

Package ‘frailtypack’

October 23, 2009

Version 2.2-9.5

Date 2009-10-22

Title Frailty models using maximum penalized likelihood estimation

Author Juan R González, Virginie Rondeau <Virginie.Rondeau@isped.u-bordeaux2.fr> and Yassin Mazroui <Yassin.Mazroui@isped.u-bordeaux2.fr>

Maintainer Juan R Gonzalez <jrgonzalez@creal.cat>

Depends R (>= 2.0.0), survival

Description Frailtypack now fits several classes of frailty models using Penalized Likelihood on the hazard function. 1) A shared gamma frailty model and Cox proportional hazards model. Left truncated, censored data and strata (max=2) are allowed. Clustered and recurrent survival times can be studied (the Andersen-Gill(1982) approach has been implemented for recurrent events). An automatic choice of the smoothing parameter is possible using an approximated cross-validation procedure. 2) Additive frailty models for proportional hazards 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 of recurrent events with terminal event.

License GPL (>= 2)

URL <http://www.r-project.org>

Encoding latin1

Repository CRAN

Date/Publication 2009-10-23 06:40:55

R topics documented:

additivePenal	2
dataAdditive	6
dataJoint	7

dataNested	8
frailtyPenal for Joint frailty models	9
frailtyPenal for Nested frailty models	13
frailtyPenal for Shared frailty models	17
plot.additivePenal	21
plot.frailtyPenal	22
plot.jointPenal	23
plot.nestedPenal	25
print.additivePenal	26
print.frailtyPenal	27
print.jointPenal	28
print.nestedPenal	30
readmission	31
slope	32
subcluster	33
summary.additivePenal	34
summary.frailtyPenal	35
summary.jointPenal	36
summary.nestedPenal	37
terminal	38

Index **40**

additivePenal	<i>Fit an Additive Frailty model using penalized likelihood estimation</i>
---------------	--

Description

Fit an additive frailty model using penalized likelihood 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 hazards models 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 gamma 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:

$$\lambda_{ij}(t|u_i, v_i) = \lambda_0(t) \exp(u_i + v_i X_{ij1} + \sum_{k=1}^p \beta_k X_{ijk})$$

$$u_i \sim \mathcal{N}(0, \sigma^2) \quad v_i \sim \mathcal{N}(0, \tau^2) \quad \text{cov}(u_i, v_i) = \rho\sigma\tau$$

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)
```

Arguments

<code>formula</code>	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.
<code>data</code>	a 'data.frame' in which to interpret the variables named in the 'formula'.
<code>correlation</code>	Logical value. Are the random effects correlated? If so, the correlation coefficient is estimated. The default is FALSE.
<code>recurrentAG</code>	Always FALSE for additive models (left-truncated data are not allowed).
<code>cross.validation</code>	Logical value. Is cross validation procedure used for estimating smoothing parameter? 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.
<code>n.knots</code>	integer giving the number of knots to use. Value required. 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)
<code>kappa1</code>	positive smoothing parameter. The coefficient κ of the integral of the squared second derivative of hazard function in the fit (penalized log likelihood). 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)
<code>kappa2</code>	Positive smoothing parameter for the second stratum when data are stratified. See <code>kappa1</code> .
<code>maxit</code>	maximum number of iterations for the Marquardt algorithm. Default is 350

Details

The estimated parameter are obtained by maximizing the penalized log-likelihood using the robust Marquardt algorithm (Marquardt,1963). The parameters are initialized with values obtained with Cox proportional hazard models. 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 ($< 10^{-6}$). 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

When an additive model is fitted, at first, a Cox proportional hazards model (without random effects) is fitted with covariates, it provides initial values for the splines coefficients and the regression coefficients. Then, the variance of the random intercept σ^2 is set at 0.25, the variance of the random slope τ^2 is set at 0.025 and the covariance of these two random terms ($cov(u_i, v_i)$) is set at $3.75 * 10^{-3}$. The last step is then the maximisation of the penalized log-likelihood on all parameters.

PARAMETERS LIMIT VALUES

As frailtypack is written in Fortran 77 some parameters had to be hard coded in. The default values of these parameters are, with the corresponding variable name in the fortran code between brackets.

maximum number of observations (ndatemax): 30000
 maximum number of groups (ngmax): 1000
 maximum number of subjects (nsujetmax): 15000
 maximum number of parameters (npmax) :50
 maximum number of covariates (nvarmax):50
 maximum number of subgroups (nssgmax):5000

If these parameters are not large enough (an error message will let you know this), you need to reset them in additive.f and recompile.

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 coefficients of the linear predictor, which multiply the columns of the model matrix.
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?
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 corresponding to each baseline function as components
lam	matrix of hazard estimates at x1 times and confidence bands.
lam2	the same value as lam for the second stratum
loglikPenal	the complete marginal penalized log-likelihood
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 function
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 function)
surv	matrix of baseline survival estimates at x1 times 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)
type	character string specifying the type of censoring. Possible values are "right", "left", "counting", "interval", "interval2". The default is "right" or "counting" depending on whether the 'time2' argument is absent (not interval-censored data) or present (interval-censored data), respectively.
varH	the variance matrix of all parameters (Sigma2, Tau2, the regression coefficients and the spline coefficients).
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 function for the recurrent events are estimated. By default seq(0,max(time),length=99), where time is the vector of survival times.
x2	see x1 value for the second stratum

Note

"Kappa" (kappal 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, 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.

See Also

[print.additivePenal](#), [plot.additivePenal](#), [summary.additivePenal](#), [cluster](#), [slope](#), [strata](#)

Examples

```

### Additive model with 1 covariate ###

## Not run:

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

# It takes around 4 minutes to converge. Var1 is boolean as a treatment variable. #

print(modAdd)
summary(modAdd)
plot(modAdd)

## 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 a censoring indicator)
- var1** 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 Medicine*, **27**, 1894-1910.

See Also[additivePenal](#)

`dataJoint`*Simulated data with recurrent events and informative terminal event*

Description

This contains simulated data with recurrent events and time until last follow-up that is related to the recurrent process. This data set is useful to illustrate how to fit a joint model where the censoring mechanism is informative about the recurring process. This database combines different observations for each subject (sample size=500, number of lines=1053 so in average more than two recurrent events per subject). The right-censored case is treated only and the calendar timescale representation is used with delayed entry. Exponential death times and recurring times are generated using a gamma frailty model. The common frailty term is iid $\Gamma(2, 2)$ and α , the coefficient associated with the frailty term in the death time hazard function, is set at 0. The right-censoring variable is set at 0.8 and the two covariates are generated from a Bernoulli distribution ($p=0.5$).

Usage`data(dataJoint)`**Format**

This data frame contains the following columns:

id identification variable (ex: subject number)
time.entry start of interval (0 or previous recurrence time)
time.end end of interval (recurrence or censoring time)
status censoring status for recurrent events (1:event, 0: censored or dead)
status.terminal censoring status for terminal events (0:alive, 1:death)
var1 dichotomous covariate (0,1)
var2 dichotomous covariate (0,1)

Source

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.

See Also[frailtyPenal for Joint frailty models](#)

`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.

See Also

[frailtyPenal](#) for Nested frailty models

frailtyPenal for Joint frailty models

Fit Joint Frailty model for recurrent and terminal events using penalized likelihood estimation

Description

Fit a joint frailty model for recurrent and terminal events using a Penalized Likelihood on the hazard function. Left-truncated and right-censored data are allowed. Stratified analysis is not possible. Joint frailty models allow studying, jointly, survival processes of recurrent and terminal events, by considering the terminal event as an informative censoring. A common frailty term (ω_i) for the two rates 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 death rate). The covariates could be different for the recurrent rate and death rate.

The joint model for the hazard functions for recurrent event $r_i(\cdot)$ and death $\lambda_i(\cdot)$ is :

$$r_i(t|\omega_i) = \omega_i r_0(t) \exp(\beta_1' \mathbf{Z}_i(t))$$

$$\lambda_i(t|\omega_i) = \omega_i^\alpha \lambda_0(t) \exp(\beta_2' \mathbf{Z}_i(t))$$

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. And where $\omega_i \sim \Gamma(\frac{1}{\theta}, \frac{1}{\theta})$ is iid.

Arguments

formula	a formula object, with the response on the left of a \sim operator, and the terms on the right. The response must be a survival object as returned by the 'Surv' function like in survival package.
formula.terminalEvent	a formula object, only requires terms on the right to indicate which variables are modelling the terminal event.
data	a 'data.frame' in which to interpret the variables named in the 'formula' and 'formula.terminalEvent'.
Frailty	Logical value. Is model with frailties fitted? If so, variance of frailty parameter is estimated. If not, Cox proportional hazards model is estimated using Penalized Likelihood on the hazard function. The default is FALSE
joint	Logical value. Is joint model fitted? If so, 'formula.terminalEvent' 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.

<code>cross.validation</code>	Logical value. Is cross validation procedure used for estimating smoothing parameter? 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 cross validation has been implemented for a Cox proportional hazard model, with no covariates. The default is FALSE.
<code>n.knots</code>	integer giving the number of knots to use. Value required. 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)
<code>kappa1</code>	positive smoothing parameter. The coefficient kappa of the integral of the squared second derivative of hazard function in the fit (penalized log likelihood). 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)
<code>kappa2</code>	positive smoothing parameter for the terminal event rate. See <code>kappa1</code> . Stratification is not allowed here.
<code>maxit</code>	maximum number of iterations for the Marquardt algorithm. Default is 350

Details

The estimated parameter are obtained by maximizing the penalized loglikelihood using the robust Marquardt algorithm (Marquardt, 1963) which is a combination between a Newton-Raphson algorithm and a steepest descent algorithm. 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, 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

When a joint frailty model is fitted, at first, the splines coefficients and the regression coefficients are initialized at 0.5, and the variance of the frailty term θ and the coefficient α associated in the death hazard function are initialized at 1. Thus, the next step is the maximisation of the penalized log-likelihood.

PARAMETERS LIMIT VALUES

As frailtypack is written in Fortran 77 some parameters had to be hard coded in. The default values of these parameters are, with the corresponding variable name in the fortran code between brackets.

maximum number of observations(`ndatemax`): 30000
maximum number of groups(`ngmax`): 1000
maximum number of subjects(`nsujetmax`): 15000
maximum number of parameters (`npmax`):50
maximum number of covariates (`nvarmax`):50

If these parameters are not large enough (an error message will let you know this), you need to reset them in joint.f and recompile.

Value

Parameters estimates of a joint frailty model, more generally a 'frailtyPenal' object. Methods defined for 'frailtyPenal' objects are provided for print, plot and summary. The following components are included in a 'frailtyPenal' object for Joint frailty models.

alpha	the coefficient associated with the frailty parameter in the terminal hazard function
call	The code used for fitting the model
coef	the coefficients of the linear predictor, which multiply the columns of the model matrix.
cross.Val	Logical value. Is cross validation procedure used for estimating the smoothing parameters?
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 corresponding to each baseline function as components
lam	matrix of hazard estimates at x1 times and confidence bands.
lam2	the same value as lam for the second stratum
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
nvar	A vector with the number of covariates of each type of hazard function as components
surv	matrix of baseline survival estimates at x1 times and confidence bands.
surv2	the same value as surv for the the second stratum
theta	variance of the frailty parameter ($Var(\omega_i)$)
type	character string specifying the type of censoring. Possible values are "right", "left", "counting", "interval", "interval2". The default is "right" or "counting" depending on whether the 'time2' argument is absent (not interval-censored data) or present (interval-censored data), respectively.
varH	the variance matrix of all parameters (theta, the regression coefficients and the spline coefficients).
varHIH	the robust estimation of the variance matrix of all parameters (theta, the regression coefficients and the spline coefficients).
x1	vector of times where both survival and hazard function for the recurrent events are estimated. By default seq(0,max(time),length=99), where time is the vector of survival times.
x2	see x1 value for the terminal events survival and hazard function

Note

"Kappa" (kappal 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 devide this value by 10 or 5 until it converges.

References

- 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.

See Also

`summary.jointPenal`, `print.jointPenal`, `plot.jointPenal`, `dataJoint`, `terminal`, `cluster`

Examples

```
### Joint model (recurrent and terminal events) with 2 covariates ###
### on a simulated dataset ###

## Not run:

data(dataJoint)

modJoint<-frailtyPenal(Surv(time.entry,time.end,status)~cluster(id)+var1+var2
                      +terminal(status.terminal),
                      formula.terminalEvent=~var1,
                      data=dataJoint,n.knots=7,
                      kappal=1, kappa2=1, Frailty = TRUE, joint=TRUE,
                      recurrentAG=TRUE)

print(modJoint)
summary(modJoint)
plot(modJoint)

## End(Not run)

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

frailtyPenal for Nested frailty models

Fit Nested Frailty model using penalized likelihood estimation

Description

Fit a nested frailty model using a Penalized Likelihood on the hazard function. 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 :

$$\lambda_{ijk}(t|v_i, w_{ij}) = v_i w_{ij} \lambda_0(t) \exp(\beta' \mathbf{X}_{ijk})$$

$$v_i \sim \Gamma\left(\frac{1}{\alpha}, \frac{1}{\alpha}\right) \text{ i.i.d. } \mathbf{E}(v_i) = 1 \quad \mathbf{Var}(v_i) = \alpha \quad w_{ij} \sim \Gamma\left(\frac{1}{\eta}, \frac{1}{\eta}\right) \text{ i.i.d. } \mathbf{E}(w_{ij}) = 1 \quad \mathbf{Var}(w_{ij}) = \eta$$

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

Arguments

formula	a formula object, with the response on the left of a \sim 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>subcluster()</code> function is required.
formula.terminalEvent	Not required.
data	a 'data.frame' in which to interpret the variables named in the 'formula'.
Frailty	Logical value. Is model with frailties fitted? If so, variance of frailty parameter is estimated. If not, Cox proportional hazards model is estimated using Penalized Likelihood on the hazardfunction. The default is FALSE.
joint	Not required
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? If so a search of the smoothing parameter using cross validation is done, with <code>kappal</code> as the seed. The cross validation is not implemented for two strata. 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. 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. The coefficient kappa of the integral of the squared second derivative of hazard function in the fit (penalized log likelihood). 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 for the second stratum, when data are stratified. See kappa1.
maxit	maximum number of iterations for the Marquardt algorithm. Default is 350

Details

The estimated parameter are obtained by maximizing the penalized loglikelihood using the robust Marquardt algorithm (Marquardt, 1963) which is a combination between a Newton-Raphson algorithm and a steepest descent algorithm. 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 smoothing parameter can be fixed a priori or chosen by maximizing a likelihood cross validation criterion. 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 ($< 10^{-6}$). To be sure of having a positive function at all stages of the algorithm, the spline coefficients were reparametrizes 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 integrations in the full log likelihood werre evaluated using Gaussian quadrature. Laguerre polynomials with 10 points were used to treat the integrations on $[0, \infty[$.

INITIAL VALUES

When a nested frailty model is fitted, at first, a Cox proportional hazards model with covariates (without random effects) is fitted and it provides initial values for the regression coefficients. Then, a model with covariates and only subgroup frailties is fitted, so it gives new initial values for the regression coefficients and for the subgroup frailty terms variance. Then, a model with covariates and only group frailty terms is fitted, so it gives new initial values for the regression coefficients and for the group frailties variance. The last step is the maximisation of the penalized log-likelihood for all parameters.

PARAMETERS LIMIT VALUES

As frailtypack is written in Fortran 77 some parameters had to be hard coded in. The default values of these parameters are, with the corresponding variable name in the fortran code between brackets.

maximum number of observations (ndatemax): 30000
 maximum number of groups (ngmax): 1000
 maximum number of subjects (nsujetmax): 15000
 maximum number of parameters (npmax) :50
 maximum number of covariates (nvarmax):50
 maximum number of subgroups (nssgmax):5000

If these parameters are not large enough (an error message will let you know this), you need to reset them in `nested.f` and recompile.

Value

a Nested frailty model or more generally an object of class 'frailtyPenal'. Methods defined for 'frailtyPenal' objects are provided for print, plot and summary. The following components are included in a 'frailtyPenal' object for nested frailty models.

<code>alpha</code>	variance of the cluster effect ($\mathbf{Var}(v_i)$)
<code>call</code>	The code used for fitting the model
<code>coef</code>	the coefficients of the linear predictor, which multiply the columns of the model matrix.
<code>cross.Val</code>	Logical value. Is cross validation procedure used for estimating the smoothing parameters?
<code>DoF</code>	degrees of freedom
<code>eta</code>	variance of the subcluster effect ($\mathbf{Var}(w_{ij})$)
<code>formula</code>	the formula part of the code used for the model
<code>groups</code>	the maximum number of groups used in the fit
<code>subgroups</code>	the maximum number of subgroups used in the fit
<code>kappa</code>	A vector with the smoothing parameters corresponding to each baseline function as components
<code>lam</code>	matrix of hazard estimates at x1 times and confidence bands.
<code>lam2</code>	the same value as lam for the second stratum
<code>n</code>	the number of observations used in the fit.
<code>n.events</code>	the number of events observed in the fit
<code>n.iter</code>	number of iterations needed to converge
<code>n.knots</code>	number of knots
<code>n.strat</code>	A vector with the number of covariates of each type of hazard function as components
<code>nvar</code>	A vector with the number of covariates of each type of hazard function as components
<code>surv</code>	matrix of baseline survival estimates at x1 times and confidence bands.
<code>surv2</code>	the same value as surv for the second stratum
<code>type</code>	a character string specifying the type of censoring. Possible values are "right", "left", "counting", "interval", "interval2". The default is "right" or "counting" depending on whether the 'time2' argument is absent (not interval-censored data) or present (interval-censored data), respectively.
<code>varH</code>	the variance matrix of all parameters (alpha, eta, the regression coefficients and the spline coefficients).
<code>varHIH</code>	the robust estimation of the variance matrix of all parameters (alpha, eta, the regression coefficients and the spline coefficients).
<code>x1</code>	vector of times where both survival and hazard function are estimated. By default <code>seq(0,max(time),length=99)</code> , where time is the vector of survival times.
<code>x2</code>	the same value as x1 for the second stratum

Note

"Kappa" (kappal 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 devide this value by 10 or 5 until it converges.

References

- 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.
- D. Marquardt (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, 431-441.

See Also

[print.nestedPenal](#), [summary.nestedPenal](#), [plot.nestedPenal](#), [cluster](#), [subcluster](#), [strata](#)

Examples

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

## Not run:

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

# It takes around 24 minutes to converge (depends on the processor)#

print(modClu)
summary(modClu)
plot(modClu)

modClu.str<-frailtyPenal(Surv(t1,t2,event)~cluster(group)+subcluster(subgroup)+
                        cov1+strata(cov2)
                        ,Frailty=TRUE,data=dataNested,n.knots=8,kappal=20000,kappa2=20000)

# It takes around 8 minutes to converge (depends on the processor)#

print(modClu.str)
summary(modClu.str)
plot(modClu.str)
```

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

```
frailtyPenal for Shared frailty models
Fit Shared Gamma Frailty model using penalized likelihood estimation
```

Description

Fit a shared gamma frailty model using a Penalized Likelihood on the hazard function. Left truncated and 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 z_i , of a shared frailty model for the j^{th} subject in the i^{th} group:

$$\lambda_{ij}(t|z_i) = \lambda_0(t)z_i \exp(\beta' \mathbf{X}_{ij})$$

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

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

Usage

```
frailtyPenal(formula, formula.terminalEvent, data, Frailty = FALSE,
             joint = FALSE, recurrentAG = FALSE, cross.validation =
             FALSE, n.knots, kappal, kappa2, maxit = 350)
```

Arguments

formula	a formula object, with the response on the left of a \sim operator, and the terms on the right. The response must be a survival object as returned by the 'Surv' function like in survival package.
formula.terminalEvent	Not required.
data	a 'data.frame' in which to interpret the variables named in the 'formula'.
Frailty	Logical value. Is model with frailties fitted? If so, variance of frailty parameter is estimated. If not, Cox proportional hazards model is estimated using Penalized Likelihood on the hazard function. The default is TRUE
joint	Not required.

<code>recurrentAG</code>	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.
<code>cross.validation</code>	Logical value. Is cross validation procedure used for estimating smoothing parameter? 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 cross validation has been implemented for a Cox proportional hazard model, with no covariates. The default is FALSE.
<code>n.knots</code>	integer giving the number of knots to use. Value required. 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)
<code>kappa1</code>	positive smoothing parameter. The coefficient kappa of the integral of the squared second derivative of hazard function in the fit (penalized log likelihood). 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)
<code>kappa2</code>	positive smoothing parameter for the second stratum, when data are stratified. See <code>kappa1</code> .
<code>maxit</code>	maximum number of iterations for the Marquardt algorithm. Default is 350

Details

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. 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.

INITIAL VALUES

When a shared frailty model is fitted, at first, the program fit a model with frailty or covariates, it gives initial values for the splines coefficient. Then, the initial value 0.1 is given to the variance of the frailty term (θ) and to the regression coefficients. Thus, the next step after the parameter initialisation is the maximisation of the penalized log-likelihood.

PARAMETERS LIMIT VALUES

As `frailtypack` is written in Fortran 77 some parameters had to be hard coded in. The default values of these parameters are, with the corresponding variable name in the fortran code between brackets.

maximum number of observations (`ndatemax`): 30000
 maximum number of groups (`ngmax`): 5000
 maximum number of subjects (`nsujetmax`): 15000
 maximum number of parameters (`npmax`): 50
 maximum number of covariates (`nvarmax`): 50

If these parameters are not large enough (an error message will let you know this), you need to reset them in `frailtypack.f` and recompile. In particular, the statements defining these parameters are

Value

The following components are included in a 'frailtyPenal' object for shared frailty models.

call	The code used for the model
coef	the coefficients of the linear predictor, which multiply the columns of the model matrix.
cross.Val	Logical value. Is cross validation procedure used for estimating the smoothing parameters?
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 corresponding to each baseline function as components
lam	matrix of hazard estimates at x1 times and confidence bands.
lam2	the same value as lam for the second stratum
loglikPenal	the complete marginal penalized log-likelihood
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 function
n.strat	number of stratum
surv	matrix of baseline survival estimates at x1 times and confidence bands.
surv2	the same value as surv for the second stratum
theta	variance of frailty parameter
type	charater string specifying the type of censoring. Possible values are "right", "left", "counting", "interval", "interval2". The default is "right" or "counting" depending on whether the 'time2' argument is absent or present, respectively.
varH	the variance matrix of all parameters (theta, the regression coefficients and the spline coefficients).
varHIH	the robust estimation of the variance matrix of all parameters (theta, the regression coefficients and the spline coefficients).
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	see x1 value for the second stratum

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 devide this value by 10 or 5 until it converges.

References

- 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

[print.frailtyPenal](#), [summary.frailtyPenal](#), [plot.frailtyPenal](#), [cluster](#), [strata](#)

Examples

```
### Shared model ###
data(kidney)
frailtyPenal(Surv(time,status)~ cluster(id)+sex+age,
             n.knots=12,kappal=10000,data=kidney,Frailty=TRUE)

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

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

### Shared model with truncated data ###

# Here is created a hypothetical truncated data
kidney$tt0<-rep(0,nrow(kidney))
kidney$tt0[1:3]<-c(2,9,13)

# then, we fit the model
frailtyPenal(Surv(tt0,time,status)~sex+age+cluster(id),
             n.knots=12,kappal=10000,data=kidney,Frailty=TRUE)

### Shared model - stratified analysis ###

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

### Shared model - recurrentAG=TRUE ###

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,kappal=100000,recurrentAG=TRUE)

### Shared model - cross.validation=TRUE ###
```

```
## Not run:

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)
## End(Not run)
```

plot.additivePenal *Plot Method for an Additive frailty model.*

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, type.plot = "hazard", conf.bands=TRUE, ...)
```

Arguments

x	An fitted additive frailty model (output from calling additivePenal 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.
...	Other graphical parameters

Value

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

See Also

[print.additivePenal](#), [summary.additivePenal](#), [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)

# It takes around 4 minutes to converge. '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, ...)
```

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.
...	Other graphical parameters

Value

Print a plot of a shared frailty model.

See Also

`print.frailtyPenal`, `summary.frailtyPenal`, `frailtyPenal` for Shared frailty models

Examples

```
data(readmission)

# /** Shared frailty model **/

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

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

# /** Cox proportional hazard model using Penalized likelihood **/

modCox<-frailtyPenal(Surv(time,event)~as.factor(dukes)+cluster(id),
                    n.knots=10,kappal=10000,data=readmission,Frailty=FALSE)

plot(modCox)

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

`plot.jointPenal` *Plot Method for a Joint frailty model.*

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, ...)
```

Arguments

`x` A joint model, or more generally an object of class `frailtyPenal` for Joint frailty model (output from calling `frailtyPenal` function).

<code>event</code>	a character string specifying the type of curve. Possible value are "terminal", "recurrent", or "both". The default is "both".
<code>type.plot</code>	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"
<code>conf.bands</code>	logical value. Determines whether confidence bands will be plotted. The default is to do so.
<code>...</code>	Other graphical parameters

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

[print.jointPenal](#), [summary.jointPenal](#), [frailtyPenal](#) for Joint frailty models

Examples

```
## Not run:

data(dataJoint)

modJoint<-frailtyPenal(Surv(time.entry,time.end,status)~cluster(id)+var1+var2
                      +terminal(status.terminal),
                      formula.terminalEvent=~var1,
                      data=dataJoint,n.knots=7,
                      kappal=1, kappa2=1,Frailty=TRUE, joint=TRUE, recurrentAG=TRUE)

# It takes around 1 minute to converge. #

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

Arguments

x	A nested model, more generally an object of class frailtyPenal for Nested frailty models (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.
...	Other graphical parameters

Value

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

See Also

[print.nestedPenal](#), [summary.nestedPenal](#), [frailtyPenal](#) for Nested frailty models

Examples

```
## Not run:  
  
data(dataNested)  
modNested<-frailtyPenal(Surv(t1,t2,event)~cluster(group)+subcluster(subgroup)+  
cov1+cov2,data=dataNested,n.knots=8,kappa1=50000)  
  
# It takes around 90 minutes to converge (depends on the processor) #
```

```
plot(modNested, conf.bands=FALSE)

## End(Not run)
```

```
print.additivePenal
    Print a Short Summary of parameter estimates of an additive frailty
    model
```

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, 3), ...)
```

Arguments

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

Value

<code>n</code>	the number of observations used in the fit.
<code>n.groups</code>	the maximum number of groups used in the fit
<code>n.events</code>	the number of events observed in the fit
<code>coef</code>	the coefficients of the linear predictor, which multiply the columns of the model matrix.
<code>SE(H)</code>	the standard error deduced from the variance matrix of theta and of the coefficients.
<code>SE(HIH)</code>	the standard error deduced from the robust estimation of the variance matrix of theta and of the coefficients.
<code>z</code>	quantile of the χ^2 mixture distribution
<code>p</code>	p-value
<code>Variance for</code>	the random intercept
	Variance for the random effect associated to the baseline risk function
<code>Variance for</code>	the random slope
	Variance for the random effect associated to the treatment effects across trials

See Also

[summary.additivePenal](#), [additivePenal](#), [plot.additivePenal](#)

Examples

```
# /*** Additive frailty model with 1 covariate ***/
## Not run:
data(dataAdditive)
modAdd<-additivePenal(Surv(t1,t2,event)~cluster(group)+var1+slope(var1),
                      correlation=TRUE,data=dataAdditive,n.knots=8,kappa1=862)
# It takes around 4 minutes to converge. 'var1' is boolean as a treatment variable. #
print(modAdd)
## End(Not run)
```

`print.frailtyPenal` *Print a Short Summary of parameter estimates of a shared gamma frailty model*

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, 3), ...)
```

Arguments

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

Value

<code>n</code>	the number of observations used in the fit.
<code>n.groups</code>	the maximum number of groups used in the fit
<code>n.events</code>	the number of events observed in the fit
<code>theta</code>	variance of frailty parameter

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

`summary.frailtyPenal`, `frailtyPenal` for Shared frailty models, `plot.frailtyPenal`

Examples

```
data(readmission)

# /** Shared frailty model **/

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

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

# /** Cox proportional hazard model using Penalized likelihood **/

modCox<-frailtyPenal(Surv(time,event)~as.factor(dukes)+cluster(id),
                    n.knots=10,kappal=10000,data=readmission,Frailty=FALSE)

print(modCox)
```

```
print.jointPenal Print a Short Summary of parameter estimates of a joint frailty model
```

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, 3), ...)
```

Arguments

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

Value

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

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
theta	variance of the common frailty term
alpha	the coefficient associated with the frailty parameter terminal event hazard function
coef	the coefficients of the linear predictor, which multiply the columns of the model matrix.
varH	the variance matrix of theta and of the coefficients.
varHIH	the robust estimation of the variance matrix of theta and of the coefficients.
SE(H)	the standard error of the estimaes deduced from the variance matrix of theta and of the coefficients.
SE(HIH)	the standard error of the estimaes deduced from the robust estimation of the variance matrix of theta and of the coefficients.
p	p-value

See Also

[summary.jointPenal](#), [frailtyPenal](#) for Joint frailty models, [plot.jointPenal](#)

Examples

```
# /*** Joint frailty model ***/

## Not run:

data(dataJoint)

modJoint<-frailtyPenal(Surv(time.entry,time.end,status)~cluster(id)+var1+var2
  +terminal(status.terminal),
  formula.terminalEvent=~var1,
  data=dataJoint,n.knots=7, Frailty=TRUE,
  kappal=1, kappa2=1, joint=TRUE, recurrentAG=TRUE)
```

```
# It takes around 1 minute to converge.#
print(modJoint)

## End(Not run)
```

```
print.nestedPenal Print a Short Summary of parameter estimates of a nested frailty
model
```

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, 3), ...)
```

Arguments

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

[summary.nestedPenal](#), [frailtyPenal](#) for Nested frailty models, [plot.nestedPenal](#),

Examples

```

# /*** Nested frailty model ***/

## 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)

# It takes around 24 minutes to converge (depends on the processor) #

print(modNested)

## End(Not run)

```

readmission

Rehospitalization colorectal cancer

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 identifier 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 interoccurrence or censoring time

event censoring 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

Source

González, JR., Fernandez, E., Moreno, V. et al. Gender differences in hospital readmission among colorectal cancer patients. *Journal of Epidemiology and Community Health*. In press, 2005.

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](#), [print.additivePenal](#), [plot.additivePenal](#), [summary.additivePenal](#)

Examples

```
## Not run:

library(frailtypack)
data(dataAdditive)

# Additive with one covariate #

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

# 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)

# 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)

## End(Not run)

```

subcluster	<i>Identify subclusters</i>
------------	-----------------------------

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 a formula of a 'frailtyPenal' object 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 for Nested frailty models](#), [print.nestedPenal](#), [plot.nestedPenal](#), [summary.nestedPenal](#)

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)

print(modClu)

# It takes around 24 minutes to converge (depends on the processor) #
## 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](#), [print.additivePenal](#), [plot.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)

# It takes around 4 minutes to converge. '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 for Shared frailty models](#), [plot.frailtyPenal](#), [summary.frailtyPenal](#)

Examples

```

data(kidney)

# /*** Shared frailty model ***/

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

# /*** Cox proportional hazard model using Penalized likelihood ***/

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

# /*** confidence interval at 95

summary(modSha)
summary(modCox)

# /*** confidence interval at 99

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

```

summary.jointPenal *summary of parameter estimates of a joint frailty model*

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` for Joint frailty models, `plot.jointPenal`, `print.jointPenal`,

Examples

```
## Not run:

data(dataJoint)

modJoint<-frailtyPenal(Surv(time.entry,time.end,status)~cluster(id)+var1+var2
  +terminal(status.terminal),
  formula.terminalEvent=~var1,
  data=dataJoint,n.knots=7,Frailty=TRUE,
  kappal=1, kappa2=1, joint=TRUE, recurrentAG=TRUE)

# It takes around 1 minute to converge

summary(modJoint)

## End(Not run)
```

```
summary.nestedPenal
```

summary of regression coefficient estimates of a nested frailty model

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

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

Value

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

See Also

[frailtyPenal](#) for Nested frailty models, [plot.nestedPenal](#), [print.nestedPenal](#)

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)

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

summary(modNested)

## 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](#) for Joint frailty models, [plot.jointPenal](#), [print.jointPenal](#), [summary.jointPenal](#)

Examples

```
## Not run:

data(dataJoint)

modJoint<-frailtyPenal(Surv(time.entry,time.end,status)~cluster(id)+var1+var2
                      +terminal(status.terminal),
                      formula.terminalEvent=~var1,
                      data=dataJoint,n.knots=7,Frailty=TRUE,
                      kappal=1, kappa2=1, joint=TRUE, recurrentAG=TRUE)

print(modJoint)
## End(Not run)
```

Index

*Topic **datasets**

dataAdditive, 6
dataJoint, 7
dataNested, 8
readmission, 31

*Topic **file**

additivePenal, 2
plot.frailtyPenal, 22

*Topic **methods**

plot.additivePenal, 21
plot.jointPenal, 23
plot.nestedPenal, 25
print.additivePenal, 26
print.frailtyPenal, 27
print.jointPenal, 28
print.nestedPenal, 30
summary.additivePenal, 34
summary.frailtyPenal, 35
summary.jointPenal, 36
summary.nestedPenal, 37

*Topic **misc**

slope, 32
subcluster, 33
terminal, 38

*Topic **models**

frailtyPenal for Joint
frailty models, 9
frailtyPenal for Nested
frailty models, 13
frailtyPenal for Shared
frailty models, 17

additivePenal, 2, 6, 21, 27, 32, 34

cluster, 5, 12, 16, 20

dataAdditive, 6
dataJoint, 7, 12
dataNested, 8

frailtyPenal (*frailtyPenal for
Shared frailty models*), 17
frailtyPenal for Joint frailty
models, 7, 9, 24, 29, 37, 39
frailtyPenal for Joint models
(*frailtyPenal for Joint
frailty models*), 9
frailtyPenal for Nested frailty
models, 8, 13, 25, 30, 33, 38
frailtyPenal for Shared frailty
models, 17, 23, 28, 35

lines.additivePenal
(*plot.additivePenal*), 21

lines.frailtyPenal
(*plot.frailtyPenal*), 22

lines.jointPenal
(*plot.jointPenal*), 23

lines.nestedPenal
(*plot.nestedPenal*), 25

plot.additivePenal, 5, 21, 27, 32, 34
plot.frailtyPenal, 20, 22, 28, 35
plot.jointPenal, 12, 23, 29, 37, 39
plot.nestedPenal, 16, 25, 30, 33, 38
print.additivePenal, 5, 21, 26, 32, 34
print.frailtyPenal, 20, 23, 27
print.jointPenal, 12, 24, 28, 37, 39
print.nestedPenal, 16, 25, 30, 33, 38
print.summary.additivePenal
(*summary.additivePenal*), 34
print.summary.frailtyPenal
(*summary.frailtyPenal*), 35
print.summary.jointPenal
(*summary.jointPenal*), 36
print.summary.nestedPenal
(*summary.nestedPenal*), 37

readmission, 31

slope, 5, 32

strata, *5, 16, 20*
subcluster, *16, 33*
summary.additivePenal, *5, 21, 27, 32,*
34
summary.frailtyPenal, *20, 23, 28, 35,*
35
summary.jointPenal, *12, 24, 29, 36, 39*
summary.nestedPenal, *16, 25, 30, 33, 37*
terminal, *12, 38*