

Package ‘frailtypack’

November 20, 2013

Version 2.5

Date 2013-11-20

Title General Frailty models: shared, joint and nested frailty models with prediction

Author Virginie Rondeau, Juan R. Gonzalez, Yassin Mazroui, Audrey Mauguén, Amadou Diakite and Alexandre Laurent

Maintainer Virginie Rondeau <Virginie.Rondeau@isped.u-bordeaux2.fr>

Depends R (>= 2.10), survival, boot, MASS, survC1

LazyLoad yes

Description Frailtypack now fits several classes of frailty models using a penalized likelihood estimation on the hazard function but also a parametric estimation. 1) A shared frailty model (with gamma or log-normal frailty distribution) and Cox proportional hazard model. Clustered and recurrent survival times can be studied. 2) Additive frailty models for proportional hazard models with two correlated random effects (intercept random effect with random slope). 3) Nested frailty models for hierarchically clustered data (with 2 levels of clustering) by including two iid gamma random effects. 4) Joint frailty models in the context of joint modelling for recurrent events with terminal event for clustered data or not. Prediction values are available. Left truncated (not for Joint model), right-censored data, interval-censored data (only for Cox proportional hazard and shared frailty model) and strata (max=2) are allowed. In each model, the random effects have a gamma distribution, but you can switch to a log-normal in shared and joint models. Now, you can also consider time-varying covariates effects in Cox, shared and joint models. The package includes concordance measures for Cox proportional hazards models and for shared frailty models.

License GPL (>= 2.0)

URL http://virginierondeau.com/BiostatisticalConsulting/Liste_of_examples.html

Repository CRAN

Date/Publication 2013-11-20 17:43:45

NeedsCompilation yes

R topics documented:

frailtypack-package	3
additivePenal	5
bcos	10
Cmeasures	10
dataAdditive	13
dataNested	14
frailtyPenal	15
hazard	25
num.id	26
plot.additivePenal	27
plot.frailtyPenal	28
plot.jointPenal	30
plot.nestedPenal	31
plot.prediction	32
prediction	33
print.additivePenal	38
print.Cmeasures	39
print.frailtyPenal	39
print.jointPenal	40
print.nestedPenal	41
print.prediction	42
readmission	42
slope	43
subcluster	44
summary.additivePenal	45
summary.frailtyPenal	46
summary.jointPenal	48
summary.nestedPenal	49
SurvIC	50
survival	51
terminal	52
timedep	53

Index

55

frailtypack-package	<i>General Frailty models: shared, joint and nested frailty models with prediction</i>
---------------------	--

Description

Frailtypack now fits several classes of frailty models using a penalized likelihood estimation on the hazard function but also a parametric estimation. 1) A shared frailty model and Cox proportional hazard model. Clustered and recurrent survival times can be studied. 2) Additive frailty models for proportional hazard models with two correlated random effects (intercept random effect with random slope). 3) Nested frailty models for hierarchically clustered data (with 2 levels of clustering) by including two iid gamma random effects. 4) Joint frailty models in the context of joint modelling for recurrent events with terminal event for clustered data or not. Prediction values are available. Left truncated (not for Joint model), right-censored data, interval-censored data (only for Cox proportional hazard and shared frailty model) and strata (max=2) are allowed. In each model, the random effects have a gamma distribution, but you can switch to a log-normal in shared and joint. Now, you can also consider time-varying effect covariates in Cox, shared and joint models. The package includes concordance measures for Cox proportional hazards models and for shared frailty models.

Details

Package:	frailtypack
Type:	Package
Version:	2.5
Date:	2013-11-20
License:	GPL (>= 2.0)
LazyLoad:	yes

Author(s)

Virginie Rondeau, Juan R. Gonzalez, Yassin Mazroui, Audrey Mauguen, Amadou Diakite and Alexandre Laurent

References

- V. Rondeau, Y. Mazroui and J. R. Gonzalez (2012). Frailtypack: An R package for the analysis of correlated survival data with frailty models using penalized likelihood estimation or parametric estimation. *Journal of Statistical Software* **47**, 1-28.
- V. Rondeau and J. R. Gonzalez (2005). Frailtypack: A computer program for the analysis of correlated failure time data using penalized likelihood estimation. *Computer Methods and Programs in Biomedicine* **80**, 2, 154-164.

- V. Rondeau, S. Michiels, B. Liqueur, and J. P. Pignon (2008). Investigating trial and treatment heterogeneity in an individual patient data meta-analysis of survival data by mean of the maximum penalized likelihood approach. *Statistics in Medicine*, **27**, 1894-1910.
- V. Rondeau, S. Mathoulin-Pellissier, H. Jacqmin-Gadda, V. Brouste, P. Soubeyran (2007). Joint frailty models for recurring events and death using maximum penalized likelihood estimation: application on cancer events. *Biostatistics*, **8**, 4, 708-721.
- V. Rondeau, D. Commenges, and P. Joly (2003). Maximum penalized likelihood estimation in a gamma-frailty model. *Lifetime Data Analysis* **9**, 139-153.
- D. Marquardt (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, 431-441.
- V. Rondeau, L. Filleul, P. Joly (2006). Nested frailty models using maximum penalized likelihood estimation. *Statistics in Medicine*, **25**, 4036-4052.

Examples

```
## Not run:

###--- Additive model with 1 covariate ---###

data(dataAdditive)
modAdd <- additivePenal(Surv(t1,t2,event)~
  cluster(group)+var1+slope(var1),
  correlation=TRUE,data=dataAdditive,
  n.knots=8,kappa1=10000,hazard="Splines")

###--- Joint model (recurrent and terminal events) with 2 covariates ---###

data(readmission)
modJoint.gap <- frailtyPenal(Surv(time,event)~
  cluster(id)+sex+dukes+charlson+terminal(death),
  formula.terminalEvent=~sex+dukes+charlson,
  data=readmission,n.knots=14,kappa1=9.55e+9,
  kappa2=1.41e+12,Frailty=TRUE,joint=TRUE,
  recurrentAG=FALSE,hazard="Splines")

###--- Nested model (or hierarchical model) with 2 covariates ---###

data(dataNested)
modClu <- frailtyPenal(Surv(t1,t2,event)~
  cluster(group)+subcluster(subgroup)+cov1+cov2,
  Frailty=TRUE,data=dataNested,n.knots=8,
  kappa1=50000,hazard="Splines")

###--- Semiparametric Shared model ---###

data(readmission)
frailtyPenal(Surv(t.start,t.stop,event)~
  as.factor(sex)+as.factor(dukes)+as.factor(charlson)+
  cluster(id),data=readmission, Frailty=TRUE,
```

```

n.knots=6,kappa1=5000,recurrentAG=TRUE,
cross.validation=TRUE,hazard="Splines")

###--- Parametric Shared model ---###

data(readmission)
frailtyPenal(Surv(t.start,t.stop,event)~
cluster(id)+sex+dukes+charlson,
data=readmission,Frailty=TRUE,recurrentAG=TRUE,
hazard="Piecewise-per",nb.int1=6)

## End(Not run)

```

additivePenal	<i>Fit an Additive Frailty model using a semiparametric penalized likelihood estimation or a parametric estimation</i>
---------------	--

Description

Fit an additive frailty model using a semiparametric penalized likelihood estimation or a parametric estimation. The main issue in a meta-analysis study is how to take into account the heterogeneity between trials and between the treatment effects across trials. Additive models are proportional hazard model with two correlated random trial effects that act either multiplicatively on the hazard function or in interaction with the treatment, which allows studying for instance meta-analysis or multicentric datasets. Right-censored data are allowed, but not the left-truncated data. A stratified analysis is possible (maximum number of strata: 2). This approach is different from the shared frailty models.

In an additive model, the hazard function for the j^{th} subject in the i^{th} trial with random trial effect u_i as well as the random treatment-by-trial interaction v_i is:

$$\begin{cases} \lambda_{ij}(t|u_i, v_i) = \lambda_0(t) \exp(u_i + v_i X_{ij1} + \sum_{k=1}^p \beta_k X_{ijk}) \\ \mathbf{cov}(u_i, v_i) = \rho \sigma \tau \\ u_i \sim \mathcal{N}(0, \sigma^2), v_i \sim \mathcal{N}(0, \tau^2) \end{cases}$$

where $\lambda_0(t)$ is the baseline hazard function, β_k the fixed effect associated to the covariate X_{ijk} ($k=1, \dots, p$), β_1 is the treatment effect and X_{ij1} the treatment variable. ρ is the corresponding correlation coefficient for the two frailty terms.

Usage

```

additivePenal(formula, data, correlation = FALSE, recurrentAG =
FALSE, cross.validation = FALSE, n.knots, kappa1,
kappa2, maxit = 350, hazard = "Splines", nb.int1,
LIMparam=1e-4, LIMlog1=1e-4, LIMderiv=1e-3)

```

Arguments

formula	a formula object, with the response on the left of a <code>~</code> operator, and the terms on the right. The response must be a survival object as returned by the 'Surv' function like in survival package. The <code>slope()</code> function is required.
data	a 'data.frame' in which to interpret the variables named in the 'formula'.
correlation	Logical value. Are the random effects correlated? If so, the correlation coefficient is estimated. The default is FALSE.
recurrentAG	Always FALSE for additive models (left-truncated data are not allowed).
cross.validation	Logical value. Is cross validation procedure used for estimating smoothing parameter in the penalized likelihood estimation? If so a search of the smoothing parameter using cross validation is done, with <code>kappa1</code> as the seed. The cross validation is not implemented for two strata. The default is FALSE.
n.knots	integer giving the number of knots to use. Value required in the penalized likelihood estimation. It corresponds to the $(n.knots+2)$ splines functions for the approximation of the hazard or the survival functions. Number of knots must be between 4 and 20. (See Note)
kappa1	positive smoothing parameter in the penalized likelihood estimation. The coefficient κ of the integral of the squared second derivative of hazard function in the fit. To obtain an initial value for <code>kappa1</code> (or <code>kappa2</code>), a solution is to fit the corresponding shared frailty model using cross validation (See <code>cross.validation</code>). We advise the user to identify several possible tuning parameters, note their defaults and look at the sensitivity of the results to varying them. Value required. (See Note)
kappa2	Positive smoothing parameter in the penalized likelihood estimation for the second stratum when data are stratified. See <code>kappa1</code> .
maxit	maximum number of iterations for the Marquardt algorithm. Default is 350
hazard	Type of hazard functions: "Splines" for semiparametric hazard functions with the penalized likelihood estimation, "Piecewise-per" for piecewise constant hazards functions using percentile, "Piecewise-equi" for piecewise constant hazard functions using equidistant intervals, "Weibull" for parametric weibull functions. Default is "Splines".
nb.int1	Number of intervals (between 1 and 20) for the parametric hazard functions ("Piecewise-per", "Piecewise-equi").
LIMparam	Convergence threshold of the Marquard algorithm for the parameters (see Details), 10^{-4} by default.
LIMlogl	Convergence threshold of the Marquard algorithm for the log-likelihood (see Details), 10^{-4} by default.
LIMderiv	Convergence threshold of the Marquard algorithm for the gradient (see Details), 10^{-3} by default.

Details

The estimated parameter are obtained by maximizing the penalized log-likelihood or by a simple log-likelihood (in the parametric case) using the robust Marquardt algorithm (Marquardt, 1963). The

parameters are initialized with values obtained with Cox proportional hazard model. The iterations are stopped when the difference between two consecutive loglikelihoods was small ($< 10^{-4}$), the estimated coefficients were stable (consecutive values $< 10^{-4}$), and the gradient small enough ($< 10^{-3}$). To be sure of having a positive function at all stages of the algorithm, the spline coefficients were reparametrized to be positive at each stage. The variance space of the two random effects is reduced, so the variances are positive, and the correlation coefficient values are constrained to be between -1 and 1. The marginal log-likelihood depends on integrations that are approximated by using the Laplace integration technique with a first order approximation. The smoothing parameter can be fixed or estimated by maximizing likelihood cross-validation criterion. The usual squared Wald statistic was modified to a mixture of two χ^2 distribution to get significance test for the variance of the random effects.

INITIAL VALUES

The splines and the regression coefficients are initialized to 0.1. An adjusted Cox model is fitted, it provides new initial values for the splines coefficients and the regression coefficients. The variances of the frailties are initialized to 0.1. Then an additive frailty model with independent frailties is fitted. At last, an additive frailty model with correlated frailties is fitted.

Value

An additive model or more generally an object of class 'additivePenal'. Methods defined for 'additivePenal' objects are provided for print, plot and summary.

b	sequence of the corresponding estimation of the splines coefficients, the random effects variances and the regression coefficients.
call	The code used for fitting the model.
coef	the regression coefficients.
cov	covariance between the two frailty terms ($cov(u_i, v_i)$)
cross.Val	Logical value. Is cross validation procedure used for estimating the smoothing parameters in the penalized likelihood estimation?
correlation	Logical value. Are the random effects correlated?
DoF	degrees of freedom associated with the "kappa".
formula	the formula part of the code used for the model.
groups	the maximum number of groups used in the fit.
kappa	A vector with the smoothing parameters in the penalized likelihood estimation corresponding to each baseline function as components.
lam	matrix of hazard estimates and confidence bands.
lam2	the same value as lam for the second stratum.
loglikPenal	the complete marginal penalized log-likelihood in the semiparametric case.
loglik	the marginal log-likelihood in the parametric case.
n	the number of observations used in the fit.
n.events	the number of events observed in the fit.
n.iter	number of iterations needed to converge.
n.knots	number of knots for estimating the baseline functions.

n.strat	number of stratum.
rho	the corresponding correlation coefficient for the two frailty terms.
sigma2	Variance for the random intercept (the random effect associated to the baseline hazard functions).
surv	matrix of baseline survival estimates and confidence bands.
surv2	the same value as surv for the the second stratum.
tau2	Variance for the random slope (the random effect associated to the treatment effect across trials).
varH	the variance matrix of all parameters before positivity constraint transformation (Sigma2, Tau2, the regression coefficients and the spline coefficients). Then after, the delta method is needed to obtain the estimated variance parameters.
varHIH	the robust estimation of the variance matrix of all parameters (Sigma2, Tau2, the regression coefficients and the spline coefficients).
varSigma2	The variance of the estimates of "sigma2".
varTau2	The variance of the estimates of "tau2".
varcov	Variance of the estimates of "cov".
x1	vector of times where both survival and hazard functions for the recurrent events are estimated. By default seq(0,max(time),length=99), where time is the vector of survival times.
x2	vector of times for the second stratum (see x1 value).
type.of.hazard	Type of hazard functions (0:"Splines", "1:Piecewise", "2:Weibull").
type.of.Piecewise	Type of Piecewise hazard functions (1:"percentile", 0:"equidistant").
nbintervR	Number of intervals (between 1 and 20) for the parametric hazard functions ("Piecewise-per", "Piecewise-equi").
npar	number of parameters.
nvar	number of explanatory variables.
noVar	indicator of explanatory variable.
LCV	the approximated likelihood cross-validation criterion in the semiparametric case (with H minus the converged hessian matrix, and l(.) the full log-likelihood).

$$LCV = \frac{1}{n}(\text{trace}(H_{pl}^{-1}H) - l(.))$$

AIC	the Akaike information Criterion for the parametric case.
-----	---

$$AIC = \frac{1}{n}(np - l(.))$$

n.knots.temp	initial value for the number of knots.
shape.weib	shape parameter for the weibull hazard function.
scale.weib	scale parameter for the weibull hazard function.
martingale.res	martingale residuals for each cluster.

frailty.pred	empirical Bayes prediction of the first frailty term.
frailty.pred2	empirical Bayes prediction of the second frailty term.
linear.pred	linear predictor: uses simply "Beta'X + u_i + v_i * X_1" in the additive Frailty models.
global_chisq	a vector with the values of each multivariate Wald test.
dof_chisq	a vector with the degree of freedom for each multivariate Wald test.
global_chisq.test	a binary variable equals to 0 when no multivariate Wald is given, 1 otherwise.
p.global_chisq	a vector with the p_values for each global multivariate Wald test.
names.factor	Names of the "as.factor" variables.
Xlevels	vector of the values that factor might have taken.
contrasts	type of contrast for factor variable.

Note

"Kappa" (kappa1 and kappa2) and "n.knots" are the arguments that the user have to change if the fitted model does not converge. "n.knots" takes integer values between 4 and 20. But with n.knots=20, the model would take a long time to converge. So, usually, begin first with n.knots=7, and increase it step by step until it converges. "Kappa" only takes positive values. So, choose a value for Kappa (for instance 10000), and if it does not converge, multiply or divide this value by 10 or 5 until it converges.

References

V. Rondeau, Y. Mazroui and J. R. Gonzalez (2012). Frailtypack: An R package for the analysis of correlated survival data with frailty models using penalized likelihood estimation or parametric estimation. *Journal of Statistical Software* **47**, 1-28.

V. Rondeau, S. Michiels, B. Lique, and J. P. Pignon (2008).

Investigating trial and treatment heterogeneity in an individual patient data meta-analysis of survival data by mean of the maximum penalized likelihood approach. *Statistics in Medicine*, **27**, 1894-1910.

See Also

[slope](#)

Examples

```
## Not run:

###--- Additive model with 1 covariate ---###

data(dataAdditive)

modAdd <- additivePenal(Surv(t1,t2,event)~cluster(group)+
var1+slope(var1),correlation=TRUE,data=dataAdditive,
```

```
n.knots=8,kappa1=10000)

#-- Var1 is boolean as a treatment variable

## End(Not run)
```

bcos

Breast Cosmesis Data

Description

The often used data set for interval-censored data, described and given in full in Finkelstein and Wolfe (1985). It involves 94 breast cancer patients who were randomized to either radiation therapy with chemotherapy or radiation therapy alone. The outcome is time until the onset of breast retraction which is interval-censored between the last clinic visit before the event was observed and the first visit when the event was observed. Patients without breast retraction were right-censored.

Usage

```
data(bcos)
```

Format

A data frame with 94 observations and 3 variables:

left left end point of the breast retraction interval

right right end point of the breast retraction interval

treatment type of treatment received

Source

Finkelstein, D.M. and Wolfe, R.A. (1985). A semiparametric model for regression analysis of interval-censored failure time data. *Biometrics* **41**, 731-740.

Cmeasures

Concordance measures in shared frailty and Cox proportional hazard models

Description

Compute concordance probability estimation for Cox proportional hazard or shared frailty models in case of grouped data (Mauguen et al. 2012). Concordance is given at different levels of comparison, taking into account the cluster membership: between-groups, within-groups and an overall measure, being a weighted average of the previous two. Can also compute the c-index (Harrell et al. 1996) at these three levels. It is possible to exclude tied pairs from concordance estimation (otherwise, account for 1/2).

Usage

```
Cmeasures(fitc, ties = 1, marginal = 0, cindex = 0, Nboot = 0,
          tau = 0, data.val)
```

Arguments

fitc	A frailtyPenal object, for a shared frailty model. If the fit is a Cox model (Frailty=FALSE), no clustering membership is taken into account and only marginal concordance probability estimation is provided. Only an overall measure is given, where all patients are compared two by two. If a counting process formulation is used to performed the fit, with 't.start' and 't.stop', the gap-times (t.stop-t.start) are used in the concordance estimation.
ties	Indicates if the tied pairs on prediction value must be included (ties=1) or excluded (ties=0) from the concordance estimation. Default is ties=1. When included, tied pairs account for 1/2 in the concordance.
marginal	Indicates if the concordance based on marginal predictions must be given (marginal=1) in addition to conditional ones or not (marginal=0). Marginal predictions do not include the frailty estimation in the linear predictor computation: uses "'Beta'X" instead of "Beta'X + log z_i". Default is marginal=0.
cindex	Indicates if the c-index (Harrell et al. 1996) must be computed (cindex=1) in addition to the concordance probability estimation or not (cindex=0). C-index is also given at the three comparison levels (between, within and overall). Default is cindex=0.
Nboot	Number of bootstrap resamplings to compute standard-error of the concordances measures, as well as a percentile 95% confidence interval. Nboot=0 indicates no bootstrap procedure. Maximum admitted is 1000. Minimum admitted is 2. Default is 0. Resampling is done at the group level. If Cox model is used, resampling is done at individual level.
tau	Time used to limit the interval on which the concordance is estimated. Note that the survival function for the underlying censoring time distribution needs to be positive at tau. If tau=0, the maximum of the observed event times is used. Default is tau=0.
data.val	A dataframe. It is possible to specify a different dataset than the one used in the model input in the argument 'fitc'. This new dataset will be a validation population and the function will compute new concordance measures from the parameters estimated on the development population. In this case for conditional measures, the frailties are a posteriori predicted. The two datasets must have the same covariates with the same coding without missing data.

Value

call	The shared frailty model evaluated.
Frailty	Logical value. Was model with frailties fitted.
frequencies	Numbers of patients, events and groups used to fit the model.

Npairs	Number of pairs of subjects, between-groups, within-groups and over all the population. If cindex=1, number of comparable (useable) pairs also available.
Nboot	Number of bootstrap resamplings required.
ties	A binary, indicating if the tied pairs on prediction were used to compute the concordance.
CPEcond	Values of Gonen & Heller's measure (conditional). If Nboot>0, give SE, the standard-error of the parameters evaluated by bootstrap, IC.low and IC.high, the lower and upper bounds of the percentile confidence interval evaluated by bootstrap (2.5% and 97.5% percentiles).
Cunocond	Values of Uno's measure (conditional). If Nboot>0, give SE, the standard-error of the parameters evaluated by bootstrap, IC.low and IC.high, the lower and upper bounds of the percentile confidence interval evaluated by bootstrap (2.5% and 97.5% percentiles).
marginal	A binary, indicating if the marginal values were computed.
CPEmarg	Values of Gonen & Heller's measure (marginal), if marginal=1. If Nboot>0, give SE, the standard-error of the parameters evaluated by bootstrap, IC.low and IC.high, the lower and upper bounds of the percentile confidence interval evaluated by bootstrap (2.5% and 97.5% percentiles).
Cunomarg	Values of Uno's measure (marginal), if marginal=1. If Nboot>0, give SE, the standard-error of the parameters evaluated by bootstrap, IC.low and IC.high, the lower and upper bounds of the percentile confidence interval evaluated by bootstrap (2.5% and 97.5% percentiles).
cindex	A binary, indicating if the c-indexes were computed.
cindexcond	Values of the C-index of Harrell (conditional). If Nboot>0, give SE, the standard-error of the parameters evaluated by bootstrap, IC.low and IC.high, the lower and upper bounds of the percentile confidence interval evaluated by bootstrap (2.5% and 97.5% percentiles).
cindexmarg	Values of the C-index of Harrell (marginal), if marginal=1. If Nboot>0, give SE, the standard-error of the parameters evaluated by bootstrap, IC.low and IC.high, the lower and upper bounds of the percentile confidence interval evaluated by bootstrap (2.5% and 97.5% percentiles).

References

- Mauguen, A., Collette, S., Pignon, J. P. and Rondeau, V. (2013). Concordance measures in shared frailty models: application to clustered data in cancer prognosis. *Statistics in Medicine* **32**, 27, 4803-4820
- Harrell, F.E. et al. (1996). Tutorial in biostatistics: multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in Medicine* **15**, 361-387.
- Gonen, M., Heller, G. (2005). Concordance probability and discriminatory power in proportional hazards regression. *Biometrika* **92**, 965-970.

See Also

[print.Cmeasures,frailtyPenal](#)

Examples

```
## Not run:

#-- load data
data(readmission)

#-- a frailtypenal fit
fit <- frailtyPenal(Surv(time,event)~cluster(id)+dukes+
charlson+chemo,data=readmission,Frailty=TRUE,cross.validation=FALSE,
n.knots=10,kappa1=1,hazard="Splines")

#-- a Cmeasures call
fit.Cmeasures <- Cmeasures(fit)
fit.Cmeasures.noties <- Cmeasures(fit, ties=0)
fit.Cmeasures.marginal <- Cmeasures(fit, marginal=1)
fit.Cmeasures.cindex <- Cmeasures(fit, cindex=1)

#-- a short summary
fit.Cmeasures
fit.Cmeasures.noties
fit.Cmeasures.marginal
fit.Cmeasures.cindex

## End(Not run)
```

dataAdditive

Simulated data as a gathering of clinical trials databases

Description

This contains simulated samples of 100 clusters with 100 subjects in each cluster, like a gathering of clinical trials databases. Two correlated centered gaussian random effects are generated with the same variance fixed at 0.3 and the covariance at -0.2. The regression coefficient β is fixed at -0.11. The percentage of right-censored data is around 30 percent which are generated from a uniform distribution on [1,150]. The baseline hazard function is considered as a simple Weibull.

Usage

```
data(dataAdditive)
```

Format

This data frame contains the following columns:

group identification variable

t1 start of interval (=0, because left-truncated data are not allowed)

t2 end of interval (death or censoring time)
event censoring status (0:alive, 1:death, as acensoring indicator)
var1 dichotomous covariate (=0 or 1,as a treatment variable)
var2 dichotomous covariate (=0 or 1,as a treatment variable)

Source

V. Rondeau, S. Michiels, B.Liquet, and J.P. Pignon (2008). Investigating trial and treatment heterogeneity in an individual patient data meta-analysis of survival data by mean of the maximum penalized likelihood approach. *Statistics in Medecine*, **27**, 1894-1910.

dataNested

Simulated data with two levels of grouping

Description

This contains a simulated sample of 400 observations which allow establishing 20 clusters with 4 subgroups and 5 subjects in each subgroup, in order to obtain two levels of grouping. This data set is useful to illustrate how to fit a nested model. Two independent gamma frailty parameters with a variance fixed at 0.1 for the cluster effect and at 0.5 for the subcluster effect were generated. Independent survival times were generated from a simple Weibull baseline risk function. The percentage of censoring data was around 30 per cent. The right-censoring variables were generated from a uniform distribution on [1,36] and a left-truncating variable was generated with a uniform distribution on [0,10]. Observations were included only if the survival time is greater than the truncated time.

Usage

`data(dataNested)`

Format

This data frame contains the following columns:

group group identification variable
subgroup subgroup identification variable
t1 start of interval (0 or truncated time)
t2 end of interval (death or censoring time)
event censoring status (0:alive, 1:death)
cov1 dichotomous covariate (0,1)
cov2 dichotomous covariate (0,1)

Source

V. Rondeau, L. Filleul, P. Joly (2006). Nested frailty models using maximum penalized likelihood estimation. *Statistics in Medecine*, **25**, 4036-4052.

frailtyPenal

*Fit a Shared, Joint or Nested Frailty model***Description****Shared Frailty model**

Fit a shared gamma or log-normal frailty model using a semiparametric Penalized Likelihood estimation or parametric estimation on the hazard function. Left truncated, right-censored data, interval-censored data and strata (max=2) are allowed. It allows to obtain a non-parametric smooth hazard of survival function. This approach is different from the partial penalized likelihood approach of Therneau et al.

The hazard function, conditionnal on the frailty term ω_i , of a shared gamma frailty model for the j^{th} subject in the i^{th} group:

$$\lambda_{ij}(t|\omega_i) = \lambda_0(t)\omega_i \exp(\beta' Z_{ij})$$

$$\omega_i \sim \Gamma\left(\frac{1}{\theta}, \frac{1}{\theta}\right) \quad \mathbf{E}(\omega_i) = 1 \quad \mathbf{Var}(\omega_i) = \theta$$

where $\lambda_0(t)$ is the baseline hazard function, β the vector of the regression coefficient associated to the covariate vector Z_{ij} for the j^{th} individual in the i^{th} group.

Otherwise, in case of a shared log-normal frailty model, we have for the j^{th} subject in the i^{th} group:

$$\lambda_{ij}(t|\eta_i) = \lambda_0(t) \exp(\eta_i + \beta' Z_{ij})$$

$$\eta_i \sim N(0, \sigma^2)$$

From now on, you can also consider time-varying effects covariates in your model, see `timedep` function for more details.

Joint Frailty model

Fit a joint either with gamma or log-normal frailty model for recurrent and terminal events using a penalized likelihood estimation on the hazard function or a parametric estimation. Right-censored data is allowed. Left-truncated data and stratified analysis are not possible. Joint frailty models allow studying, jointly, survival processes of recurrent and terminal events, by considering the terminal event as an informative censoring.

There is two kinds of joint frailty models that can be fitted with `frailtyPenal` :

- The first one (Rondeau et al. 2007) includes a common frailty term to the individuals (ω_i) for the two rates which will take into account the heterogeneity in the data, associated with unobserved covariates. The frailty term acts differently for the two rates (ω_i for the recurrent rate and ω_i^α for the death rate). The covariates could be different for the recurrent rate and death rate.

For the j^{th} recurrence ($j = 1, \dots, n_i$) and the i^{th} subject ($i = 1, \dots, G$), the joint gamma frailty model for recurrent event hazard function $r_{ij}(\cdot)$ and death rate $\lambda_i(\cdot)$ is :

$$\begin{cases} r_{ij}(t|\omega_i) = \omega_i r_0(t) \exp(\beta_1' \mathbf{Z}_i(t)) & \text{(Recurrent)} \\ \lambda_i(t|\omega_i) = \omega_i^\alpha \lambda_0(t) \exp(\beta_2' \mathbf{Z}_i(t)) & \text{(Death)} \end{cases}$$

where $r_0(t)$ (resp. $\lambda_0(t)$) is the recurrent (resp. terminal) event baseline hazard function, β_1 (resp. β_2) the regression coefficient vector, $\mathbf{Z}_i(t)$ the covariate vector. The random effects of frailties $\omega_i \sim \Gamma(\frac{1}{\theta}, \frac{1}{\theta})$ and are iid.

The joint log-normal frailty model will be :

$$\begin{cases} r_{ij}(t|\eta_i) = r_0(t) \exp(\eta_i + \beta_1' \mathbf{Z}_i(t)) & \text{(Recurrent)} \\ \lambda_i(t|\eta_i) = \lambda_0(t) \exp(\alpha \eta_i + \beta_2' \mathbf{Z}_i(t)) & \text{(Death)} \end{cases}$$

where

$$\eta_i \sim N(0, \sigma^2)$$

- The second one (Rondeau et al. 2011) is quite similar but the frailty term is common to the individuals from a same group. This model is useful for joint modelling two clustered survival outcomes. In this case, j is for the subject and i for the cluster.

$$\begin{cases} r_{ij}(t|u_i) = u_i r_0(t) \exp(\beta_1' \mathbf{Z}_{ij}(t)) & \text{(Time to event)} \\ \lambda_{ij}(t|u_i) = u_i^\alpha \lambda_0(t) \exp(\beta_2' \mathbf{Z}_{ij}(t)) & \text{(Death)} \end{cases}$$

In case of a log-normal distribution of the frailties, we will have :

$$\begin{cases} r_{ij}(t|v_i) = r_0(t) \exp(v_i + \beta_1' \mathbf{Z}_{ij}(t)) & \text{(Time to event)} \\ \lambda_{ij}(t|v_i) = \lambda_0(t) \exp(\alpha v_i + \beta_2' \mathbf{Z}_{ij}(t)) & \text{(Death)} \end{cases}$$

where

$$v_i \sim N(0, \sigma^2)$$

This joint frailty model can also be applied to clustered recurrent events and a terminal event (example on "readmission" data below).

From now on, you can also consider time-varying effects covariates in your model, see `timedep` function for more details.

Nested Frailty model

Fit a nested frailty model using a Penalized Likelihood on the hazard function or using a parametric estimation. Nested frailty models allow survival studies for hierarchically clustered data by including two iid gamma random effects. Left truncated and censored data are allowed. Stratification analysis is allowed (maximum of strata=2).

The hazard function conditional on the two frailties v_i and w_{ij} for the k^{th} individual of the j^{th} subgroup of the i^{th} group is :

$$\begin{cases} \lambda_{ijk}(t|v_i, w_{ij}) = v_i w_{ij} \lambda_0(t) \exp(\beta' \mathbf{X}_{ijk}) \\ v_i \sim \Gamma\left(\frac{1}{\alpha}, \frac{1}{\alpha}\right) \text{ i.i.d. } \mathbf{E}(v_i) = 1 \quad \mathbf{Var}(v_i) = \alpha \\ w_{ij} \sim \Gamma\left(\frac{1}{\eta}, \frac{1}{\eta}\right) \text{ i.i.d. } \mathbf{E}(w_{ij}) = 1 \quad \mathbf{Var}(w_{ij}) = \eta \end{cases}$$

where $\lambda_0(t)$ is the baseline hazard function, \mathbf{X}_{ijk} denotes the covariate vector and β the corresponding vector of regression parameters.

Usage

```
frailtyPenal(formula, formula.terminalEvent, data, Frailty = FALSE,
             joint = FALSE, recurrentAG = FALSE, cross.validation =
             FALSE, n.knots, kappa1, kappa2, maxit = 350, hazard =
             "Splines", nb.int1, nb.int2, RandDist = "Gamma",
             betaknots = 1, betaorder = 3, B, LIMparam = 1e-4,
             LIMlog1 = 1e-4, LIMderiv = 1e-4)
```

Arguments

formula	a formula object, with the response on the left of a <code>~</code> operator, and the terms on the right. The response must be a survival object as returned by the <code>'Surv'</code> function like in survival package. In case of interval-censored data, the response must be an object as returned by the <code>'SurvIC'</code> function from this package
formula.terminalEvent	only for joint model: a formula object, only requires terms on the right to indicate which variables are modelling the terminal event.
data	a <code>'data.frame'</code> in which to interpret the variables named in the <code>'formula'</code>
Frailty	Logical value. Is model with frailties fitted? If so, variance of frailty parameter is estimated. If not, Cox proportional hazard model is estimated. The default is FALSE
joint	Logical value. Is joint model fitted? If so, <code>'formula.terminalEvent'</code> is required. The default is FALSE
recurrentAG	Logical value. Is Andersen-Gill model fitted? If so indicates that recurrent event times with the counting process approach of Andersen and Gill is used. This formulation can be used for dealing with time-dependent covariates. The default is FALSE.
cross.validation	Logical value. Is cross validation procedure used for estimating smoothing parameter in the penalized likelihood estimation? If so a search of the smoothing parameter using cross validation is done, with <code>kappa1</code> as the seed. The cross validation is not implemented for two strata, neither for interval-censored data. The cross validation has been implemented for a Cox proportional hazard model, with no covariates. The default is FALSE.
n.knots	integer giving the number of knots to use. Value required in the penalized likelihood estimation. It corresponds to the <code>(n.knots+2)</code> splines functions for the approximation of the hazard or the survival functions. Number of knots must be between 4 and 20. (See Note)
kappa1	positive smoothing parameter in the penalized likelihood estimation. The coefficient <code>kappa</code> of the integral of the squared second derivative of hazard function in the fit (penalized log likelihood). To obtain an initial value for <code>kappa1</code> (or <code>kappa2</code>), a solution is to fit the corresponding shared frailty model using cross validation (See <code>cross.validation</code>). We advise the user to identify several possible tuning parameters, note their defaults and look at the sensitivity of the results to varying them. Value required.(See Note)

kappa2	positive smoothing parameter in the penalized likelihood estimation for the second stratum when data are stratified or for the terminal event rate for joint model (stratification not allowed in joint model). See kappa1.
maxit	maximum number of iterations for the Marquardt algorithm. Default is 350
hazard	Type of hazard functions: "Splines" for semiparametric hazard functions with the penalized likelihood estimation, "Piecewise-per" for piecewise constant hazard function using percentile (not available for interval-censored data), "Piecewise-equi" for piecewise constant hazard function using equidistant intervals, "Weibull" for parametric weibull functions. Default is "Splines".
nb.int1	Number of intervals (between 1 and 20) for the parametric hazard functions ("Piecewise-per", "Piecewise-equi"). In a joint model, this argument is used for the recurrent hazard function.
nb.int2	only for joint model : number of intervals (between 1 and 20) for the death parametric hazard function ("Piecewise-per", "Piecewise-equi").
RandDist	Type of random effect distribution: "Gamma" for a gamma distribution, "LogN" for a log-normal distribution. Default is "Gamma". Not implemented for nested model.
betaknots	Number of inner knots used for the estimation of B-splines. Default is 1. See 'timedep' function for more details. Not implemented for nested model.
betaorder	Order of the B-splines. Default is cubic B-splines (order = 3). See 'timedep' function for more details. Not implemented for nested model.
B	A vector of initial values for regression coefficients. This vector should be of the same size as the whole vector of covariates. Default is 0.1 for each.
LIMparam	Convergence threshold of the Marquard algorithm for the parameters (see Details), 10^{-4} by default.
LIMlogl	Convergence threshold of the Marquard algorithm for the log-likelihood (see Details), 10^{-4} by default.
LIMderiv	Convergence threshold of the Marquard algorithm for the gradient (see Details), 10^{-4} by default.

Details

Typical usages are for a Cox model

```
frailtyPenal(Surv(time,event)~var1+var2, data, ...)
```

for a shared model

```
frailtyPenal(Surv(time,event)~cluster(group)+var1+var2,
             data, Frailty = TRUE, ...)
```

for a joint model

```
frailtyPenal(Surv(time,event)~cluster(group)+var1+var2+
             var3+terminal(death), formula.terminalEvent=~
             var1+var4, data, Frailty = TRUE, joint = TRUE, ...)
```

for a joint model for clustered data

```
frailtyPenal(Surv(time,event)~cluster(group)+num.id(group2)+
             var1+var2+var3+terminal(death),
             formula.terminalEvent=~var1+var4, data,
             Frailty = TRUE, joint = TRUE, ...)
```

for a nested model

```
frailtyPenal(Surv(time,event)~cluster(group)+subcluster(sbgrou)+
             var1+var2, data, Frailty = TRUE, ...)
```

The estimated parameter are obtained using the robust Marquardt algorithm (Marquardt, 1963) which is a combination between a Newton-Raphson algorithm and a steepest descent algorithm. The iterations are stopped when the difference between two consecutive loglikelihoods was small ($< 10^{-4}$), the estimated coefficients were stable (consecutive values $< 10^{-4}$), and the gradient small enough ($< 10^{-4}$). When frailty parameter is small, numerical problems may arise. To solve this problem, an alternative formula of the penalized log-likelihood is used (see Rondeau, 2003 for further details). Cubic M-splines of order 4 are used for the hazard function, and I-splines (integrated M-splines) are used for the cumulative hazard function.

The inverse of the hessian matrix is the variance estimator and to deal with the positivity constraint of the variance component and the spline coefficients, a squared transformation is used and the standard errors are computed by the Δ -method (Knight & Xekalaki, 2000). The smooth parameter can be chosen by maximizing a likelihood cross validation criterion (Joly and other, 1998). The integrations in the full log likelihood were evaluated using Gaussian quadrature. Laguerre polynomials with 20 points were used to treat the integrations on $[0, \infty[$.

INITIAL VALUES

The splines and the regression coefficients are initialized to 0.1. In case of shared model, the program fits, firstly, an adjusted Cox model to give new initial values for the splines and the regression coefficients. The variance of the frailty term θ is initialized to 0.1. Then, a shared frailty model is fitted.

In case of a joint frailty model, the splines and the regression coefficients are initialized to 0.5. The program fits an adjusted Cox model to have new initial values for the regression and the splines coefficients. The variance of the frailty term θ and the coefficient α associated in the death hazard function are initialized to 1. Then, it fits a joint frailty model.

In case of a nested model, the program fits an adjusted Cox model to provide new initial values for the regression and the splines coefficients. The variances of the frailties are initialized to 0.1. Then, a shared frailty model with covariates with only subgroup frailty is fitted to give a new initial value for the variance of the subgroup frailty term. Then, a shared frailty model with covariates and only group frailty terms is fitted to give a new initial value for the variance of the group frailties. In a last step, a nested frailty model is fitted.

Value

The following components are included in a 'frailtyPenal' object for each model.

- b sequence of the corresponding estimation of the coefficients for the hazard functions (parametric or semiparametric), the random effects variances and the regression coefficients.

call	The code used for the model.
formula	the formula part of the code used for the model.
coef	the regression coefficients.
cross.Val	Logical value. Is cross validation procedure used for estimating the smoothing parameters in the penalized likelihood estimation?
DoF	Degrees of freedom associated with the "kappa".
formula	the formula part of the code used for the model.
groups	the maximum number of groups used in the fit.
kappa	A vector with the smoothing parameters in the penalized likelihood estimation corresponding to each baseline function as components.
lam	matrix of hazard estimates and confidence bands.
lam2	the same value as lam for the second stratum.
loglikPenal	the complete marginal penalized log-likelihood in the semiparametric case.
loglik	the marginal log-likelihood in the parametric case.
n	the number of observations used in the fit.
n.events	the number of events observed in the fit.
n.iter	number of iterations needed to converge.
n.knots	number of knots for estimating the baseline functions in the penalized likelihood estimation.
n.strat	number of stratum.
surv	matrix of baseline survival estimates and confidence bands.
surv2	the same value as surv for the second stratum.
varH	the variance matrix of all parameters before positivity constraint transformation. Thenafter, the delta method is needed to obtain the estimated variance parameters. That is why some variances don't match with the printed values at the end of the model.
varHIH	the robust estimation of the variance matrix of all parameters.
x1	vector of times where both survival and hazard function are estimated. By default seq(0,max(time),length=99), where time is the vector of survival times.
x2	vector of times for the second stratum or the terminal event (see x1 value).
type.of.hazard	Type of hazard functions (0:"Splines", "1:Piecewise", "2:Weibull").
type.of.Piecewise	Type of Piecewise hazard functions (1:"percentile", 0:"equidistant").
nbintervR	Number of intervals (between 1 and 20) for the parametric hazard functions ("Piecewise-per", "Piecewise-equi").
npar	number of parameters.
nvar	number of explanatory variables.
noVar	indicator of explanatory variables.

LCV the approximated likelihood cross-validation criterion in the semiparametric case (with H minus the converged hessian matrix, and $l(.)$ the full log-likelihood).

$$LCV = \frac{1}{n}(\text{trace}(H_{pl}^{-1}H) - l(.))$$

AIC the Akaike information Criterion for the parametric case.

$$AIC = \frac{1}{n}(np - l(.))$$

n.knots.temp initial value for the number of knots.
 shape.weib shape parameter for the Weibull hazard function.
 scale.weib scale parameter for the Weibull hazard function.
 martingale.res martingale residuals for each cluster.
 martingaleCox martingale residuals for observation in the Cox model.
 Frailty Logical value. Was model with frailties fitted ?
 frailty.pred empirical Bayes prediction of the frailty term (ie, using conditional posterior distributions).
 frailty.var variance of the empirical Bayes prediction of the frailty term (only for gamma frailty models).
 frailty.sd standard error of the frailty empirical Bayes prediction (only for gamma frailty models).
 global_chisq a vector with the values of each multivariate Wald test.
 dof_chisq a vector with the degree of freedom for each multivariate Wald test.
 global_chisq.test a binary variable equals to 0 when no multivariate Wald is given, 1 otherwise.
 p.global_chisq a vector with the p_values for each global multivariate Wald test.
 names.factor Names of the "as.factor" variables.
 Xlevels vector of the values that factor might have taken.
 contrasts type of contrast for factor variable.

The following components are specific to **shared** models.

theta variance of the gamma frailty parameter ($\text{Var}(\omega_i)$)
 sigma2 variance of the log-normal frailty parameter ($\text{Var}(\eta_i)$)
 linear.pred linear predictor: uses simply "Beta'X" in the cox proportional hazard model or "Beta'X + log w_i" in the shared gamma frailty models, otherwise uses "Beta'X + w_i" for log-normal frailty distribution.
 BetaTpsMat matrix of time varying-effects and confidence bands (the first column used for absciss of times)

The following components are specific to **joint** models.

theta variance of the gamma frailty parameter ($\text{Var}(\omega_i)$) or ($\text{Var}(u_i)$)

sigma2	variance of the log-normal frailty parameter ($\text{Var}(\eta_i)$) or ($\text{Var}(v_i)$)
alpha	the coefficient associated with the frailty parameter in the terminal hazard function.
nbintervR	Number of intervals (between 1 and 20) for the recurrent parametric hazard functions ("Piecewise-per", "Piecewise-equi").
nbintervDC	Number of intervals (between 1 and 20) for the death parametric hazard functions ("Piecewise-per", "Piecewise-equi").
nvar	A vector with the number of covariates of each type of hazard function as components.
nvarRec	number of recurrent explanatory variables.
nvarEnd	number of death explanatory variables.
noVar1	indicator of recurrent explanatory variables.
noVar2	indicator of death explanatory variables.
martingale.res	martingale residuals for each cluster (recurrent).
martingaledeath.res	martingale residuals for each cluster (death).
linear.pred	linear predictor: uses "Beta'X + log w_i" in the gamma frailty model, otherwise uses "Beta'X + eta_i" for log-normal frailty distribution
lineardeath.pred	linear predictor for the terminal part : "Beta'X + alpha.log w_i" for gamma, "Beta'X + alpha.eta_i" for log-normal frailty distribution
Xlevels	vector of the values that factor might have taken for the recurrent part.
contrasts	type of contrast for factor variable for the recurrent part.
Xlevels2	vector of the values that factor might have taken for the death part.
contrasts2	type of contrast for factor variable for the death part.
BetaTpsMat	matrix of time varying-effects and confidence bands for recurrent event (the first column used for absciss of times of recurrence)
BetaTpsMatDc	matrix of time varying-effects and confidence bands for terminal event (the first column used for absciss of times of death)

The following components are specific to **nested** models.

alpha	variance of the cluster effect ($\text{Var}(v_i)$)
eta	variance of the subcluster effect ($\text{Var}(w_{ij})$)
subgroups	the maximum number of subgroups used in the fit.
frailty.pred.group	empirical Bayes prediction of the frailty term by group.
frailty.pred.subgroup	empirical Bayes prediction of the frailty term by subgroup.
linear.pred	linear predictor: uses "Beta'X + log v_i.w_ij".
subbyg	subgroup by group.
n.strat	A vector with the number of covariates of each type of hazard function as components.

Note

From a prediction aim, we recommend you to input a data sorted by the group variable with numerical numbers from 1 to n (number of groups). In case of a nested model, we recommend you to input a data sorted by the group variable then sorted by the subgroup variable both with numerical numbers from 1 to n (number of groups) and from 1 to m (number of subgroups). "Kappa" (kappa1 and kappa2) and "n.knots" are the arguments that the user have to change if the fitted model does not converge. "n.knots" takes integer values between 4 and 20. But with n.knots=20, the model would take a long time to converge. So, usually, begin first with n.knots=7, and increase it step by step until it converges. "Kappa" only takes positive values. So, choose a value for Kappa (for instance 10000), and if it does not converge, multiply or divide this value by 10 or 5 until it converges.

References

- V. Rondeau, Y. Mazroui and J. R. Gonzalez (2012). Frailtypack: An R package for the analysis of correlated survival data with frailty models using penalized likelihood estimation or parametric estimation. *Journal of Statistical Software* **47**, 1-28.
- V. Rondeau, J.P. Pignon, S. Michiels (2011). A joint model for the dependance between clustered times to tumour progression and deaths: A meta-analysis of chemotherapy in head and neck cancer. *Statistical methods in medical research* **897**, 1-19.
- V. Rondeau, S. Mathoulin-Pellissier, H. Jacqmin-Gadda, V. Brouste, P. Soubeyran (2007). Joint frailty models for recurring events and death using maximum penalized likelihood estimation: application on cancer events. *Biostatistics* **8**, 4, 708-721.
- V. Rondeau, L. Filleul, P. Joly (2006). Nested frailty models using maximum penalized likelihood estimation. *Statistics in Medecine*, **25**, 4036-4052.
- V. Rondeau, D. Commenges, and P. Joly (2003). Maximum penalized likelihood estimation in a gamma-frailty model. *Lifetime Data Analysis* **9**, 139-153.
- McGilchrist CA and Aisbett CW (1991). Regression with frailty in survival analysis. *Biometrics* **47**, 461-466.
- D. Marquardt (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, 431-441.

See Also

[SurvIC](#), [subcluster](#), [terminal](#), [num.id](#), [timedep](#)

Examples

```
## Not run:

###--- COX proportional hazard model (SHARED without frailties) ---###
###--- estimated with penalized likelihood ---###

data(kidney)
frailtyPenal(Surv(time,status)~sex+age,
n.knots=12,kappa1=10000,data=kidney,Frailty=FALSE)

###--- Shared Frailty model ---###
```

```

frailtyPenal(Surv(time,status)~cluster(id)+sex+age,
n.knots=12,kappa1=10000,data=kidney,Frailty=TRUE)

#-- with an initialisation of regression coefficients

frailtyPenal(Surv(time,status)~cluster(id)+sex+age,
n.knots=12,kappa1=10000,data=kidney,Frailty=TRUE,B=c(-1.44,0))

#-- with truncated data

data(dataNested)

frailtyPenal(Surv(t1,t2,event) ~ cluster(group),
data=dataNested,n.knots=10,kappa1=10000,
Frailty=TRUE,cross.validation=TRUE,recurrentAG=FALSE)

#-- stratified analysis

data(readmission)
frailtyPenal(Surv(time,event)~cluster(id)+dukes+strata(sex),
n.knots=10,kappa1=10000,kappa2=10000,data=readmission,
Frailty=TRUE)

#-- recurrentAG=TRUE

frailtyPenal(Surv(t.start,t.stop,event)~cluster(id)+sex+dukes+
charlson,data=readmission,Frailty=TRUE,
n.knots=6,kappa1=100000,recurrentAG=TRUE)

#-- cross.validation=TRUE

frailtyPenal(Surv(t.start,t.stop,event)~cluster(id)+sex+dukes+
charlson,data=readmission, Frailty=TRUE,n.knots=6,
kappa1=5000,recurrentAG=TRUE,cross.validation=TRUE)

#-- log-normal distribution

frailtyPenal(Surv(t.start,t.stop,event)~sex+dukes+charlson+cluster(id),
data=readmission, Frailty=TRUE,n.knots=6,kappa1=5000,
recurrentAG=TRUE,RandDist="LogN")

###--- Joint Frailty model (recurrent and terminal events) ---###

data(readmission)
#-- Gap-time
modJoint.gap <- frailtyPenal(Surv(time,event)~cluster(id)+sex+dukes+charlson+
terminal(death),formula.terminalEvent=~sex+dukes+charlson,
data=readmission,n.knots=14,kappa1=9.55e+9,kappa2=1.41e+12,
Frailty=TRUE, joint=TRUE, recurrentAG=FALSE)

#-- Calendar time
modJoint.calendar <- frailtyPenal(Surv(t.start,t.stop,event)~cluster(id)+sex

```



```

+dukes+charlson+terminal(death),formula.terminalEvent=~sex
+dukes+charlson,data=readmission,n.knots=10,kappa1=9.55e9,
kappa2=1.41e12,Frailty=TRUE, joint=TRUE, recurrentAG=TRUE)

#-- log-normal distribution

modJoint.log <- frailtyPenal(Surv(t.start,t.stop,event)~cluster(id)+sex
+dukes+charlson+terminal(death),formula.terminalEvent=~sex
+dukes+charlson,data=readmission,n.knots=10,kappa1=9.55e9,
kappa2=1.41e12,Frailty=TRUE, joint=TRUE, recurrentAG=TRUE,
RandDist="LogN")

###--- Joint Frailty model for clustered data ---###

#-- here is a generated cluster (31 clusters of 13 subjects)
readmission <- transform(readmission,group=id%31+1)

joi.clus <- frailtyPenal(Surv(t.start,t.stop,event)~cluster(group)+
num.id(id)+dukes+charlson+sex+chemo+terminal(death),
formula.terminalEvent=~dukes+charlson+sex+chemo,
data=readmission,Frailty=TRUE, joint=TRUE, recurrentAG=TRUE,
n.knots=10,kappa1=2.11e+08,kappa2=9.53e+11)

###--- Nested Frailty model ---###

data(dataNested)
modClu <- frailtyPenal(Surv(t1,t2,event)~cluster(group)+
subcluster(subgroup)+cov1+cov2,Frailty=TRUE,
data=dataNested,n.knots=8,kappa1=50000)

modClu.str <- frailtyPenal(Surv(t1,t2,event)~cluster(group)+
subcluster(subgroup)+cov1+strata(cov2),Frailty=TRUE,
data=dataNested,n.knots=8,kappa1=50000,kappa2=50000)

## End(Not run)

```

hazard

Hazard function.

Description

Let t be a continuous variable, we determine the value of the hazard function to t after run fit.

Usage

```
hazard(t, ObjFrailty)
```

Arguments

t time for hazard function.
 ObjFrailty an object from the frailtypack fit.

Value

return the value of hazard function in t.

Examples

```
## Not run:

#-- a fit Shared
data(readmission)
fit.shared <- frailtyPenal(Surv(time,event)~dukes+cluster(id)+
  strata(sex),n.knots=10,kappa1=10000,kappa2=10000,data=readmission,
  Frailty=TRUE)

#-- calling survival
hazard(20,fit.shared)

## End(Not run)
```

num.id

Identify individuals in Joint model for clustered data

Description

This is a special function used in addition to the `cluster()` function in the context of survival joint models for clustered data. This function identifies subject index. It is used on the right hand side of a 'frailtyPenal' formula. Using `num.id()` in a formula implies that a joint frailty model for clustered data is fitted (Rondeau et al. 2011).

Usage

```
num.id(x)
```

Arguments

x A character or numeric variable which is supposed to indicate the variable identifying individuals

References

V. Rondeau, J.P. Pignon, S. Michiels (2011). A joint model for the dependance between clustered times to tumour progression and deaths: A meta-analysis of chemotherapy in head and neck cancer. *Statistical methods in medical research* **897**, 1-19.

See Also

[frailtyPenal](#)

Examples

```
## Not run:

data(readmission)
#-- here is a generated cluster (31 clusters of 13 subjects)
readmission <- transform(readmission,group=id%31+1)

###--- Joint Frailty model for clustered data ---###

#-- splines
joi.clus <- frailtyPenal(Surv(t.start,t.stop,event)~
  cluster(group)+num.id(id)+dukes+charlson+sex+chemo+terminal(death),
  formula.terminalEvent=~dukes+charlson+sex+chemo,
  data=readmission,Frailty=TRUE,joint=TRUE,recurrentAG=TRUE,
  n.knots=10,kappa1=2.11e+08,kappa2=9.53e+11)

## End(Not run)
```

plot.additivePenal	<i>Plot Method for an Additive frailty model.</i>
--------------------	---

Description

Plots estimated baseline survival and hazard functions of an additive frailty model, more generally of a class ‘additivePenal’ object. Confidence bands are allowed.

Usage

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

Arguments

`x` A fitted additive frailty model (output from calling `additivePenal`)
`...` Other graphical parameters like those in [plot.frailtyPenal](#)

Value

Print a plot of HR and survival function of a class `additivePenal` object

See Also

[additivePenal](#)

Examples

```
## Not run:

data(dataAdditive)

modAdd <- additivePenal(Surv(t1,t2,event)~cluster(group)+var1+slope(var1),
  correlation=TRUE,data=dataAdditive,n.knots=8,kappa1=862,hazard="Splines")

#-- 'var1' is boolean as a treatment variable

plot(modAdd)

## End(Not run)
```

`plot.frailtyPenal` *Plot Method for a Shared frailty model.*

Description

Plots estimated baseline survival and hazard functions from an object of class 'frailtyPenal'. Confidence bands are allowed.

Usage

```
## S3 method for class 'frailtyPenal'
plot(x, type.plot = "hazard", conf.bands=TRUE, pos.legend="topright",
  cex.legend=0.7, main, color, ...)
```

Arguments

x	A shared frailty model, more generally a frailtyPenal class object (output from calling frailtyPenal function).
type.plot	a character string specifying the type of curve. Possible value are "hazard", or "survival". The default is "hazard". Only the first words are required, e.g "haz", "su"
conf.bands	Logical value. Determines whether confidence bands will be plotted. The default is to do so.
pos.legend	The location of the legend can be specified by setting this argument to a single keyword from the list '"bottomright"', '"bottom"', '"bottomleft"', '"left"', '"topleft"', '"top"', '"topright"', '"right"' and '"center"'. The default is '"topright"'
cex.legend	character expansion factor <i>relative</i> to current 'par("cex")'. Default is 0.7
main	title of plot
color	color of the curve (integer or color name in quotation marks)
...	other unused arguments

Value

Print a plot of a shared frailty model.

See Also

[frailtyPenal](#)

Examples

```
## Not run:

data(readmission)

###--- Shared frailty model ---###

modSha <- frailtyPenal(Surv(time,event)~as.factor(dukes)+cluster(id),
  n.knots=10,kappa1=10000,data=readmission,Frailty=TRUE,hazard="Splines")

plot(modSha,type="surv",conf=FALSE)

###--- Cox proportional hazard model ---###

modCox <- frailtyPenal(Surv(time,event)~as.factor(dukes),n.knots=10,
  kappa1=10000,data=readmission,Frailty=FALSE,hazard="Splines")

plot(modCox)

#-- no confidence bands
plot(modSha,conf.bands=FALSE)
plot(modCox,conf.bands=FALSE)
```

```
## End(Not run)
```

plot.jointPenal	<i>Plot Method for a Joint frailty model.</i>
-----------------	---

Description

Plots estimated baseline survival and hazard functions of a joint frailty model (output from an object of class 'frailtyPenal' for joint frailty models) for each type of event (terminal or recurrent). Confidence bands are allowed.

Usage

```
## S3 method for class 'jointPenal'
plot(x, event = "both", type.plot = "hazard", conf.bands = FALSE,
     pos.legend="topright", cex.legend=0.7, ylim, main, color1, color2, ...)
```

Arguments

x	A joint model, or more generally an object of class frailtyPenal for Joint frailty model (output from calling frailtyPenal function).
event	a character string specifying the type of curve. Possible value are "terminal", "recurrent", or "both". The default is "both".
type.plot	a character string specifying the type of curve. Possible value are "hazard", or "survival". The default is "hazard". Only the first words are required, e.g "haz", "su"
conf.bands	logical value. Determines whether confidence bands will be plotted. The default is to do so.
pos.legend	The location of the legend can be specified by setting this argument to a single keyword from the list '"bottomright"', '"bottom"', '"bottomleft"', '"left"', '"topleft"', '"top"', '"topright"', '"right"' and '"center"'. The default is '"topright"'
cex.legend	character expansion factor <i>relative</i> to current 'par("cex")'. Default is 0.7
ylim	y-axis limits
main	plot title
color1	color of the curve (integer or color name in quotation marks)
color2	color of the second curve if both recurrent and terminal events are plotted
...	other unused arguments

Value

Print a plot of the baseline survival or hazard functions for each type of event or both with the confidence bands or not (conf.bands argument)

See Also

[frailtyPenal](#)

Examples

```
## Not run:

data(readmission)

#-- Gap-time
modJoint.gap <- frailtyPenal(Surv(time,event)~cluster(id)+sex+dukes+
  charlson+terminal(death),formula.terminalEvent=~sex+dukes+charlson,
  data=readmission,n.knots=14,kappa1=9.55e+9,kappa2=1.41e+12,
  Frailty=TRUE, joint=TRUE)

#-- It takes around 1 minute to converge --#

plot(modJoint.gap,type.plot="haz",event="recurrent",conf.bands=TRUE)
plot(modJoint.gap,type.plot="haz",event="terminal",conf.bands=TRUE)
plot(modJoint.gap,type.plot="haz",event="both",conf.bands=TRUE)
plot(modJoint.gap,type.plot="su",event="recurrent",conf.bands=TRUE)
plot(modJoint.gap,type.plot="su",event="terminal",conf.bands=TRUE)
plot(modJoint.gap, event="terminal",type.plot="su",conf.bands=TRUE)
plot(modJoint.gap,type.plot="su","terminal",conf.bands=TRUE)

## End(Not run)
```

plot.nestedPenal

Plot Method for a Nested frailty model.

Description

Plots estimated baseline survival and hazard functions (output from an object of class ‘frailtyPenal’ for nested frailty models). Confidence bands are allowed.

Usage

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

Arguments

- `x` A nested model, more generally an object of class `frailtyPenal` for Nested frailty models (output from calling `frailtyPenal` function).
- `...` Other graphical parameters like those in [plot.frailtyPenal](#)

Value

Print a plot of the baseline survival or hazard functions with the confidence bands or not (`conf.bands` argument)

See Also

[frailtyPenal](#)

Examples

```
## Not run:

data(dataNested)
modNested <- frailtyPenal(Surv(t1,t2,event)~cluster(group)+
  subcluster(subgroup)+cov1+cov2,data=dataNested,n.knots=8,
  kappa1=50000,hazard="Splines",Frailty=TRUE)

plot(modNested,conf.bands=FALSE)

## End(Not run)
```

plot.prediction	<i>Plot Method for predictions.</i>
-----------------	-------------------------------------

Description

Probabilites of events and deaths. Confidence bands are allowed.

Usage

```
## S3 method for class 'predFrailty'
plot(x, conf.bands=FALSE, pos.legend="topleft", cex.legend=0.7, ...)
```


Arguments

<code>x</code>	An object from the 'prediction' function
<code>conf.bands</code>	Logical value. Determines whether confidence bands will be plotted. The default is FALSE.
<code>pos.legend</code>	The location of the legend can be specified by setting this argument to a single keyword from the list '"bottomright"', '"bottom"', '"bottomleft"', '"left"', '"topleft"', '"top"', '"topright"', '"right"' and '"center"'. The default is '"topright"'
<code>cex.legend</code>	size of the legend. Default is 0.7
<code>...</code>	Other unused arguments

Value

Print as many plots as there are individuals with their probabilities to die.

Examples

```
## Not run:

data(readmission)

##-- predictions on a Cox proportionnal hazard model --##
cox <- frailtyPenal(Surv(time,event)~sex+dukes,
  n.knots=10,kappa1=10000,data=readmission,Frailty=FALSE)

##-- construction of the dataframe for predictions
datapred <- data.frame(sex=0,dukes=0)
datapred$sex <- as.factor(datapred$sex)
levels(datapred$sex)<- c(1,2)
datapred$dukes <- as.factor(datapred$dukes)
levels(datapred$dukes)<- c(1,2,3)
datapred[1,] <- c(1,2)
datapred[2,] <- c(2,3)

##-- prediction of the event for two patients, between time 100 and time 2000
pred.cox <- prediction(cox,datapred,100,50,2000)
plot(pred.cox)

## End(Not run)
```

Description

For Cox proportionnal hazard model

A predictive probability of event between t and horizon time $t+h$.

$$P(t, t+h) = \frac{S_i(t) - S_i(t+h)}{S_i(t)} = 1 - \left(\frac{S_0(t+h)}{S_0(t)} \right)^{\exp(\beta' Z_i)}$$

For Shared Frailty model for clustered (not recurrent) events

Two kinds of predictive probabilities can be calculated:

- a conditional predictive probability of event between t and horizon time $t+h$, i.e. given a specific group

$$P^{cond}(t, t+h) = \frac{S_{ij}(t) - S_{ij}(t+h)}{S_{ij}(t)} = 1 - \left(\frac{S_0(t+h)}{S_0(t)} \right)^{\omega_i \exp(\beta' Z_{ij})}$$

- a marginal predictive probability of event between t and horizon time $t+h$, i.e. averaged over the population

$$P^{marg}(t, t+h) = 1 - \left(\frac{1 + \theta H_0(t) \exp(\beta' Z_{ij}(t))}{1 + \theta H_0(t+h) \exp(\beta' Z_{ij}(t))} \right)^{1/\theta}$$

For Joint Frailty model

You can predict risk of death knowing patients' prognosis i.e. predicting the probability of death in a specific time window given the history of the patient before the time of prediction t . The history of the patient is the information on covariates before time t , but also the number of recurrences and the time of occurrences. Three types of probability are computed:

- a probability of death between t and $t+h$ given that the patient has exactly J recurrences before t

$$P^1(t, t+h) = P(D_i \leq t+h | D_i > t, H_i^{J,1}) = \frac{\int_0^\infty [S_i^D(t) - S_i^D(t+h)] (u_i)^J S_{i(J+1)}^R(t) g(u_i) du_i}{\int_0^\infty S_i^D(t) (u_i)^J S_{i(J+1)}^R(t) g(u_i) du_i}$$

- a probability of death between t and $t+h$ given that the patient has at least J recurrences before t

$$P^2(t, t+h) = P(D_i \leq t+h | D_i > t, H_i^{J,2}) = \frac{\int_0^\infty [S_i^D(t) - S_i^D(t+h)] (u_i)^J S_{iJ}^R(X_{iJ}) g(u_i) du_i}{\int_0^\infty S_i^D(t) (u_i)^J S_{iJ}^R(X_{iJ}) g(u_i) du_i}$$

- a probability of death between t and $t+h$ considering the recurrence history only in the parameters estimation. It corresponds to the average probability of death between t and $t+h$ for a patient with these given characteristics.

$$P^3(t, t+h) = P(D_i \leq t+h | D_i > t) = \frac{\int_0^\infty [S_i^D(t) - S_i^D(t+h)] g(u_i) du_i}{\int_0^\infty S_i^D(t) g(u_i) du_i}$$

Usage

```
prediction(fit, data, predTime, horizon, predMax, type="marginal",
          group, MC.sample=0)
```

Arguments

<code>fit</code>	A frailtyPenal or jointPenal object
<code>data</code>	Dataframe for the prediction. See Details.
<code>predTime</code>	First time of predictions
<code>horizon</code>	Length of the split
<code>predMax</code>	Last time of predictions
<code>type</code>	Only for shared frailty model, type of the prediction "marginal" or "conditional". Need to specify argument 'group' if type="conditional". Marginal by default
<code>group</code>	Only for shared frailty model, the group on which you want to make the conditional prediction (its frailty value will be used)
<code>MC.sample</code>	Number of samples used to calculate confidence bands with a Monte-Carlo method. If MC.sample=0 (default value), no confidence bands calculated.

Details

The dataframe building is an important step. It will contain profiles of patient on which you want to do predictions. To make predictions on a Cox proportionnal hazard or a shared frailty model, only covariates need to be included. You have to distinguish between numerical and categorical variables (factors). If we fit a shared frailty model with two covariates sex (factor 1:male 2:female) and age (numeric), here is the associated dataframe for the three profiles of prediction.

```
datapred <- data.frame(sex=0,age=0)
datapred$sex <- as.factor(datapred$sex)
levels(datapred$sex)<- c(1,2)
datapred[1,] <- c(1,40) # man 40 years old
datapred[2,] <- c(2,45) # woman 45 years old
datapred[3,] <- c(1,60) # man 60 years old
```

If you want to use the prediction function on a joint frailty model, the construction will be a little bit different. In this case, the prediction for the terminal event takes into account covariates but also history of relapse times for a patient. You have to create as much rows in the dataframe as there are relapses before the prediction time for a specific patient. If you did your joint model in gap-time using `Surv(time,event)`, relapse times correspond to the "time" variable. In case of calendar-time with `Surv(start,stop,event)`, relapse times correspond to the "stop" variable. For patients without relapses, "time" or "stop" variable should be equal to 0. Finally, the same cluster variable name needs to be in the joint model and in the dataframe for predictions. For instance, we observe relapses of a disease and fit a joint model adjusted for two covariates sex (1:male 2:female) and chemo (treatment by chemotherapy 1:no 2:yes). We describe 3 different profiles of prediction: 1) a man with four relapses at 100, 200, 300 and 400 days, 2) a man with only one relapse at 1000 days, 3) a woman without relapse, all treated by chemotherapy.

```

datapred <- data.frame(time=0,id=0,sex=0,chemo=0)
datapred$sex <- as.factor(datapred$sex)
levels(datapred$sex) <- c(1,2)
datapred$chemo <- as.factor(datapred$chemo)
levels(datapred$chemo) <- c(1,2)
datapred[1,] <- c(100,1,1,2) # first relapse of the patient 1
datapred[2,] <- c(200,1,1,2) # second relapse of the patient 1
datapred[3,] <- c(300,1,1,2) # third relapse of the patient 1
datapred[4,] <- c(400,1,1,2) # fourth relapse of the patient 1
datapred[5,] <- c(1000,2,1,2) # one relapse at 1000 days for patient 2
datapred[6,] <- c(0,3,2,2) # patient 3 did not relapse

```

Value

The following components are included in a 'predFrailty' object obtained by using prediction function for Cox proportionnal hazard and shared frailty model.

time	Vector of prediction times
pred	Predictions estimated for each profile
icproba	Logical value. Were confidence bands estimated ?
predLow	Lower limit of confidence interval for each prediction
predHigh	Upper limit of confidence interval for each prediction
type	Type of prediction probability (marginal or conditional)
horizon	Prediction horizon
group	For conditional probability, the group on which you make predictions

The following components are included in a 'predJoint' object obtained by using prediction function for joint frailty model.

npred	Number of individual predictions
time	Vector of prediction times
group	Id of each patient
proba1	Estimation of probability of type 1: exactly j recurrences
proba2	Estimation of probability of type 2: at least j recurrences
proba3	Estimation of probability of type 3
icproba	Logical value. Were confidence bands estimated ?
probalow1	Lower limit of Monte-Carlo confidence interval for probability of type 1
probahigh1	Upper limit of Monte-Carlo confidence interval for probability of type 1
probalow2	Lower limit of Monte-Carlo confidence interval for probability of type 2
probahigh2	Upper limit of Monte-Carlo confidence interval for probability of type 2
probalow3	Lower limit of Monte-Carlo confidence interval for probability of type 3
probahigh3	Upper limit of Monte-Carlo confidence interval for probability of type 3

References

A. Mauguen, B. Rachet, S. Mathoulin-Pelissier, G. MacGrogan, A. Laurent, V. Rondeau (2013). Dynamic prediction of risk of death using history of cancer recurrences in joint frailty models. *Statistics in Medicine*.

V. Rondeau, A. Laurent, A. Mauguen, P. Joly, C. Helmer. Simple or joint frailty models for clustered and interval-censored outcome with a dependant terminal event: investigating the intra-couple correlation in the risk of dementia and death. To appear.

Examples

```
## Not run:

data(readmission)

##-- predictions on a Cox proportionnal hazard model --##
cox <- frailtyPenal(Surv(time,event)~sex+dukes,
n.knots=10,kappa1=10000,data=readmission,Frailty=FALSE)

##-- construction of the dataframe for predictions
datapred <- data.frame(sex=0,dukes=0)
datapred$sex <- as.factor(datapred$sex)
levels(datapred$sex)<- c(1,2)
datapred$dukes <- as.factor(datapred$dukes)
levels(datapred$dukes)<- c(1,2,3)
datapred[1,] <- c(1,2)
datapred[2,] <- c(2,3)

##-- prediction of the event for two patients, between time 100 and time 2000
pred.cox <- prediction(cox,datapred,100,50,2000)
plot(pred.cox)

##-- predictions on a shared frailty model --##
sha <- frailtyPenal(Surv(time,event)~cluster(id)+sex+dukes,
n.knots=10,kappa1=10000,data=readmission,Frailty=TRUE)

##-- marginal prediction
pred.sha.marg <- prediction(sha,datapred,100,50,2000,type="marginal")
plot(pred.sha.marg)

##-- conditional prediction, given a specific cluster (group=5)
pred.sha.cond <- prediction(sha,datapred,100,50,2000,type="conditional",group=5)
plot(pred.sha.cond)

##-- predictions on a joint frailty model --##
joi <- frailtyPenal(Surv(t.start,t.stop,event)~cluster(id)
+sex+dukes+terminal(death),formula.terminalEvent=~sex
+dukes,data=readmission,n.knots=10,kappa1=9.55e9,
kappa2=1.41e12,Frailty=TRUE,joint=TRUE,recurrentAG=TRUE)

##-- creation of the dataframe for predictions
```

```

datapredj <- data.frame(t.stop=0,id=0,sex=0,dukes=0)
datapredj$sex <- as.factor(datapredj$sex)
levels(datapredj$sex) <- c(1,2)
datapredj$dukes <- as.factor(datapredj$dukes)
levels(datapredj$dukes) <- c(1,2,3)
datapredj[1,] <- c(100,1,1,2)
datapredj[2,] <- c(200,1,1,2)
datapredj[3,] <- c(300,1,1,2)
datapredj[4,] <- c(400,1,1,2)
datapredj[5,] <- c(380,2,1,2)

#-- only recurrences observed before the prediction
#-- time should be included in the 'datapred'
pred.joint <- prediction(joi,datapredj,500,50,2000)
plot(pred.joint)

## End(Not run)

```

print.additivePenal	<i>Print a Short Summary of parameter estimates of an additive frailty model</i>
---------------------	--

Description

Prints a short summary of the parameter estimates of an additive frailty model or more generally of an 'additivePenal' object

Usage

```

## S3 method for class 'additivePenal'
print(x, digits = max(options())$digits - 4, 6), ...)

```

Arguments

x	the result of a call to the additivePenal function
digits	number of digits to print
...	other unused arguments

Value

Print the parameter estimates of the survival or hazard functions.

See Also

[additivePenal](#)

print.Cmeasures	<i>Print a short summary of results of Cmeasure function.</i>
-----------------	---

Description

Print a short summary of results of the concordance measure estimated by the Cmeasure function.

Usage

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

Arguments

x	a Cmeasures object.
...	Other unused arguments

Value

Print concordance measures estimated.

See Also

[Cmeasures](#)

print.frailtyPenal	<i>Print a Short Summary of parameter estimates of a shared frailty model</i>
--------------------	---

Description

Prints a short summary of parameter estimates of a 'frailtyPenal' object

Usage

```
## S3 method for class 'frailtyPenal'
print(x, digits = max(options())$digits - 4, 6), ...)
```

Arguments

x	the result of a call to the frailtyPenal function.
digits	number of digits to print.
...	other unusued arguments.

Value

Print the parameter estimates of the survival or hazard functions.

See Also

[frailtyPenal](#)

<code>print.jointPenal</code>	<i>Print a Short Summary of parameter estimates of a joint frailty model</i>
-------------------------------	--

Description

Prints a short summary of parameter estimates of a joint frailty model, or more generally an object of class 'frailtyPenal' for joint frailty models.

Usage

```
## S3 method for class 'jointPenal'  
print(x, digits = max(options())$digits - 4, 6), ...)
```

Arguments

<code>x</code>	the result of a call to the <code>jointPenal</code> function
<code>digits</code>	number of digits to print
<code>...</code>	other unused arguments

Value

Print, separately for each type of event (recurrent and terminal), the parameter estimates of the survival or hazard functions.

See Also

[frailtyPenal](#)

print.nestedPenal	<i>Print a Short Summary of parameter estimates of a nested frailty model</i>
-------------------	---

Description

Prints a short summary of parameter estimates of a nested frailty model

Usage

```
## S3 method for class 'nestedPenal'
print(x, digits = max(options())$digits - 4, 6), ...)
```

Arguments

x	the result of a call to the frailtyPenal function for nested frailty models
digits	number of digits to print
...	other unused arguments

Value

n	the number of observations used in the fit.
n.groups	the maximum number of groups used in the fit
n.events	the number of events observed in the fit
eta	variance of the subcluster effect ($Var(w_{ij})$)
theta	variance of the cluster effect ($Var(v_i)$)
coef	the coefficients of the linear predictor, which multiply the columns of the model matrix.
SE(H)	the standard error of the estimates deduced from the variance matrix of theta and of the coefficients.
SE(HIH)	the standard error of the estimates deduced from the robust estimation of the variance matrix of theta and of the coefficients.
p	p-value

See Also

[frailtyPenal](#)

<code>print.prediction</code>	<i>Print a short summary of results of prediction function.</i>
-------------------------------	---

Description

Print a short summary of results of prediction function.

Usage

```
## S3 method for class 'predFrailty'  
print(x, digits = 3, ...)
```

Arguments

<code>x</code>	An object from the 'prediction' function
<code>digits</code>	Number of digits to print
<code>...</code>	Other unused arguments

Value

Print the probabilities estimated.

See Also

[prediction](#)

<code>readmission</code>	<i>Rehospitalization colorectal cancer</i>
--------------------------	--

Description

This contains rehospitalization times after surgery in patients diagnosed with colorectal cancer

Usage

```
data(readmission)
```

Format

This data frame contains the following columns:

id identificator of each subject. Repeated for each recurrence
enum which readmission
t.start start of interval (0 or previous recurrence time)
t.stop recurrence or censoring time
time interocurrence or censoring time
event rehospitalization status. All event are 1 for each subject excepting last one that it is 0
chemo Did patient receive chemotherapy? 1: No; 2:Yes
sex gender: 1:Males 2:Females
dukes Dukes' tumoral stage: 1:A-B; 2:C 3:D
charlson Comorbidity Charlson's index. Time-dependent covariate. 0: Index 0; 1: Index 1-2; 3: Index ≥ 3
death death indicator. 1:dead and 0:alive

Source

Gonzalez, JR., Fernandez, E., Moreno, V., Ribes, J., Peris, M., Navarro, M., Cambray, M. and Borras, JM (2005). Sex differences in hospital readmission among colorectal cancer patients. *Journal of Epidemiology and Community Health*, **59**, 6, 506-511.

slope

Identify variable associated with the random slope

Description

This is a special function used in the context of survival additive models. It identifies the variable which is in interaction with the random slope (v_i). Generally, this variable is the treatment variable. Using `interaction()` in a formula implies that an additive frailty model is fitted.

Usage

`slope(x)`

Arguments

x A factor, a character or a numerical variable

Value

x The variable in interaction with the random slope

Note

It is necessary to specify which variable is in interaction with the random slope, even if only one explanatory variable is included in the model.

See Also

[additivePenal](#)

Examples

```
## Not run:

data(dataAdditive)

##-- Additive with one covariate --##

modAdd1cov <- additivePenal(Surv(t1,t2,event)~cluster(group)+var1+
  slope(var1),data=dataAdditive,n.knots=8,kappa1=10000,hazard="Splines")

##-- Additive with two covariates --##

set.seed(1234)
dataAdditive$var2 <- rbinom(nrow(dataAdditive),1,0.5)

modAdd2cov <- additivePenal(Surv(t1,t2,event)~cluster(group)+var1+
  var2+slope(var1),data=dataAdditive,n.knots=8,kappa1=10000,
  hazard="Splines")

##-- Additive with 2 covariates and stratification --##

dataAdditive$var2 <- rbinom(nrow(dataAdditive),1,0.5)

modAddstrat <- additivePenal(Surv(t1,t2,event)~cluster(group)+
  strata(var2)+var1+slope(var1),data=dataAdditive,n.knots=8,
  kappa1=10000,kappa2=10000,hazard="Splines")

## End(Not run)
```

subcluster

Identify subclusters

Description

This is a special function used in the context of survival nested models. It identifies correlated groups of observations within other groups defined by using 'cluster' function from 'survival' package, and is used on the right hand side of 'frailtyPenal' formula for fitting a nested model. Using subcluster() in a formula implies that a nested frailty model is estimated.

Usage

```
subcluster(x)
```

Arguments

x A character, factor, or numeric variable which is supposed to indicate the variable subgroup

Value

x A variable identified as a subcluster

See Also

[frailtyPenal](#)

Examples

```
## Not run:

data(dataNested)
modClu <- frailtyPenal(Surv(t1,t2,event)~cluster(group)+
  subcluster(subgroup)+cov1+cov2,Frailty=TRUE,data=dataNested,
  n.knots=8,kappa1=50000,kappa2=50000,hazard="Splines")

print(modClu)

## End(Not run)
```

summary.additivePenal *summary of parameter estimates of an additive frailty model*

Description

This function returns hazard ratios (HR) and its confidence intervals

Usage

```
## S3 method for class 'additivePenal'
summary(object, level = 0.95, len = 6, d = 2, lab="hr", ...)
```

Arguments

object	output from a call to additivePenal.
level	significance level of confidence interval. Default is 95%.
d	the desired number of digits after the decimal point. Default of 6 digits is used.
len	the total field width. Default is 6.
lab	label of printed results.
...	other unused arguments.

Value

Prints HR and its confidence intervals for each covariate. Confidence level is allowed (level argument)

See Also

[additivePenal](#)

Examples

```
## Not run:

data(dataAdditive)

modAdd <- additivePenal(Surv(t1,t2,event)~cluster(group)+var1+slope(var1),
  correlation=TRUE,data=dataAdditive,n.knots=8,kappa1=862,hazard="Splines")

#- 'var1' is boolean as a treatment variable.

summary(modAdd)

## End(Not run)
```

summary.frailtyPenal *summary of parameter estimates of a shared frailty model*

Description

This function returns hazard ratios (HR) and its confidence intervals

Usage

```
## S3 method for class 'frailtyPenal'
summary(object, level = 0.95, len = 6, d = 2, lab="hr", ...)
```

Arguments

object	output from a call to frailtyPenal.
level	significance level of confidence interval. Default is 95%.
d	the desired number of digits after the decimal point. Default of 6 digits is used.
len	the total field width. Default is 6.
lab	label of printed results.
...	other unused arguments.

Value

Prints HR and its confidence intervals. Confidence level is allowed (level argument)

See Also

[frailtyPenal](#)

Examples

```
## Not run:

data(kidney)

##-- Shared frailty model --##

modSha <- frailtyPenal(Surv(time,status)~age+sex+cluster(id),
  Frailty=TRUE,n.knots=8,kappa1=10000,data=kidney,hazard="Splines")

##-- Cox proportional hazard model --##

modCox <- frailtyPenal(Surv(time,status)~age+sex,n.knots=8,
  kappa1=10000,data=kidney,Frailty=FALSE,hazard="Splines")

#-- confidence interval at 95

summary(modSha)
summary(modCox)

#-- confidence interval at 99

summary(modSha,level=0.99)
summary(modCox,level=0.99)

## End(Not run)
```

summary.jointPenal	<i>summary of parameter estimates of a joint frailty model</i>
--------------------	--

Description

This function returns hazard ratios (HR) and its confidence intervals

Usage

```
## S3 method for class 'jointPenal'
summary(object, level = 0.95, len = 6, d = 2, lab="hr", ...)
```

Arguments

object	output from a call to frailtyPenal for joint models
level	significance level of confidence interval. Default is 95%.
d	the desired number of digits after the decimal point. Default of 6 digits is used.
len	the total field width. Default is 6.
lab	label of printed results.
...	other unused arguments.

Value

Prints HR and its confidence intervals for each covariate. Confidence level is allowed (level argument)

See Also

[frailtyPenal](#)

Examples

```
## Not run:

data(readmission)

##-- gap-time
modJoint.gap <- frailtyPenal(Surv(time,event)~cluster(id)+sex+dukes+
charlson+terminal(death),formula.terminalEvent=~sex+dukes+charlson,
data=readmission,n.knots=14,kappa1=9.55e+9,kappa2=1.41e+12,
Frailty=TRUE, joint=TRUE)

##-- calendar time
modJoint.calendar <- frailtyPenal(Surv(t.start,t.stop,event)~cluster(id)+
sex+dukes+charlson+terminal(death),formula.terminalEvent=~sex+dukes+charlson,
data=readmission,n.knots=10,kappa1=9.55e+9,kappa2=1.41e+12,Frailty=TRUE,
```



```

joint=TRUE,recurrentAG=TRUE)

#-- It takes around 1 minute to converge

summary(modJoint.gap)
summary(modJoint.calendar)

## End(Not run)

```

summary.nestedPenal	<i>summary of regression coefficient estimates of a nested frailty model</i>
---------------------	--

Description

This function returns hazard ratios (HR) and its confidence intervals for each regression coefficient

Usage

```

## S3 method for class 'nestedPenal'
summary(object, level = 0.95, len = 6, d = 2, lab="hr", ...)

```

Arguments

object	output from a call to nestedPenal.
level	significance level of confidence interval. Default is 95%.
d	the desired number of digits after the decimal point. Default of 6 digits is used.
len	the total field width. Default is 6.
lab	label of printed results.
...	other unused arguments.

Value

Prints HR and its confidence intervals for each regression coefficient. Confidence level is allowed (level argument)

See Also

[frailtyPenal](#)

Examples

```
## Not run:

data(dataNested)

modNested <- frailtyPenal(Surv(t1,t2,event)~cluster(group)+
  subcluster(subgroup)+cov1+cov2,Frailty=TRUE,data=dataNested,
  n.knots=8,kappa1=50000,kappa2=50000,hazard="Splines")

#- It takes 90 minutes to converge (depends on processor)

summary(modNested)

## End(Not run)
```

SurvIC	<i>Create a survival object for interval censoring and possibly left truncated data</i>
--------	---

Description

This is a function used in case of interval-censoring as a response variable in a model formula only for Cox proportional hazard or shared frailty model. Sometimes, an unobserved event might occur in a time interval [L,U]. RecurrentAG argument gets invalid with the use of SurvIC. Time dependant variables are not allowed.

Usage

```
SurvIC(t0, lower, upper, event)
```

Arguments

lower	Starting time of the interval for interval-censored data. Time of right-censoring instead.
upper	Ending time of the interval for interval-censored data. For right-censored data, lower and upper time must be equal (for numerical reason).
event	Status indicator 0=right-censored, 1=interval-censored
t0	Truncation time for left truncated data only. To be ignored otherwise.

Details

Typical usages are `SurvIC(lower, upper, event)` or `SurvIC(t0, lower, upper, event)`

Examples

```
## Not run:

data(bcos)
bcos$event <- ifelse(bcos$left!=bcos$right,1,0)

###--- Cox proportional hazard model with interval censoring ---###

cox.ic <- frailtyPenal(SurvIC(left,right,event)~treatment,
data=bcos,Frailty=FALSE,n.knots=8,kappa1=10000)

###--- Shared model with interval censoring ---###

bcos$group <- c(rep(1:20,4),1:14)

sha.ic <- frailtyPenal(SurvIC(left,right,event)~cluster(group)+
treatment,data=bcos,Frailty=TRUE,n.knots=8,kappa1=10000)

## End(Not run)
```

survival

Survival function

Description

Let t be a continuous variable, we determine the value of the survival function to t after run fit.

Usage

```
survival(t, ObjFrailty)
```

Arguments

t	time for survival function.
ObjFrailty	an object from the frailtypack fit.

Value

return the value of survival function in t .

Examples

```
## Not run:

#-- a fit Shared
data(readmission)

fit.shared <- frailtyPenal(Surv(time,event)~dukes+cluster(id)+
  strata(sex),n.knots=10,kappa1=10000,kappa2=10000,data=readmission,
  Frailty=TRUE)

#-- calling survival
survival(20,fit.shared)

## End(Not run)
```

terminal

Identify terminal indicator

Description

This is a special function used in the context of recurrent event models with terminal event (e.g., censoring variable related to recurrent events). It contains the status indicator, normally 0=alive, 1=dead, and is used on the right hand side of a formula of a 'frailtyPenal' object. Using `terminal()` in a formula implies that a joint frailty model for recurrent events and terminal events is fitted.

Usage

```
terminal(x)
```

Arguments

x	A numeric variable but should be a Boolean which equals 1 if the subject is dead and 0 if he is alive or censored, as a death indicator.
---	--

Value

x	a death indicator
---	-------------------

See Also

[frailtyPenal](#)

timedep

*Identify time-varying effects***Description**

This is a special function used in the context of Cox models and shared and joint frailty models. It identifies time-varying effects of covariates in the model. It is used in 'frailtyPenal' on the right hand side of formula or of formula.terminalEvent.

When considering time-varying effects in a survival model, regression coefficients can be modeled with a linear combination of B-splines $B(t)$ with coefficients ζ of order q with m interior knots :

$$\beta(t) = \sum_{j=-q+1}^m \zeta_j B_{j,q}(t)$$

You can notice that a linear combination of B-splines of order 1 without any interior knots (0 interior knot) is the same as a model without time-varying effect (or with constant effect over time).

Statistical tests (likelihood ratio tests) can be done in order to know whether the time-dependant coefficients are significantly different from zero or to test whether a covariate has a time-dependant effect significantly different from zero or not. These tests are correct only with a parametric approach yet.

- Proportional Hazard assumption ?

Time-dependency of a covariate effect can be tested. We need to estimate $m + q$ parameters ζ_j for $j = -q + 1, \dots, m$ for a time-varying coefficient. Only one ($q = 1, m = 0$) parameter is estimated for a constant effect. A global test is done.

$$H_0 : \beta(t) = \beta$$

The corresponding LR statistic has a χ^2 distribution of degree $m + q - 1$.

- Significant association ?

We can also use a LR test to test whether a covariate has a significant effect on the hazard function. The null hypothesis is :

$$H_0 : \beta(t) = 0$$

For that we fit a model considering the covariate with a regression coefficient modeled using B-splines and a model without the covariate. Hence, the LR statistic has a χ^2 distribution of degree $m + q$.

Usage

timedep(x)

Arguments

x A numerical or a factor variable that would have a time-varying effect on the event

Value

x A variable identified with a time-varying effect

References

Y. Mazroui, A. Mauguen, S. Mathoulin-Pelissier, G. MacGrogan, V. Brouste, V. Rondeau (2013). Time-varying coefficients in a multivariate frailty model: Application to breast cancer recurrences of several types and death. To appear.

Examples

```
## Not run:

data(readmission)

###--- Shared Frailty model with time-varying effect ---###

sha.time <- frailtyPenal(Surv(time,event)~cluster(id)+dukes+charlson+
timedep(sex)+chemo,Frailty=TRUE,data=readmission,n.knots=8,kappa1=1,
betaknots=3,betaorder=1)

#-- print results of the fit and the associated curves for the
#-- time-dependant effects
print(sha.time)

###--- Joint Frailty model with time-varying effect ---###

joi.time <- frailtyPenal(Surv(time,event)~cluster(id)+timedep(sex)+
chemo+terminal(death),formula.terminalEvent=~timedep(sex)+chemo,
joint=TRUE,Frailty=TRUE,data=readmission,n.knots=8,kappa1=1,kappa2=1,
betaknots=3,betaorder=1)

print(joi.time)

## End(Not run)
```

Index

*Topic **concordance**

Cmeasures, 10

*Topic **datasets**

bcos, 10

dataAdditive, 13

dataNested, 14

readmission, 42

*Topic **file**

additivePenal, 5

plot.frailtyPenal, 28

plot.prediction, 32

*Topic **methods**

plot.additivePenal, 27

plot.jointPenal, 30

plot.nestedPenal, 31

print.additivePenal, 38

print.Cmeasures, 39

print.frailtyPenal, 39

print.jointPenal, 40

print.nestedPenal, 41

print.prediction, 42

summary.additivePenal, 45

summary.frailtyPenal, 46

summary.jointPenal, 48

summary.nestedPenal, 49

*Topic **misc**

num.id, 26

prediction, 33

slope, 43

subcluster, 44

terminal, 52

timedep, 53

*Topic **models**

frailtyPenal, 15

*Topic **package**

frailtypack-package, 3

additivePenal, 5, 28, 38, 44, 46

bcos, 10

CbootstrapFP (Cmeasures), 10

cindexes (Cmeasures), 10

Cmeasures, 10, 39

dataAdditive, 13

dataNested, 14

factor.names (frailtyPenal), 15

frailtypack (frailtypack-package), 3

frailtypack-package, 3

frailtyPenal, 12, 15, 27, 29, 31, 32, 40, 41, 45, 47–49, 52

hazard, 25

lines.additivePenal

(plot.additivePenal), 27

lines.frailtyPenal (plot.frailtyPenal), 28

lines.jointPenal (plot.jointPenal), 30

lines.nestedPenal (plot.nestedPenal), 31

num.id, 23, 26

plot.additivePenal, 27

plot.frailtyPenal, 28, 28, 32

plot.jointPenal, 30

plot.nestedPenal, 31

plot.predFrailty (plot.prediction), 32

plot.prediction, 32

plot.predJoint (plot.prediction), 32

prediction, 33, 42

print.additivePenal, 38

print.Cmeasures, 12, 39

print.frailtyPenal, 39

print.jointPenal, 40

print.nestedPenal, 41

print.predFrailty (print.prediction), 42

print.prediction, 42

print.predJoint (print.prediction), 42

`print.summary.additivePenal`
 (`summary.additivePenal`), 45
`print.summary.frailtyPenal`
 (`summary.frailtyPenal`), 46
`print.summary.jointPenal`
 (`summary.jointPenal`), 48
`print.summary.nestedPenal`
 (`summary.nestedPenal`), 49

`readmission`, 42

`slope`, 9, 43
`statFP (Cmeasures)`, 10
`subcluster`, 23, 44
`summary.additivePenal`, 45
`summary.frailtyPenal`, 46
`summary.jointPenal`, 48
`summary.nestedPenal`, 49
`SurvIC`, 23, 50
`survival`, 51

`terminal`, 23, 52
`timedep`, 23, 53
`timedep.names (frailtyPenal)`, 15

`waldtest (frailtyPenal)`, 15