive

BugReports https://github.com/MaudeWagner/nlive/issues

Encoding UTF-8

LazyLoad yes

LazyData yes

RoxygenNote 7.2.3

Depends R (>= 2.10)

Repository CRAN

Imports dplyr, nlraa, sqldf, ggplot2, graphics, lcmm, saemix, Rmisc,
sitar, Rmpfr, stats, knitr, fastDummies, viridis

Suggests rmarkdown, MASS, survival, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Maude Wagner [aut, cre] (ORCID:
<https://orcid.org/0000-0002-1834-3218>),
Ana W. Capuano [aut],
Emmanuelle Comets [ctb]

Date/Publication 2025-02-15 00:10:02 UTC

Contents

`dataCog` *Made-up longitudinal data on global cognition before death*

Description

The dataCog contains 1200 individuals with one line per visit. Repeated measures of global cognition (cognition) were collected over a maximum period of 30 years. Information on the age at death is also provided (ageDeath, ageDeath90).

Format

A data frame with 11867 observations over 1200 subjects and 5 variables:

ID subject identification number

time the retrospective (negative) time before death (in years)

cognition composite score of global cognition

ageDeath age at death (in years)

ageDeath90 age at death centered by the mean

Examples

```
summary(dataCog)
```

`nlive` *Automated Estimation of Sigmoidal and Piecewise Linear Mixed Models*

Description

The nlive() function allows to fit a Sigmoidal Mixed Model with 4 parameters, a Piecewise Linear Mixed Model with abrupt change, or a Piecewise Linear Mixed Model with a smooth polynomial transition in the context of longitudinal Gaussian outcomes. This function was designed to be intuitive enough to the less sophisticated users, while using recent developments such as the stochastic approximation expectation-maximization (SAEM) algorithm for efficient estimation. It was designed to optimize the initial values of the main parameters and help interpretation of the output by providing different features such as annotated outputs and graphs.

Usage

```
nlive(
  model,
  dataset,
  ID,
  outcome,
  time,
  predictor.all = NULL,
  predictor.par1 = NULL,
  predictor.par2 = NULL,
  predictor.par3 = NULL,
  predictor.par4 = NULL,
  start = NULL,
  plot.xlabel = NULL,
  plot.ylabel = NULL,
  traj.marg = FALSE,
  traj.marg.group = NULL,
  spag.plot.title = NULL,
  traj.marg.title = NULL,
  traj.marg.group.title = NULL,
  traj.marg.group.val = NULL
)
```

Arguments

`model` indicator of the model to fit (1=Sigmoidal Mixed Model, 2=Piecewise Mixed Model with abrupt change, 3=Piecewise Mixed Model with smooth transition)

`dataset` data frame containing the variables ID, outcome, time, predictor.all, and predictor.par1 to predictor.par4.

`ID` name of the variable representing the grouping structure specified with " (e.g., "ID" representing the unique identifier of participants).

`outcome` name of the time-varying variable representing the longitudinal outcome specified with " (e.g., "outcome").

`time` name of the variable representing the timescale specified with " (e.g., "time"), which can be negative or positive.

`predictor.all` optional vector indicating the name of the variable(s) that the four main parameters of the model of interest will be adjusted to (e.g. predictor.all=c("X1","X2")). Default to NULL.

`predictor.par1` optional vector indicating the name of the variable(s) that the first main parameter of the model of interest will be adjusted to (e.g. predictor.all=c("X1","X2")). For model 1, the first parameter = last level. For models 2 and 3, first parameter = intercept. Default to NULL.

`predictor.par2` optional vector indicating the name of the variable(s) that the second main parameter of the model of interest will be adjusted to (e.g. predictor.all=c("X1","X2")). For model 1, the second parameter = initial level. For models 2 and 3, second parameter = slope before the change-point. Default to NULL.

`predictor.par3` optional vector indicating the name of the variable(s) that the third main parameter of the model of interest will be adjusted to (e.g. predictor.all=c("X1","X2"). For model 1, the third parameter = midpoint. For models 2 and 3, third parameter = slope after the change-point. Default to NULL.

`predictor.par4` optional vector indicating the name of the variable(s) that the fourth main parameter of the model of interest will be adjusted to (e.g. predictor.all=c("X1","X2")). For model 1, the fourth parameter is the Hill slope. For models 2 and 3, the fourth parameter is the changepoint. Default to NULL.

`start` optional vector to override the specification of the four initial values for the main parameters. For model 1, the values must be included in the following order: last level, initial level, midpoint, Hill slope. For models 2 and 3, the values must be included in the following order: intercept, slope before the changepoint, slope after the changepoint, changepoint. Default to NULL.

`plot.xlabel` optional text for the title of the x-axis of all plots

`plot.ylabel` optional text for the title of the y-axis of all plots

`traj.marg` optional logical indicating if the marginal estimated trajectory should be plotted for the most common profile of covariates, if any. Default to FALSE.

`traj.marg.group`

optional name of the grouping variable listed in one of the predictor arguments to plot and contrast the estimated marginal trajectories between two specific groups, specified with " (e.g., traj.marg.group="X1"). If the variable is binary, the trajectories are contrasted between the two groups of interest. If the variable is continuous, the 10th and 90th percentile values will automatically be considered. The default value is NULL.

`spag.plot.title`

optional text for the title of the spaghetti plot

`traj.marg.title`

optional text for the title of the marginal estimated trajectory

`traj.marg.group.title`

optional text for the title of the marginal estimated trajectories contrasted between groups `traj.marg.group.val`

optional vector that can be used when *traj.marg.group* receives a quantitative variable and that allows to manually specify two percentile values to be considered for contrasting the traj.marg.group. The two values must be between 0 and 1 (e.g., traj.marg.group.val=c(0.2,0.8); for percentiles 20th and 80th). Default to 10th and 90th percentiles (i.e., traj.marg.group.val=c(0.1,0.9)).

Details

CAUTIONS REGARDING THE USE OF THE FUNCTION

traj.marg: if "TRUE", this argument automatically plots the estimated marginal trajectories of the longitudinal outcome for the most common profile of covariates, if any (i.e., ref "1" for binary variables and mean values for continuous variables). Thus, users must ensure that continuous variables are centered on the mean.

Value

An object of class SaemixObject (from the existing *saemix* R package) containing the results of the fit of the data by the non-linear mixed *model* of interest. The *nlive* function automatically provides (i) a spaghetti plot of the observed outcome for 70 randomly selected statistical units in the dataset and (ii) the standard *saemix* output, including the fixed effects estimates, the variance of random effects, and Likelihood of the fitted *model*. The outputs are printed on the terminal and the numerical and graphical outputs are stored in a directory.

Author(s)

Maude Wagner, Ana W. Capuano, Emmanuelle Comets

`<maude_wagner@rush.edu>`

References

Capuano AW, Wagner M. nlive: an R package to facilitate the application of the sigmoidal and random changepoint mixed models. BMC Medical Research Methodology. 2023;23(1):257. van den Hout A, Muniz-Terrera G, Matthews F. Smooth random change point models. Statistics in Medicine. 2011;30(6):599-610. Comets E, Lavenu A, Lavielle MM. Parameter estimation in nonlinear mixed effect models using saemix, an R implementation of the SAEM algorithm. Journal of Statistical Software. 2017;80(3):1-41.

See Also

`nlive.smm`, `nlive.pmma`, `nlive.pmms`

Examples

```
#### Fitting a sigmoidal mixed model - with no covariate
## Not run:
head(dataCog)
requireNamespace('nlraa')
smm.fit = nlive(model=1, dataset=dataCog, ID="ID", outcome="cognition", time="time")
```

```
## End(Not run)

#### plot(smm.fit): diagnostic plots to assess the goodness-of-fit of smm.fit
#### psi(smm.fit): estimates of individual parameters

#### Fitting a piecewise mixed model with abrupt change - with no covariate
## Not run:
pmm.abrupt.fit = nlive(model=2, dataset=dataCog, ID="ID", outcome="cognition", time="time")

## End(Not run)
#### plot(pmm.abrupt.fit): diagnostic plots to assess the goodness-of-fit of pmm.abrupt.fit
#### psi(pmm.abrupt.fit): estimates of individual parameters


#### Fitting a piecewise mixed model with smooth change - with all parameters
#### adjusted for ageDeath90. Here, the nlive() function will also provide a
#### plot of the estimated marginal trajectory in the whole study sample.
## Not run:
pmm.smooth.fit = nlive(model=3, dataset=dataCog, ID="ID", outcome="cognition", time="time",
predictor.all = c("ageDeath90"), traj.marg = TRUE)

## End(Not run)
#### plot(pmm.smooth.fit): diagnostic plots to assess the goodness-of-fit of the model
#### psi(pmm.smooth.fit): estimates of individual parameters
```

nlive.inspect *Generation of key plots for a longitudinal variable of interest*

Description

The nlive.inspect() function allows to generate basic graphs to describe the longitudinal observed measures of a variable of interest in the dataset

Usage

```
nlive.inspect(
  dataset,
  ID,
  variable,
  time,
  plot.xlabel = NULL,
  plot.ylabel = NULL,
  spag.plot.title = NULL
)
```

Arguments

`dataset` data frame containing the ID, variable, and time.

`ID` name of the variable representing the grouping structure specified with " (e.g., "ID" representing the unique identifier of participants).

`variable` name of the time-varying variable of interest specified with " (e.g., "variable").

`time` name of the variable representing the timescale specified with " (e.g., "time"). Can be negative or positive.

`plot.xlabel` optional text for the title of the x-axis of all plots.

`plot.ylabel` optional text for the title of the y-axis of all plots.

`spag.plot.title`
optional text for the title of the spaghetti plot.

Value

The *nlive.inspect* function automatically provides (i) an histogram of all the repeated measures of the variable available over time, (ii) a spaghetti plot of the longitudinal observed variable for 70 randomly selected statistical units, (iii) repeated boxplots of the longitudinal observed variable for each time unit. The outputs are printed on the terminal and the numerical and graphical outputs are stored in a directory

Author(s)

Maude Wagner, Ana W. Capuano, Emmanuelle Comets

`<maude_wagner@rush.edu>`

References

Capuano AW, Wagner M. nlive: an R package to facilitate the application of the sigmoidal and random changepoint mixed models. BMC Medical Research Methodology. 2023;23(1):257. Hadley Wickham (2016). ggplot2: elegant graphics for data analysis. Springer.

Examples

```
## Not run:
nlive.inspect(dataset=dataCog, ID="ID", variable="cognition", time="time")

## End(Not run)
```

nlive.pmma *Automated Estimation of the Piecewise Linear Mixed Model with Abrupt Change*

Description

The nlive.pmma() function allows to fit a Piecewise Linear Mixed Models with abrupt change in the context of longitudinal Gaussian outcomes. This function was designed to be intuitive enough to the less sophisticated users, while using recent developments such as the stochastic approximation expectation-maximization (SAEM) algorithm for efficient estimation. It was designed to optimize the initial values of the main parameters and help interpretation of the output by providing different features such as annotated outputs and graphs.

Usage

```
nlive.pmma(
  dataset,
  ID,
  outcome,
  time,
  var.all = NULL,
  var.last.level = NULL,
  var.slope1 = NULL,
  var.slope2 = NULL,
  var.changepoint = NULL,
  start = NULL,
  plot.xlabel = NULL,
  plot.ylabel = NULL,
  traj.marg = FALSE,
  traj.marg.group = NULL,
  traj.marg.title = NULL,
  traj.marg.group.title = NULL,
  traj.marg.group.val = NULL
)
```

Arguments

`dataset` data frame containing the variables ID, outcome, time, var.all, and all other var. arguments.

`ID` name of the variable representing the grouping structure specified with " (e.g., "ID" representing the unique identifier of participants).

`outcome` name of the time-varying variable representing the longitudinal outcome specified with " (e.g., "outcome").

`time` name of the variable representing the timescale specified with " (e.g., "time"), which can be negative or positive.

`var.all` optional vector indicating the name of the variable(s) that the four main parameters of the model will be adjusted to (e.g. var.all=c("X1","X2")). Default to NULL.

`var.last.level` optional vector indicating the name of the variable(s) that the last level parameter of the model of interest will be adjusted to (e.g. var.last.level=c("X1","X2")). Default to NULL.

`var.slope1` optional vector indicating the name of the variable(s) that the slope1 (before changepoint) parameter of the model of interest will be adjusted to (e.g. var.slope1=c("X1","X2"). Default to NULL.

`var.slope2` optional vector indicating the name of the variable(s) that the slope2 (after changepoint) parameter of the model of interest will be adjusted to (e.g. var.slope2=c("X1","X2"). Default to NULL.

`var.changepoint`

optional vector indicating the name of the variable(s) that the changepoint parameter of the model of interest will be adjusted to (e.g. var.changepoint=c("X1","X2")). Default to NULL.

`start` optional vector to override the specification of the four initial values for the main parameters - values must be included in the following order: intercept, slope before the changepoint, slope after the changepoint, changepoint. Default to NULL.

`plot.xlabel` optional text for the title of the x-axis of all plots.

`plot.ylabel` optional text for the title of the y-axis of all plots.

`traj.marg` optional logical indicating if the marginal estimated trajectory should be plotted for the most common profile of covariates, if any. Default to FALSE.

`traj.marg.group`

optional name of the grouping variable listed in one of the predictor arguments to plot and contrast the estimated marginal trajectories between two specific groups, specified with " (e.g., traj.marg.group="X1"). If the variable is binary, the trajectories are contrasted between the two groups of interest. If the variable is continuous, the 10th and 90th percentile values will automatically be considered. The default value is NULL.

`traj.marg.title`

optional text for the title of the marginal estimated trajectory

`traj.marg.group.title`

optional text for the title of the marginal estimated trajectories contrasted between groups

`traj.marg.group.val`

optional vector that can be used when *traj.marg.group* receives a quantitative variable and that allows to manually specify two percentile values to be considered for contrasting the traj.marg.group. The two values must be between 0 and 1 (e.g., traj.marg.group.val=c(0.2,0.8); for percentiles 20th and 80th). Default to 10th and 90th percentiles (i.e., traj.marg.group.val=c(0.1,0.9)).

Details

CAUTIONS REGARDING THE USE OF THE FUNCTION traj.marg: if "TRUE", this argument automatically plots the estimated marginal trajectories of the longitudinal outcome for the most common profile of covariates, if any (i.e., ref "1" for binary variables and mean values for continuous variables). Thus, users must ensure that continuous variables are centered on the mean.

Value

An object of class SaemixObject (from the existing *saemix* R package) containing the results of the fit of the data by the PMM-abrupt. The *nlive.pmma* function automatically provides the standard saemix output, including the fixed effects estimates, the variance of random effects, and Likelihood of the fitted model. The outputs are printed on the terminal and the numerical and graphical outputs are stored in a directory.

Author(s)

Maude Wagner, Ana W. Capuano, Emmanuelle Comets

`<maude_wagner@rush.edu>`

References

Capuano AW, Wagner M. nlive: an R package to facilitate the application of the sigmoidal and random changepoint mixed models. BMC Medical Research Methodology. 2023;23(1):257. van den Hout A, Muniz-Terrera G, Matthews F. Smooth random change point models. Statistics in Medicine. 2011;30(6):599-610. Comets E, Lavenu A, Lavielle MM. Parameter estimation in nonlinear mixed effect models using saemix, an R implementation of the SAEM algorithm. Journal of Statistical Software. 2017;80(3):1-41.

Examples

```
#### Fitting a piecewise mixed model with abrupt change - with no covariate
## Not run:
head(dataCog)
pmm.abrupt.fit = nlive.pmma(dataset=dataCog, ID=”ID”, outcome=”cognition”, time=”time”)

## End(Not run)
#### plot(pmm.abrupt.fit): diagnostic plots to assess the goodness-of-fit of pmm.abrupt.fit
#### psi(pmm.abrupt.fit): estimates of individual parameters
```

`nlive.pmms` *Automated Estimation of the Piecewise Linear Mixed Model with Smooth Change*

Description

The nlive.pmms() function allows to fit a Piecewise Linear Mixed Models with smooth change in the context of longitudinal Gaussian outcomes. This function was designed to be intuitive enough to the less sophisticated users, while using recent developments such as the stochastic approximation expectation-maximization (SAEM) algorithm for efficient estimation. It was designed to optimize the initial values of the main parameters and help interpretation of the output by providing different features such as annotated outputs and graphs.

Usage

```
nlive.pmms(
  dataset,
  ID,
  outcome,
  time,
  var.all = NULL,
  var.last.level = NULL,
  var.slope1 = NULL,
  var.slope2 = NULL,
  var.changepoint = NULL,
  start = NULL,
  plot.xlabel = NULL,
  plot.ylabel = NULL,
  traj.marg = FALSE,
  traj.marg.group = NULL,
  traj.marg.title = NULL,
  traj.marg.group.title = NULL,
  traj.marg.group.val = NULL
)
```

Arguments

`dataset` data frame containing the variables ID, outcome, time, var.all, and all other var. arguments.

`ID` name of the variable representing the grouping structure specified with " (e.g., "ID" representing the unique identifier of participants).

`outcome` name of the time-varying variable representing the longitudinal outcome specified with " (e.g., "outcome").

`time` name of the variable representing the timescale specified with " (e.g., "time"), which can be negative or positive.

`var.all` optional vector indicating the name of the variable(s) that the four main parameters of the model will be adjusted to (e.g. var.all=c("X1","X2")). Default to NULL.

`var.last.level` optional vector indicating the name of the variable(s) that the last level parameter of the model of interest will be adjusted to (e.g. var.last.level=c("X1","X2")). Default to NULL.

`var.slope1` optional vector indicating the name of the variable(s) that the slope1 (before changepoint) parameter of the model of interest will be adjusted to (e.g. var.slope1=c("X1","X2"). Default to NULL.

`var.slope2` optional vector indicating the name of the variable(s) that the slope2 (after changepoint) parameter of the model of interest will be adjusted to (e.g. var.slope2=c("X1","X2"). Default to NULL.

`var.changepoint`

optional vector indicating the name of the variable(s) that the changepoint parameter of the model of interest will be adjusted to (e.g. var.changepoint=c("X1","X2")). Default to NULL.

`start` optional vector to override the specification of the four initial values for the main parameters - values must be included in the following order: intercept, slope before the changepoint, slope after the changepoint, changepoint. Default to NULL.

`plot.xlabel` optional text for the title of the x-axis of all plots

`plot.ylabel` optional text for the title of the y-axis of all plots

`traj.marg` optional logical indicating if the marginal estimated trajectory should be plotted for the most common profile of covariates, if any. Default to FALSE.

`traj.marg.group`

optional name of the grouping variable listed in one of the predictor arguments to plot and contrast the estimated marginal trajectories between two specific groups, specified with " (e.g., traj.marg.group="X1"). If the variable is binary, the trajectories are contrasted between the two groups of interest. If the variable is continuous, the 10th and 90th percentile values will automatically be considered. The default value is NULL.

`traj.marg.title`

optional text for the title of the marginal estimated trajectory

`traj.marg.group.title`

optional text for the title of the marginal estimated trajectories contrasted between groups

`traj.marg.group.val`

optional vector that can be used when *traj.marg.group* receives a quantitative variable and that allows to manually specify two percentile values to be considered for contrasting the traj.marg.group. The two values must be between 0 and 1 (e.g., traj.marg.group.val=c(0.2,0.8); for percentiles 20th and 80th). Default to 10th and 90th percentiles (i.e., traj.marg.group.val=c(0.1,0.9)).

Details

CAUTIONS REGARDING THE USE OF THE FUNCTION

traj.marg: if "TRUE", this argument automatically plots the estimated marginal trajectories of the longitudinal outcome for the most common profile of covariates, if any (i.e., ref "1" for binary variables and mean values for continuous variables). Thus, users must ensure that continuous variables are centered on the mean.

Value

An object of class SaemixObject (from the existing *saemix* R package) containing the results of the fit of the data by the PMM-smooth. The *nlive.pmms* function automatically provides the standard saemix output, including the fixed effects estimates, the variance of random effects, and Likelihood of the fitted model. The outputs are printed on the terminal and the numerical and graphical outputs are stored in a directory.

Author(s)

Maude Wagner, Ana W. Capuano, Emmanuelle Comets

`<maude_wagner@rush.edu>`

References

Capuano AW, Wagner M. nlive: an R package to facilitate the application of the sigmoidal and random changepoint mixed models. BMC Medical Research Methodology. 2023;23(1):257. van den Hout A, Muniz-Terrera G, Matthews F. Smooth random change point models. Statistics in Medicine. 2011;30(6):599-610. Comets E, Lavenu A, Lavielle MM. Parameter estimation in nonlinear mixed effect models using saemix, an R implementation of the SAEM algorithm. Journal of Statistical Software. 2017;80(3):1-41.

Examples

```
#### Fitting a piecewise mixed model with abrupt change - with no covariate
## Not run:
head(dataCog)
pmm.smooth.fit = nlive.pmms(dataset=dataCog, ID="ID", outcome="cognition", time="time")

## End(Not run)
#### plot(pmm.smooth.fit): diagnostic plots to assess the goodness-of-fit of pmm.smooth.fit
#### psi(pmm.smooth.fit): estimates of individual parameters
```

`nlive.smm` *Automated Estimation of the Sigmoidal Mixed Model*

Description

The *nlive.smm()* function allows to fit a Sigmoidal Mixed Model with 4 parameters in the context of longitudinal Gaussian outcomes. This function was designed to be intuitive enough to the less sophisticated users, while using recent developments such as the stochastic approximation expectation-maximization (SAEM) algorithm for efficient estimation. It was designed to optimize the initial values of the main parameters and help interpretation of the output by providing different features such as annotated outputs and graphs.

Usage

```
nlive.smm(
  dataset,
  ID,
  outcome,
  time,
  var.all = NULL,
  var.first.level = NULL,
  var.last.level = NULL,
  var.midpoint = NULL,
  var.Hslope = NULL,
  start = NULL,
  plot.xlabel = NULL,
  plot.ylabel = NULL,
  traj.marg = FALSE,
  traj.marg.group = NULL,
  traj.marg.title = NULL,
  traj.marg.group.title = NULL,
  traj.marg.group.val = NULL
)
```

Arguments

`dataset` data frame containing the variables ID, outcome, time, var.all, and all other var. arguments.

`ID` name of the variable representing the grouping structure specified with " (e.g., "ID" representing the unique identifier of participants).

`outcome` name of the time-varying variable representing the longitudinal outcome specified with " (e.g., "outcome").

`time` name of the variable representing the timescale specified with " (e.g., "time"), which can be negative or positive.

`var.all` optional vector indicating the name of the variable(s) that the four main parameters of the model will be adjusted to (e.g. var.all=c("X1","X2")). Default to NULL.

`var.first.level` optional vector indicating the name of the variable(s) that the first level parameter of the model will be adjusted to (e.g. var.first.level=c("X1","X2")). Default to NULL.

`var.last.level` optional vector indicating the name of the variable(s) that the last level parameter of the model of interest will be adjusted to (e.g. var.last.level=c("X1","X2")). Default to NULL.

`var.midpoint` optional vector indicating the name of the variable(s) that the third main parameter of the model of interest will be adjusted to (e.g. var.midpoint=c("X1","X2"). Default to NULL.

`var.Hslope` optional vector indicating the name of the variable(s) that the fourth main parameter of the model of interest will be adjusted to (e.g. var.Hslope=c("X1","X2")). Default to NULL.

`start` optional vector to override the specification of the four initial values for the main parameters - values must be included in the following order: last level, initial level, midpoint, Hill slope. Default to NULL.

`plot.xlabel` optional text for the title of the x-axis of all plots

`plot.ylabel` optional text for the title of the y-axis of all plots

`traj.marg` optional logical indicating if the marginal estimated trajectory should be plotted for the most common profile of covariates, if any. Default to FALSE.

`traj.marg.group`
optional name of the grouping variable listed in one of the predictor arguments to plot and contrast the estimated marginal trajectories between two specific groups, specified with " (e.g., traj.marg.group="X1"). If the variable is binary, the trajectories are contrasted between the two groups of interest. If the variable is continuous, the 10th and 90th percentile values will automatically be considered. The default value is NULL.

`traj.marg.title`
optional text for the title of the marginal estimated trajectory

`traj.marg.group.title`
optional text for the title of the marginal estimated trajectories contrasted between groups

`traj.marg.group.val`
optional vector that can be used when *traj.marg.group* receives a quantitative variable and that allows to manually specify two percentile values to be considered for contrasting the traj.marg.group. The two values must be between 0 and 1 (e.g., traj.marg.group.val=c(0.2,0.8); for percentiles 20th and 80th). Default to 10th and 90th percentiles (i.e., traj.marg.group.val=c(0.1,0.9)).

Details

CAUTIONS REGARDING THE USE OF THE FUNCTION

traj.marg: if "TRUE", this argument automatically plots the estimated marginal trajectories of the longitudinal outcome for the most common profile of covariates, if any (i.e., ref "1" for binary variables and mean values for continuous variables). Thus, users must ensure that continuous variables are centered on the mean.

Value

An object of class SaemixObject (from the existing *saemix* R package) containing the results of the fit of the data by the SMM. The *nlive.smm* function automatically provides the standard saemix output, including the fixed effects estimates, the variance of random effects, and Likelihood of the fitted model. The outputs are printed on the terminal and the numerical and graphical outputs are stored in a directory.

Author(s)

Maude Wagner, Ana W. Capuano, Emmanuelle Comets

`<maude_wagner@rush.edu>`

References

Capuano AW, Wagner M. nlive: an R package to facilitate the application of the sigmoidal and random changepoint mixed models. BMC Medical Research Methodology. 2023;23(1):257. van den Hout A, Muniz-Terrera G, Matthews F. Smooth random change point models. Statistics in Medicine. 2011;30(6):599-610. Comets E, Lavenu A, Lavielle MM. Parameter estimation in nonlinear mixed effect models using saemix, an R implementation of the SAEM algorithm. Journal of Statistical Software. 2017;80(3):1-41.

Examples

```
#### Fitting a sigmoidal mixed model - with no covariate
## Not run:
head(dataCog)
requireNamespace('nlraa')
smm.fit = nlive.smm(dataset=dataCog, ID="ID", outcome="cognition", time="time")

## End(Not run)
#### plot(smm.fit): diagnostic plots to assess the goodness-of-fit of smm.fit
#### psi(smm.fit): estimates of individual parameters
```

`nlive.splines` *Automated Estimation of the Linear Mixed Model with Splines#'*

Description

The nlive.splines function allows to fit a Linear Mixed Models with the function of time approximated with natural cubic splines or B-splines in the context of longitudinal Gaussian outcomes. This function was designed to be intuitive enough to the less sophisticated users, while using the existing hlme() function from the lcmm R package as well as the existing ns() and bs() functions from the splines R package. #' CAUTIONS REGARDING THE USE OF THE FUNCTION

Usage

```
nlive.splines(
  dataset,
  ID,
  time,
  formula,
  random,
  splines = NULL,
  knots = NULL,
  Boundary.knots = NULL,
  traj.marg = FALSE
)
```

Arguments

`dataset` data frame containing the variables ID, outcome, time, var.all, and all other var. arguments.

`ID` name of the variable representing the grouping structure specified with " (e.g., "ID" representing the unique identifier of participants).

`time` name of the variable representing the timescale specified with " (e.g., "time"), which can be negative or positive.

`formula` two-sided linear formula object for the fixed-effects in the linear mixed model. The response outcome is on the left of ~ and the covariates are separated by + on the right of ~. By default, an intercept is included. If no intercept, -1 should be the first term included on the right of ~.

`random` optional one-sided formula for the random-effects in the linear mixed model. Covariates with a random-effect are separated by +. Default is an intercept. If no intercept, -1 should be the first term included.

`splines` "ns" for Natural Cubic Splines. "bs" for Cubic B-splines.

`knots` inner knots that define the spline. Typical values are the mean or median for one knot, quantiles for more knots. Default is two equally spaced inner knots (i.e., 33th, 66th percentiles).

`Boundary.knots` boundary points at which to impose the natural boundary conditions and anchor the spline basis. Default to the range of the data.

`traj.marg` optional logical indicating if the marginal estimated trajectory should be plotted for the most common profile of covariates, if any. Default to FALSE.

#' @return An object of from the existing *lcmm* R package containing the results of the fit of the data by a linear mixed model and with the function of time approximated using natural cubic splines.

Details

traj.marg: if "TRUE", this argument automatically plots the estimated marginal trajectories of the longitudinal outcome for the most common profile of covariates, if any (i.e., ref "1" for binary variables and mean values for continuous variables). Thus, users must ensure that continuous variables are centered on the mean.

Author(s)

Maude Wagner, Ana W. Capuano, Cecile Proust-lima

`<maude_wagner@rush.edu>`

References

Proust-Lima, C., Philipps, V., & Liquet, B. (2017). Estimation of Extended Mixed Models Using Latent Classes and Latent Processes: The R Package lcmm. Journal of Statistical Software, 78(2), 1–56. https://doi.org/10.18637/jss.v078.i02 Perperoglou, A., Sauerbrei, W., Abrahamowicz, M. et al. A review of spline function procedures in R. BMC Med Res Methodol 19, 46 (2019). https://doi.org/10.1186/s12874-019-0666-3

Examples

```
#### Fitting a linear mixed model with ns splines and 2 inner knots (33th, 66th percentiles)
## Not run:
head(dataCog)
lmm.ns.fit = nlive.splines(dataCog, ID="ID", outcome="cognition", time="time", splines.type="ns")

## End(Not run)
#### plot(lmm.ns.fit): diagnostic plots to assess the goodness-of-fit of lmm.ns.fit
```

Index