Package: FlexVarJM (via r-universe)

October 10, 2026

Title Estimate Joint Models with Subject-Specific Variance

Version 0.1.0

Description Estimation of mixed models including a subject-specific variance which can be time and covariate dependent. In the joint model framework, the package handles left truncation and allows a flexible dependence structure between the competing events and the longitudinal marker. The estimation is performed under the frequentist framework, using the Marquardt-Levenberg algorithm. (Courcoul, Tzourio, Woodward, Barbieri, Jacqmin-Gadda (2023) <doi:10.48550/arXiv.2306.16785>).

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Imports ggplot2, lcmm, marqLevAlg, mvtnorm, randtoolbox, Rcpp, stats, survminer, utils

Depends R (>= 3.5.0), splines, survival

URL https://github.com/LeonieCourcoul/FlexVarJM

BugReports https://github.com/LeonieCourcoul/FlexVarJM/issues

LazyData true

Suggests knitr, rmarkdown

VignetteBuilder knitr

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Author Léonie Courcoul [aut, cre], Antoine Barbieri [aut], Hélène Jacqmin-Gadda [aut]

Maintainer Léonie Courcoul <leonie.courcoul@u-bordeaux.fr>

Repository https://cran.r-universe.dev

Date/Publication 2023-11-21 02:40:27 UTC

RemoteUrl https://github.com/cran/FlexVarJM

RemoteRef HEAD

RemoteSha ae1340b8596beba46a7e3d748ba7daa79204c44f

Contents

data.GaussKronrod . . . 2
data.GaussKronrod2 . . . 3
data.manag.long . . . 4
data.manag.surv . . . 4
data.time . . . 5
Data_toy . . . 6
gaussKronrod . . . 6
goodness_of_fit . . . 7
initial.long . . . 8
log_llh . . . 9
log_llh_rcpp . . . 12
lsjm . . . 16
lsmm . . . 20
pred_s.t.bootstrap.tps . . . 22
pred_s.t.ponctuel.tps . . . 23
predyn . . . 23

Index **26**

`data.GaussKronrod` *Initialisation of Survival Data at Gauss Kronrod time points*

Description

Initialisation of Survival Data at Gauss Kronrod time points

Usage

```
data.GaussKronrod(data.id, Time, k = 15)
```

Arguments

`data.id`	A database with covariates of interest and 1 line per subject
`Time`	A vector of time event
`k`	The number of Gauss Kronrod points, by default k = 15

Value

A list with the following components :

`K` an integer, the number of points

`P` a vector, of value Time/2

`st` a matrix with nrow = number of subjects and ncol = k. The new time to compute the survival function

`wk` a vector of weights

`data.id2` a database with K lines per subjects

`id.GK` the vector of IDs

`data.GaussKronrod2` *Initialisation of Survival Data at Gauss Kronrod time points 2*

Description

Initialisation of Survival Data at Gauss Kronrod time points 2

Usage

```
data.GaussKronrod2(data.id, a, b, k = 15)
```

Arguments

`data.id`	A database with covariates of interest and 1 line per subject
`a`	First born
`b`	Second born
`k`	The number of Gauss Kronrod points, by default k = 15

Value

A list with the following components :

`K` an integer, the number of points

`P` a vector, of value Time/2

`st` a matrix with nrow = number of subjects and ncol = k. The new time to compute the survival function

`wk` a vector of weights

`data.id2` a database with K lines per subjects

`id.GK` the vector of IDs

data.manag.long *Management of longitudinal data*

Description

Management of longitudinal data

Usage

```
data.manag.long(formGroup, formFixed, formRandom, data.long1)
```

Arguments

formGroup	A formula which indicates the group variable
formFixed	A formula which indicates the fixed effects for the longitudinal submodel
formRandom	A formula which indicates the random effects for the longitudinal submodel
data.long1	A dataframe with the longitudinal data

Value

A list with the following components :

data_long a clean dataframe for the longitudinal data

y.new.prog the vector of responses variable

X a matrix with the fixed effects

U a matrix with the random effects

id a vector with the identification of individuals

offset a vector with the number of measurements for each individual

I an integer, the number of individuals

data.manag.surv *Management of survival data*

Description

Management of survival data

Usage

```
data.manag.surv(formGroup, formSurv, data.long1, formSurv_CompRisk)
```

Arguments

formGroup	A formula which indicates the group variable
formSurv	A formula which indicates the variables used in the survival submodel
data.long1	Database
formSurv_CompRisk	A formula which indicates the variables used in the competing survival submodel

Value

A list with the following components :

tmp the final database for survival analysis

Time a vector of observed times

event1 a vector of first event indicator

nTime length of Time vector

Z matrix of covariables of first survival submodel

event2 a vector of second event indicator

Z_CR matrix of covariables of second survival submodel

---

data.time *Management of data for longitudinal submodel*

Description

Management of data for longitudinal submodel

Usage

```
data.time(data.id, Time, formFixed, formRandom, timeVar)
```

Arguments

data.id	A dataframe
Time	A vector of Time of events
formFixed	A formula for the fixed effects of the longitudinal submodel
formRandom	A formula for the random effects of the longitudinal submodel
timeVar	The name of the column of time in data.id. This variable must appears in data.id

Value

A list with the following components

Xtime a matrix of fixed effects at each time of measure

Utime a matrix of random effects at each time of measure

Data_toy *Data_toy*

Description

A simulated dataset for the example of the FlexVarJM package.

Usage

```
Data_toy
```

Format

A data frame with 2076 rows and 5 variables:

ID the Id of each subject

visit the time of measurement

y the value of the marker

time the time of the first event observed

event the first event observed : 0 = censoring, 1 = first event, 2 = second event

gaussKronrod *Gauss-Kronrod nodes and weights*

Description

Gauss-Kronrod nodes and weights

Usage

```
gaussKronrod(k = 15)
```

Arguments

k	the number of points for Gauss-Kronrod approximation : choice between 7 and 15. 15 by default.

Value

A list with the following components :

`sk` A k-vector of nodes

`wk` A k-vector of weights

goodness_of_fit *Predictions for the goodness of fit, of the random effects, the current value for each individuals and the cumulative hazard function for both events*

Description

Predictions for the goodness of fit, of the random effects, the current value for each individuals and the cumulative hazard function for both events

Usage

```
goodness_of_fit(object, graph = FALSE, break.times = NULL)
```

Arguments

object	an object of class lsjm
graph	a boolean to indicate to print graphics, False by default
break.times	a vector of times for the time points of longitudinal graphic

Value

A list which contains the following elements :

tables A list with the table of the predicted random effect, the table of the predicted current value, table(s) of predictive cumulative hazard function(s)

graphs A list with 2 or 3 graphs : one for the longitudinal adjustment and one for each risk function

Examples

```
#Fit a joint model with competing risks and subject-specific variability
example <- lsjm(formFixed = y~visit,
formRandom = ~ visit,
formGroup = ~ID,
formSurv = Surv(time, event ==1 ) ~ 1,
timeVar = "visit",
data.long = Data_toy,
variability_hetero = TRUE,
formFixedVar =~visit,
formRandomVar =~visit,
correlated_re = TRUE,
sharedtype = c("current value", "variability"),
hazard_baseline = "Weibull",
formSlopeFixed =~1,
formSlopeRandom = ~1,
```

```
indices_beta_slope = c(2),
competing_risk = TRUE,
formSurv_CR = Surv(time, event ==2 ) ~ 1,
hazard_baseline_CR = "Weibull",
sharedtype_CR = c("current value", "variability"),
S1 = 100,
S2 = 1000,
nproc = 1,
maxiter = 100,
Comp.Rcpp = TRUE
)

#Assesment of the goodness of fit:
gof <- goodness_of_fit(example, graph = TRUE)
gof$tables
gof$graphs
```

`initial.long` *Initialisation of Longitudinal Submodel*

Description

Initialisation of Longitudinal Submodel

Usage

```
initial.long(formFixed, formRandom, idVar, data.long1, ncX, nproc = nproc)
```

Arguments

`formFixed`	A formula which indicates the fixed effects for the longitudinal submodel
`formRandom`	A formula which indicates the random effects for the longitudinal submodel
`idVar`	A character, indicates the name of the group variable
`data.long1`	A dataframe with the longitudinal data
`ncX`	An integer, the number of columns of matrix X, ie, the number of fixed effects
`nproc`	An integer, the number of cores for parallel computation

Value

A list with the following components :

`long_model` the result of the hlme function

`priorMean.beta` the estimated parameters for fixed effects in the linear mixed effects model

`sigma` the estimated sigma of the model

log_llh *Log-likelihood computation*

Description

Log-likelihood computation

Usage

```
log_llh(
  param,
  nb.e.a,
  nb.priorMean.beta,
  nb.alpha,
  competing_risk,
  nb.alpha.CR,
  variability_hetero,
  S,
  Zq,
  sharedtype,
  sharedtype_CR,
  hazard_baseline,
  hazard_baseline_CR,
  ord.splines,
  Xtime,
  Utime,
  nb_pointsGK,
  Xs,
  Us,
  Xslope,
  Uslope,
  Xs.slope,
  Us.slope,
  indices_beta_slope,
  Time,
  st_calc,
  B,
  Bs,
  wk,
  Z,
  P,
  left_trunc,
  Z_CR,
  X_base,
  offset,
  U,
  y.new.prog,
```

```
  event1,
  event2,
  Ind,
  Xs.0,
  Us.0,
  Xs.slope.0,
  Us.slope.0,
  P.0,
  st.0,
  Bs.0,
  B.CR,
  Bs.CR,
  Bs.0.CR,
  nb.e.a.sigma = nb.e.a.sigma,
  nb.omega = nb.omega,
  Otime = Otime,
  Wtime = Wtime,
  Os = Os,
  Ws = Ws,
  O_base = O_base,
  W_base = W_base,
  correlated_re = correlated_re,
  Os.0 = Os.0,
  Ws.0 = Ws.0
)
```

Arguments

param	a vector : paramaters to be estimated
nb.e.a	integer : number of RE
nb.priorMean.beta	integer : number of fixed effects
nb.alpha	integer : number of covariates in survival model
competing_risk	boolean : allow competing risk or not, FALSE by default
nb.alpha.CR	integer : number of covariates in survival model for competing risks
variability_hetero	boolean : allow the heterogeneous variability or not
S	integer : the number of QMC points
Zq	vector : sobol points
sharedtype	vector : dependence structure for survival model : "RE" (random effects) or "CV" (current value) or "CVS" (current value and slope) or "S" (slope)
sharedtype_CR	vector : dependence structure for competing risk survival model : "RE" (random effects) or "CV" (current value) or "CVS" (current value and slope) or "S" (slope)
hazard_baseline	char : baseline hazard function : "Exponential" or "Weibull" or "Splines"

`hazard_baseline_CR`
: char : baseline hazard function, competing risk : "Exponential" or "Weibull" or "Splines"

`ord.splines`
: integer : the order of splines function for baseline hazard function

`Xtime`
: matrix : fixed effects at event time

`Utime`
: matrix : RE at event time

`nb_pointsGK`
: integer : number of points for Gauss-Kronrod approximation, 7 or 15 (default)

`Xs`
: matrix : fixed effects at Gauss-Kronrod times

`Us`
: matrix : RE at Gauss-Kronrod times

`Xslope`
: matrix : fixed effects of slope at event times

`Uslope`
: matrix : RE of slope at event times

`Xs.slope`
: matrix : fixed effects of slope at Gauss-Kronrod times

`Us.slope`
: matrix : RE of slope at Gauss-Kronrod times

`indices_beta_slope`
: vector : position of beta which will be used in the slope computation

`Time`
: vector : observed event times

`st_calc`
: matrix : Gauss-Kronrod times

`B`
: matrix : splines for baseline hazard function of event 1

`Bs`
: matrix : splines for baseline survival function of event 1

`wk`
: vector : Gauss-Kronrod weights

`Z`
: matrix : covariates for survival function of event 1

`P`
: vector : Time/2

`left_trunc`
: boolean : left truncation indicator

`Z_CR`
: matrix : covariates for survival function of event 2

`X_base`
: matrix : fixed effects for longitudinal submodel

`offset`
: vector : number of lines per subjects

`U`
: matrix : RE for longitudinal submodel

`y.new.prog`
: vector : y measures for longitudinal submodel

`event1`
: vector : event 1 indicator

`event2`
: vector : event 2 indicator

`Ind`
: integer : number of subjects

`Xs.0`
: same for left truncation

`Us.0`
: same for left truncation

`Xs.slope.0`
: same for left truncation

`Us.slope.0`
: same for left truncation

`P.0`
: same for left truncation

`st.0`
: same for left truncation

`Bs.0`
: same for left truncation

B.CR	same for left truncation
Bs.CR	same for left truncation
Bs.0.CR	same for left truncation
nb.e.a.sigma	integer : number of RE for variability
nb.omega	integer : number of fixed effects for variability
Otime	matrix : fixed effects of variability at event time
Wtime	matrix : RE of variability at event time
Os	matrix : fixed effects of variability at Gauss-Kronrod times
Ws	matrix : random effects of variability at Gauss-Kronrod times
O_base	matrix : fixed effects for variability
W_base	matrix : fixed effects for variability
correlated_re	boolean : indicator to allow all the random effects to be correlated
Os.0	matrix : same for left truncation
Ws.0	matrix : same for left truncation

Value

The value of the log-likelihood

`log_llh_rcpp` *Log-likelihood computation in RCPP*

Description

Log-likelihood computation in RCPP

Usage

```
log_llh_rcpp(
  param,
  nb.e.a,
  nb.priorMean.beta,
  nb.alpha,
  competing_risk,
  nb.alpha.CR,
  variability_hetero,
  S,
  Zq,
  sharedtype,
  sharedtype_CR,
  hazard_baseline,
  hazard_baseline_CR,
  ord.splines,
```

```
    Xtime,
    Utime,
    nb_pointsGK,
    Xs,
    Us,
    Xslope,
    Uslope,
    Xs.slope,
    Us.slope,
    indices_beta_slope,
    Time,
    st_calc,
    B,
    Bs,
    wk,
    Z,
    P,
    left_trunc,
    Z_CR,
    X_base,
    offset,
    U,
    y.new.prog,
    event1,
    event2,
    Ind,
    Xs.0,
    Us.0,
    Xs.slope.0,
    Us.slope.0,
    P.0,
    st.0,
    Bs.0,
    B.CR,
    Bs.CR,
    Bs.0.CR,
    nb.e.a.sigma = nb.e.a.sigma,
    nb.omega = nb.omega,
    Otime = Otime,
    Wtime = Wtime,
    Os = Os,
    Ws = Ws,
    O_base = O_base,
    W_base = W_base,
    correlated_re = correlated_re,
    Os.0 = Os.0,
    Ws.0 = Ws.0
  )
```

Arguments

param	a vector : paramaters to be estimated
nb.e.a	integer : number of RE
nb.priorMean.beta	integer : number of fixed effects
nb.alpha	integer : number of covariates in survival model
competing_risk	boolean : allow competing risk or not, FALSE by default
nb.alpha.CR	integer : number of covariates in survival model for competing risks
variability_hetero	boolean : allow the heterogeneous variability or not
S	integer : the number of QMC points
Zq	vector : sobol points
sharedtype	vector : dependence structure for survival model : "RE" (random effects) or "CV" (current value) or "CVS" (current value and slope) or "S" (slope)
sharedtype_CR	vector : dependence structure for competing risk survival model : "RE" (random effects) or "CV" (current value) or "CVS" (current value and slope) or "S" (slope)
hazard_baseline	char : baseline hazard function : "Exponential" or "Weibull" or "Splines"
hazard_baseline_CR	char : baseline hazard function, competing risk : "Exponential" or "Weibull" or "Splines"
ord.splines	integer : the order of splines function for baseline hazard function
Xtime	matrix : fixed effects at event time
Utime	matrix : RE at event time
nb_pointsGK	integer : number of points for Gauss-Kronrod approximation, 7 or 15 (default)
Xs	matrix : fixed effects at Gauss-Kronrod times
Us	matrix : RE at Gauss-Kronrod times
Xslope	matrix : fixed effects of slope at event times
Uslope	matrix : RE of slope at event times
Xs.slope	matrix : fixed effects of slope at Gauss-Kronrod times
Us.slope	matrix : RE of slope at Gauss-Kronrod times
indices_beta_slope	vector : position of beta which will be used in the slope computation
Time	vector : observed event times
st_calc	matrix : Gauss-Kronrod times
B	matrix : splines for baseline hazard function of event 1
Bs	matrix : splines for baseline survival function of event 1
wk	vector : Gauss-Kronrod weights

Z	matrix : covariates for survival function of event 1
P	vector : Time/2
left_trunc	boolean : left truncation indicator
Z_CR	matrix : covariates for survival function of event 2
X_base	matrix : fixed effects for longitudinal submodel
offset	vector : number of lines per subjects
U	matrix : RE for longitudinal submodel
y.new.prog	vector : y measures for longitudinal submodel
event1	vector : event 1 indicator
event2	vector : event 2 indicator
Ind	integer : number of subjects
Xs.0	same for left truncation
Us.0	same for left truncation
Xs.slope.0	same for left truncation
Us.slope.0	same for left truncation
P.0	same for left truncation
st.0	same for left truncation
Bs.0	same for left truncation
B.CR	same for left truncation
Bs.CR	same for left truncation
Bs.0.CR	same for left truncation
nb.e.a.sigma	integer : number of RE for variability
nb.omega	integer : number of fixed effects for variability
Otime	matrix : fixed effects of variability at event time
Wtime	matrix : RE of variability at event time
Os	matrix : fixed effects of variability at Gauss-Kronrod times
Ws	matrix : random effects of variability at Gauss-Kronrod times
O_base	matrix : fixed effects for variability
W_base	matrix : fixed effects for variability
correlated_re	boolean : indicator to allow all the random effects to be correlated
Os.0	matrix : same for left truncation
Ws.0	matrix : same for left truncation

Value

The value of the log-likelihood

`lsjm` *lsjm : Estimation of joint model for longitudinal data with a subject-specific time-dependent variability and time-to-event data.*

Description

This function fits complex joint models with shared random effects. The longitudinal submodel estimates longitudinal data with a mixed-effects model in which we suppose that the variance of the residual error is time-dependent and subject-specific. The survival submodel handles right-censored and left-truncated time-to-event data and competing risks. The dependence structure between the longitudinal and the survival data can be the random effects from the mixed model or the current value of the marker and/or the slope of the marker. We can also adjust on the current variance of the marker. (See below) Parameters are estimated simultaneously through a maximum likelihood method, using a Marquardt-Levenberg algorithm.

Usage

```
lsjm(
  formFixed,
  formRandom,
  formGroup,
  formSurv,
  timeVar,
  data.long,
  variability_hetero = TRUE,
  formFixedVar,
  formRandomVar,
  correlated_re = FALSE,
  sharedtype = c("current value", "variability"),
  hazard_baseline = "Exponential",
  formSlopeFixed = NULL,
  formSlopeRandom = NULL,
  indices_beta_slope = NULL,
  nb_pointsGK = 15,
  ord.splines = 3,
  competing_risk = FALSE,
  formSurv_CR = NULL,
  hazard_baseline_CR = "Exponential",
  sharedtype_CR = c("current value", "variability"),
  left_trunc = FALSE,
  Time.0 = NULL,
  S1 = 1000,
  S2 = 5000,
  nproc = 1,
  clustertype = "SOCK",
  maxiter = 100,
  print.info = FALSE,
```

```
  file = NULL,
  epsa = 0.001,
  epsb = 0.001,
  epsd = 0.001,
  binit = NULL,
  Comp.Rcpp = TRUE
)
```

Arguments

formFixed	A formula for the fixed effects of the longitudinal submodel
formRandom	A formula for the random effects of the longitudinal submodel
formGroup	A formula which indicates the group variable
formSurv	A formula which indicates the variables used in the survival submodel
timeVar	The name of the column of time in data.long. This variable must appears in data.long
data.long	A dataframe with the longitudinal data
variability_hetero	A logical to indicate if we suppose a subject_specific variability
formFixedVar	A formula for the fixed effects of the variance predictor
formRandomVar	A formula for the random effects of the variance predictor
correlated_re	A logical to indicate if the random effects of the marker and the variance predictors are correlated (By default there are supposed to be independent)
sharedtype	char : dependence structure for survival model : "RE" (random effects) or "CV" (current value) or "CVS" (current value and slope) or "S" (slope)
hazard_baseline	char : baseline hazard function : "Exponential" or "Weibull" or "Splines"
formSlopeFixed	A formula for the fixed effects of the slope of the longitudinal submodel : the derivative of the formFixed
formSlopeRandom	A formula for the random effects of the slope of the longitudinal submodel : the derivative of the formRandom
indices_beta_slope	A vector of index indicating which beta of the formFixed formula is used in the formSlopeFixed formula
nb_pointsGK	the number of points for Gauss-Kronrod approximation : choice between 7 and 15. 15 by default.
ord.splines	A numeric, the order of splines for the baseline risk function (3 by default)
competing_risk	A logical indicating if the model handles with competing risks
formSurv_CR	In case of competing risk A formula which indicates the variables used in the survival submodel for the second event
hazard_baseline_CR	In case of competing risk : a character for the baseline hazard function of the second event

sharedtype_CR	In case of competing risk ; a character for the dependence structure
left_trunc	A logical indicating if the model handles with left truncated data
Time.0	In case of left truncation : a vector of entry times
S1	An integer : the number of QMC draws for the first step
S2	An integer : the number of QMC draws for the second step
nproc	An integer : the number of processors for parallel computing
clustertype	one of the supported types from makeCluster function
maxiter	optional maximum number of iterations for the marqLevAlg iterative algorithm.
print.info	logical indicating if the outputs of each iteration should be written
file	optional character giving the name of the file where the outputs of each iteration should be written (if print.info=TRUE)
epsa	optional threshold for the convergence criterion based on the parameter stability.
epsb	optional threshold for the convergence criterion based on the objective function stability.
epsd	optional threshold for the relative distance to maximum. This criterion has the nice interpretation of estimating the ratio of the approximation error over the statistical error, thus it can be used for stopping the iterative process whatever the problem.
binit	optional initials parameters.
Comp.Rcpp	boolean to indicate if the computation is performed with RCPP program or R program. True by default.

Details

A. LONGITUDINAL SUBMODEL

The longitudinal submodel is defined by a linear mixed effects model with the residual variance which could be supposed to be time-dependent and subject-specific : $\begin{cases} Y_{ij} = Y_i(t_{ij}) = \widetilde{Y}_i(t_{ij}) + \epsilon_{ij} = X_{ij}^\top \beta + Z_{ij}^\top b_i + \\ \epsilon_{ij}(t_{ij}) \sim \mathcal{N}(0, \sigma_i^2(t_{ij})) \quad \text{with} \quad \log(\sigma_i(t_{ij})) = \end{cases}$

with X_{ij}, O_{ij}, Z_{ij} and M_{ij} four vectors of explanatory variables for subject i at visit j, respectively associated with the fixed-effect vectors β and μ, and the subject-specific random-effect vector b_i and τ_i, such as $\begin{pmatrix} b_i \\ \tau_i \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \Sigma_b & \Sigma_{\tau b} \\ \Sigma'_{\tau b} & \Sigma_\tau \end{pmatrix}\right)$**:**

$Y_i(t_{ij}) = \tilde{Y}_i(t_{ij}) + \epsilon_{ij} = X_{ij}^\top \beta + Z_{ij}^\top b_i + \epsilon_{ij}$

with X_{ij} and Z_{ij} two covariate vectors for subject i at visit j, respectively associated with the vector of fixed effects β and the vector of subject-specific individual random effects b_i. The vector b_i is assumed to be normally distributed and a specific-subject random effect on the variance of the measure error can be added: $\epsilon_{ij} \sim \mathcal{N}(0, \sigma_i^2)$ and

$$\begin{pmatrix} b_i \\ \log \sigma_i \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ \mu_\sigma \end{pmatrix}, \begin{pmatrix} \Sigma_b & 0 \\ 0 & \tau_\sigma^2 \end{pmatrix}\right)$$

B. SURVIVAL SUBMODEL

The risk function for the event \$k = {1,2}\$ is defined by: $\lambda_{ik}(t) = \lambda_{0k}(t) \exp\left(W_i^\top \gamma_k + \alpha_{1k} \tilde{y}_i(t) + \alpha_{2k} \tilde{y}'_i(t) + \alpha_{\sigma k} \sigma_i(t)\right)$

with $\lambda_{0k}(t)$ the baseline risk function, W_i a vector of baseline covariates associated with the regression coefficient γ_k, and α_{1k}, α_{2k} and $\alpha_{\sigma k}$ the regression coefficients associated with the

current value $\tilde{y}_i(t)$, the current slope $\tilde{y}'_i(t)$ and the current variability $\sigma_i(t)$ of the marker, respectively. Different parametric forms for the baseline risk function can be considered, such as exponential, Weibull, or, for more flexibility, a B-splines base.

Value

A FlexVarJoint object which contains the following elements :

`result` A marqLevAlg object with the results of the estimation.

`table.res` The table of results : Estimation and SE

`time.compute` Computation time

`control` A list of control elements

Examples

```
#fit a joint model with competing risks and subject-specific variability
example <- lsjm(formFixed = y~visit,
formRandom = ~ visit,
formGroup = ~ID,
formSurv = Surv(time, event ==1 ) ~ 1,
timeVar = "visit",
data.long = Data_toy,
variability_hetero = TRUE,
formFixedVar =~visit,
formRandomVar =~visit,
correlated_re = TRUE,
sharedtype = c("current value", "variability"),
hazard_baseline = "Weibull",
formSlopeFixed =~1,
formSlopeRandom = ~1,
indices_beta_slope = c(2),
competing_risk = TRUE,
formSurv_CR = Surv(time, event ==2 ) ~ 1,
hazard_baseline_CR = "Weibull",
sharedtype_CR = c("current value", "variability"),
S1 = 100,
S2 = 1000,
nproc = 1,
maxiter = 100,
Comp.Rcpp = TRUE
)

summary(example)
```

`lsmm` *lsmm : Estimation of location scale mixed model*

Description

This function fits complex mixed effects model with a time and covariate dependent variance. We suppose that the variance of the residual error is time-dependent and subject-specific. Parameters are estimated simultaneously through a maximum likelihood method, using a Marquardt-Levenberg algorithm.

Usage

```
lsmm(
  formFixed,
  formRandom,
  formGroup,
  timeVar,
  data.long,
  variability_hetero = TRUE,
  formFixedVar,
  formRandomVar,
  correlated_re = FALSE,
  S1 = 1000,
  S2 = 5000,
  nproc = 1,
  clustertype = "SOCK",
  maxiter = 100,
  print.info = FALSE,
  file = NULL,
  epsa = 0.001,
  epsb = 0.001,
  epsd = 0.001,
  binit = NULL
)
```

Arguments

`formFixed`	A formula for the fixed effects of the longitudinal submodel
`formRandom`	A formula for the random effects of the longitudinal submodel
`formGroup`	A formula which indicates the group variable
`timeVar`	The name of the column of time in data.long. This variable must appears in data.long
`data.long`	A dataframe with the longitudinal data
`variability_hetero`	A logical to indicate if we suppose a subject_specific variability

formFixedVar	A formula for the fixed effects of the variance predictor
formRandomVar	A formula for the random effects of the variance predictor
correlated_re	A logical to indicate if the random effects of the marker and the variance predictors are correlated (By default there are supposed to be independent)
S1	An integer : the number of QMC draws for the first step
S2	An integer : the number of QMC draws for the second step
nproc	An integer : the number of processors for parallel computing
clustertype	one of the supported types from makeCluster function
maxiter	optional maximum number of iterations for the marqLevAlg iterative algorithm.
print.info	logical indicating if the outputs of each iteration should be written
file	optional character giving the name of the file where the outputs of each iteration should be written (if print.info=TRUE)
epsa	optional threshold for the convergence criterion based on the parameter stability.
epsb	optional threshold for the convergence criterion based on the objective function stability.
epsd	optional threshold for the relative distance to maximum. This criterion has the nice interpretation of estimating the ratio of the approximation error over the statistical error, thus it can be used for stopping the iterative process whatever the problem.
binit	optional initials parameters.

Details

The model is defined by : #' $$\left\{ \begin{array}{l} Y_{ij} = Y_i(t_{ij}) = \widetilde{Y}_i(t_{ij}) + \epsilon_{ij} = X_{ij}^\top \beta + Z_{ij}^\top b_i + \epsilon_{ij}, \\ \epsilon_{ij}(t_{ij}) \sim \mathcal{N}(0, \sigma_i^2(t_{ij})) \quad \text{with} \quad \log(\sigma_i(t_{ij})) = O_{ij}^\top \mu + M_{ij}^\top \tau_i \end{array} \right.$$

with X_{ij}, O_{ij}, Z_{ij} and M_{ij} four vectors of explanatory variables for subject i at visit j, respectively associated with the fixed-effect vectors β and μ, and the subject-specific random-effect vector b_i and τ_i, such as $\begin{pmatrix} b_i \\ \tau_i \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \Sigma_b & \Sigma_{\tau b} \\ \Sigma'_{\tau b} & \Sigma_\tau \end{pmatrix}\right)$**:**

$Y_i(t_{ij}) = \tilde{Y}_i(t_{ij}) + \epsilon_{ij} = X_{ij}^\top \beta + Z_{ij}^\top b_i + \epsilon_{ij}$

with X_{ij} and Z_{ij} two covariate vectors for subject i at visit j, respectively associated with the vector of fixed effects β and the vector of subject-specific individual random effects b_i. The vector b_i is assumed to be normally distributed and a specific-subject random effect on the variance of the measure error can be added: $\epsilon_{ij} \sim \mathcal{N}(0, \sigma_i^2)$ and

$$\begin{pmatrix} b_i \\ \log \sigma_i \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ \mu_\sigma \end{pmatrix}, \begin{pmatrix} \Sigma_b & 0 \\ 0 & \tau_\sigma^2 \end{pmatrix}\right)$$

Value

A FlexVarJoint object which contains the following elements :

result A marqLevAlg object with the results of the estimation.

table.res The table of results : Estimation and SE

time.compute Computation time

control A list of control elements

Examples

```
#fit a joint model with competing risks and subject-specific variability
example <- lsmm(formFixed = y~visit,
formRandom = ~ visit,
formGroup = ~ID,
timeVar = "visit",
data.long = Data_toy,
variability_hetero = TRUE,
formFixedVar =~visit,
formRandomVar =~visit,
correlated_re = TRUE,
S1 = 100,
S2 = 1000,
nproc = 1,
maxiter = 100
)

summary(example)
```

`pred_s.t.bootstrap.tps`

Predictions computation

Description

Predictions computation

Usage

```
pred_s.t.bootstrap.tps(newdata, object, s, window, event = 1, nb.draws)
```

Arguments

`newdata`	data frame : collected data for a new individual
`object`	lsjm object : estimation of the model
`s`	numeric : the time to begin prediction
`window`	numeric : the side of the prediction window
`event`	integer (0, 1 or 2) : the event of interest for the prediction
`nb.draws`	integer : the number of draws to compute the IC

pred_s.t.ponctuel.tps *Predictions computation*

Description

Predictions computation

Usage

```
pred_s.t.ponctuel.tps(newdata, object, s, window, event = 1)
```

Arguments

newdata	data frame : collected data for a new individual
object	lsjm object : estimation of the model
s	numeric : the time to begin prediction
window	numeric : the side of the prediction window
event	integer (0, 1 or 2) : the event of interest for the prediction

predyn *Dynamic prediction for new individuals*

Description

Dynamic prediction for new individuals

Usage

```
predyn(
  newdata,
  object,
  s,
  times,
  event = 1,
  IC = 95,
  nb.draws = 500,
  graph = FALSE
)
```

Arguments

newdata	data frame : collected data for a new individual
object	lsjm object : estimation of the model
s	numeric : the time to begin prediction
times	numeric vector : future times to calculate predictions
event	integer (0, 1 or 2) : the event of interest for the prediction
IC	integer : percentage of confidence for the interval confidence (between 0 and 100), 95 by default, NULL if no IC
nb.draws	integer : the number of simulations to compute the interval confidence (by bootstrap), 500 by default
graph	boolean : indicator to plot the graphs or not

Value

A table of dynamic predictions

Examples

```
#fit a joint model with competing risks and subject-specific variability
example <- lsjm(formFixed = y~visit,
formRandom = ~ visit,
formGroup = ~ID,
formSurv = Surv(time, event ==1 ) ~ 1,
timeVar = "visit",
data.long = Data_toy,
variability_hetero = TRUE,
formFixedVar =~visit,
formRandomVar =~visit,
correlated_re = TRUE,
sharedtype = c("current value", "variability"),
hazard_baseline = "Weibull",
formSlopeFixed =~1,
formSlopeRandom = ~1,
indices_beta_slope = c(2),
competing_risk = TRUE,
formSurv_CR = Surv(time, event ==2 ) ~ 1,
hazard_baseline_CR = "Weibull",
sharedtype_CR = c("current value", "variability"),
S1 = 100,
S2 = 1000,
nproc = 1,
maxiter = 100,
Comp.Rcpp = TRUE
)

#Prediction for individuals 1 and 3 to experiment the event 1
#at time 1.5, 2, and 3, given their measurements until time 1:
```

```
newdata <- Data_toy[which(Data_toy$ID %in% c(1,3)),]
pred.new <- predyn(newdata,example,1, c(1.5,2,2.8,3), event = 1, IC = 95,
nb.draws = 100, graph = TRUE)
```

Index

∗ **datasets**
Data_toy, 6

data.GaussKronrod, 2
data.GaussKronrod2, 3
data.manag.long, 4
data.manag.surv, 4
data.time, 5
Data_toy, 6

gaussKronrod, 6
goodness_of_fit, 7

initial.long, 8

log_llh, 9
log_llh_rcpp, 12
lsjm, 16
lsmm, 20

pred_s.t.bootstrap.tps, 22
pred_s.t.ponctuel.tps, 23
predyn, 23