

Package ‘nlive’

October 13, 2022

Type Package

Title Automated Estimation of Sigmoidal and Piecewise Linear Mixed Models

Version 0.1.0

Date 2022-09-12

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>

Encoding UTF-8

LazyLoad yes

LazyData yes

RoxygenNote 7.1.1

Depends R (>= 2.10)

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

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

Config/testthat/edition 3

NeedsCompilation no

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

Repository CRAN

Date/Publication 2022-09-13 07:30:02 UTC

R topics documented:

dataset	2
nlive	2

dataset	<i>Made-up longitudinal data on global cognition before death</i>
---------	---

Description

The dataset 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(dataset)
```

nlive	<i>Automated Estimation of Sigmoidal and Piecewise Linear Mixed Models</i>
-------	--

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. This data frame needs to be named "dataset" (see example below).
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"). Can be negative or positive. Note that model 1, SMM, will always report a positive value under the midpoint parameter (e.g. if the time in the data goes from 0 down to -10 and the time of the midpoint is -2, the model will report a midpoint of 2).
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.

<code>predictor.par2</code>	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. <code>predictor.all=c("X1","X2")</code>). For model 1, the second parameter = initial level. For models 2 and 3, second parameter = slope before the change-point. Default to NULL.
<code>predictor.par3</code>	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. <code>predictor.all=c("X1","X2")</code>). For model 1, the third parameter = midpoint. For models 2 and 3, third parameter = slope after the change-point. Default to NULL.
<code>predictor.par4</code>	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. <code>predictor.all=c("X1","X2")</code>). For model 1, the fourth parameter is the Hill slope. For models 2 and 3, the fourth parameter is the changepoint. Default to NULL.
<code>start</code>	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.
<code>plot.xlabel</code>	optional text for the title of the x-axis of all plots
<code>plot.ylabel</code>	optional text for the title of the y-axis of all plots
<code>traj.marg</code>	optional logical indicating if the marginal estimated trajectory should be plotted for the most common profile of covariates, if any. Default to FALSE.
<code>traj.marg.group</code>	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., <code>traj.marg.group="X1"</code>). 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.
<code>spag.plot.title</code>	optional text for the title of the spaghetti plot
<code>traj.marg.title</code>	optional text for the title of the marginal estimated trajectory
<code>traj.marg.group.title</code>	optional text for the title of the marginal estimated trajectories contrasted between groups
<code>traj.marg.group.val</code>	optional vector that can be used when <code>traj.marg.group</code> receives a quantitative variable and that allows to manually specify two percentile values to be considered for contrasting the <code>traj.marg.group</code> . The two values must be between 0 and 1 (e.g., <code>traj.marg.group.val=c(0.2,0.8)</code> ; for percentiles 20th and 80th). Default to 10th and 90th percentiles (i.e., <code>traj.marg.group.val=c(0.1,0.9)</code>).

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

The Sigmoidal mixed model formulation and computational methods are described in Capuano A, Wilson R, Leurgans S, Dawson J, Bennett D, Hedecker D (2018). Sigmoidal mixed models for longitudinal data. *Statistical Methods in Medical Research*, 27(3):863-875. The Piecewise linear mixed model with abrupt and polynomial smooth change are described in van den Hout A, Muniz-Terrera G, Matthews F (2011). Smooth random change point models. *Statistics in Medicine*, 30(6):599-610. The *saemix* R package and SAEM algorithm are described in Comets E, Lavenu A, Lavielle MM (2017). Parameter estimation in nonlinear mixed effect models using *saemix*, an R implementation of the SAEM algorithm. *Journal of Statistical Software*, 80(3):1-41.

Examples

```
#### Fitting a piecewise mixed model with abrupt change - with no covariate

head(dataset)
pmm.abrupt.fit = nlive(model=2, dataset=dataset, ID="ID", outcome="cognition", time="time")

#### plot(pmm.abrupt.fit) to obtain the diagnostic plots to assess the
#### goodness-of-fit of pmm.abrupt.fit
#### psi(pmm.abrupt.fit): to obtain the 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.

pmm.smooth.fit = nlive(model=3, dataset=dataset, ID="ID", outcome="cognition", time="time",
predictor.all = c("ageDeath90"), traj.marg = TRUE)

#### plot(pmm.smooth.fit): to obtain the diagnostic plots to assess the
```

```
#### goodness-of-fit of the model  
#### psi(pmm.smooth.fit): to obtain the estimates of individual parameters
```

Index

* **dataset**
dataset, 2

dataset, 2

nlive, 2