Package 'JM'

August 20, 2009

Title Joint Modelling of Longitudinal and Survival Data

Version 0.4-0

Date 2009-08-20

Author Dimitris Rizopoulos <d.rizopoulos@erasmusmc.nl>

Maintainer Dimitris Rizopoulos <d.rizopoulos@erasmusmc.nl>

Description Shared parameter models for the joint modelling of longitudinal and time-to-event data.

Depends R (>= 2.7.0), MASS, nlme, splines, survival

LazyLoad yes

LazyData yes

License GPL (>= 2)

URL http://wiki.r-project.org/rwiki/doku.php?id=packages:cran:jm

Repository CRAN

Date/Publication 2009-08-20 13:47:54

R topics documented:

`aids` *Didanosine versus Zalcitabine in HIV Patients*

Description

A randomized clinical trial in which both longitudinal and survival data were collected to compare the efficacy and safety of two antiretroviral drugs in treating patients who had failed or were intolerant of zidovudine (AZT) therapy.

Format

A data frame with 1408 observations on the following 9 variables.

`patient` patients identifier; in total there are 467 patients.

`Time` the time to death or censoring.

`death` a numeric vector with 0 denoting censoring and 1 death.

`CD4` the CD4 cells count.

`obstime` the time points at which the CD4 cells count was recorded.

`drug` a factor with levels `ddC` denoting zalcitabine and `ddI` denoting didanosine.

`gender` a factor with levels `female` and `male`.

`prevOI` a factor with levels `AIDS` denoting previous opportunistic infection (AIDS diagnosis) at study entry, and `noAIDS` denoting no previous infection.

`AZT` a factor with levels `intolerance` and `failure` denoting AZT intolerance and AZT failure, respectively.

Note

The data frame `aids.id` contains the first CD4 cell count measurement for each patient. This data frame is used to fit the survival model.

Source

http://www.biostat.umn.edu/~brad/data.html, http://www.biostat.umn.edu/~brad/software.html

References

Goldman, A., Carlin, B., Crane, L., Launer, C., Korvick, J., Deyton, L. and Abrams, D. (1996) Response of CD4+ and clinical consequences to treatment using ddI or ddC in patients with advanced HIV infection. *Journal of Acquired Immune Deficiency Syndromes and Human Retrovirology* **11**, 161–169.

Guo, X. and Carlin, B. (2004) Separate and joint modeling of longitudinal and event time data using standard computer packages. *The American Statistician* **58**, 16–24.

Examples

```
summary(aids.id)
```

---

anova *Anova Method for Fitted Joint Models*

Description

Performs a likelihood ratio test between two nested joint models.

Usage

```
## S3 method for class 'jointModel':
anova(object, object2, test = TRUE, ...)
```

Arguments

`object`	an object inheriting from class `jointModel`, nested in `object2`.
`object2`	an object inheriting from class `jointModel`.
`test`	logical; if `TRUE` the likelihood ratio test is performed.
`...`	additional arguments; currently none is used.

Value

An object of class `aov.jointModel` with components,

`nam0`	the name of `object`.
`L0`	the log-likelihood under the null hypothesis (`object`).
`aic0`	the AIC value for the model given by `object`.
`bic0`	the BIC value for the model given by `object`.
`nam1`	the name of `object2`.
`L1`	the log-likelihood under the alternative hypothesis (`object2`).
`aic1`	the AIC value for the model given by `object2`.
`bic1`	the BIC value for the model given by `object2`.
`df`	the degrees of freedom for the test (i.e., the difference in the number of parameters).
`LRT`	the value of the Likelihood Ratio Test statistic (returned if `test = TRUE`).
`p.value`	the *p*-value of the test (returned if `test = TRUE`).

Warning

The code minimally checks whether the models are nested! The user is responsible to supply nested models in order the LRT to be valid.

Author(s)

Dimitris Rizopoulos ⟨d.rizopoulos@erasmusmc.nl⟩

See Also

jointModel

Examples

```
# linear mixed model fit without treatment effect
fitLME.null <- lme(sqrt(CD4) ~ obstime,
    random = ~ 1 | patient, data = aids)
# cox model fit without treatment effect
fitCOX.null <- coxph(Surv(Time, death) ~ 1,
    data = aids.id, x = TRUE)
# joint model fit, under the additive log cumulative hazard model
# without treatment effect
fitJOINT.null <- jointModel(fitLME.null, fitCOX.null,
    timeVar = "obstime", method = "ch-GH")

# linear mixed model fit with treatment effect
fitLME.alt <- lme(sqrt(CD4) ~ obstime * drug - drug,
    random = ~ 1 | patient, data = aids)
# cox model fit with treatment effect
fitCOX.alt <- coxph(Surv(Time, death) ~ drug,
    data = aids.id, x = TRUE)
# joint model fit, under the additive log cumulative hazard model
# with treatment effect
fitJOINT.alt <- jointModel(fitLME.alt, fitCOX.alt, timeVar = "obstime",
    method = "ch-GH")

# likelihood ratio test for treatment effect
anova(fitJOINT.null, fitJOINT.alt)
```

---

coef *Estimated Coefficients for Joint Models*

Description

Extracts estimated coefficients from fitted joint models.

Usage

```
## S3 method for class 'jointModel':
coef(object, process = c("Longitudinal", "Event"),
    include.splineCoefs = FALSE, ...)
## S3 method for class 'jointModel':
fixef(object, process = c("Longitudinal", "Event"),
    include.splineCoefs = FALSE, ...)
```

Arguments

`object`	an object inheriting from class `jointModel`.
`process`	for which model (i.e., linear mixed model or survival model) to extract the estimated coefficients.
`include.splineCoefs`	logical; if `TRUE` and the method argument in `jointModel()` is `"ph-GH"` or `"ch-Laplace"`, the estimated B-spline coefficients are included as well.
`...`	additional arguments; currently none is used.

Details

When `process = "Event"` both methods return the same output. However, for `process = "Longitudinal"`, the `coef()` method returns the subject-specific coefficients, whereas `fixef()` only the fixed effects.

Value

A numeric vector or a matrix of the estimated parameters for the fitted model.

Author(s)

Dimitris Rizopoulos ⟨d.rizopoulos@erasmusmc.nl⟩

See Also

`ranef.jointModel`

Examples

```
# linear mixed model fit
fitLME <- lme(sqrt(CD4) ~ obstime * drug - drug,
    random = ~ 1 | patient, data = aids)
# cox model fit
fitCOX <- coxph(Surv(Time, death) ~ drug, data = aids.id, x = TRUE)

# joint model fit, under the additive log cumulative hazard model
fitJOINT <- jointModel(fitLME, fitCOX,
    timeVar = "obstime", method = "ch-GH")

# fixed effects for the longitudinal process
fixef(fitJOINT)
```

```
# fixed effects + random effects estimates for the longitudinal
# process
coef(fitJOINT)

# fixed effects for the event process
fixef(fitJOINT, process = "Event")
coef(fitJOINT, process = "Event")

# fixed effects for the event process,
# include spline coefficients
fixef(fitJOINT, process = "Event", include.splineCoefs = TRUE)
```

`fitted` *Fitted Values for Joint Models*

Description

Calculates fitted values for joint models.

Usage

```
## S3 method for class 'jointModel':
fitted(object, process = c("Longitudinal", "Event"),
type = c("Marginal", "Subject", "EventTime"), scale = c("survival",
"cumulative-Hazard", "log-cumulative-Hazard"), M = 200, ...)
```

Arguments

`object`	an object inheriting from class `jointModel`.
`process`	for which model (i.e., linear mixed model or survival model) to calculate the fitted values.
`type`	what type of fitted values to calculate for the survival outcome. See **Details**.
`scale`	in which scale to calculate; relevant only when `process = "Event"`.
`M`	how many times to simulate random effects; see **Details** for more info.
`...`	additional arguments; currently none is used.

Details

For `process = "Longitudinal"`, let X denote the design matrix for the fixed effects β, and Z the design matrix for the random effects b. Then for `type = "Marginal"` the fitted values are $X\hat{\beta}$, whereas for `type = "Subject"` they are $X\hat{\beta} + Z\hat{b}$. For `type = "EventTime"` is the same as `type = "Subject"` but evaluated at the observed event times.

For `process = "Event"` and `type = "Subject"` the linear predictor conditional on the random effects estimates is calculated for each sample unit. Depending on the value of the `scale` argument the fitted survival function, cumulative hazard function or log cumulative hazard function

is returned. For `type = "Marginal"`, random effects values for each sample unit are simulated `M` times from a normal distribution with zero mean and covariance matrix the estimated covariance matrix for the random effects. The marginal survival function for the ith sample unit is approximated by

$$S_i(t) = \int S_i(t|b_i)p(b_i)db_i \approx \sum_{m=1}^{M} S_i(t|b_{im}),$$

where $p(b_i)$ denotes the normal probability density function, and b_{im} the mth simulated value for the random effect of the ith sample unit. The cumulative hazard and log cumulative hazard functions are calculated as $H_i(t) = -\log S_i(t)$ and $\log H_i(t) = \log\{-\log S_i(t)\}$, respectively.

Value

a numeric vector of fitted values.

Author(s)

Dimitris Rizopoulos ⟨d.rizopoulos@erasmusmc.nl⟩

See Also

`residuals.jointModel`

Examples

```
# linear mixed model fit
fitLME <- lme(log(serBilir) ~ drug * year,
    random = ~ 1 | id, data = pbc2)
# survival regression fit
fitSURV <- survreg(Surv(years, status2) ~ drug,
    data = pbc2.id, x = TRUE)
# joint model fit, under the (default) Weibull model
fitJOINT <- jointModel(fitLME, fitSURV, timeVar = "year")

# fitted for the longitudinal process
head(cbind(
    "Marg" = fitted(fitJOINT),
    "Subj" = fitted(fitJOINT, type = "Subject")
))

# fitted for the event process - survival function
head(cbind(
    "Marg" = fitted(fitJOINT, process = "Ev"),
    "Subj" = fitted(fitJOINT, process = "Ev", type = "Subject")
))

# fitted for the event process - cumulative hazard function
head(cbind(
    "Marg" = fitted(fitJOINT, process = "Ev",
        scale = "cumulative-Hazard"),
    "Subj" = fitted(fitJOINT, process = "Ev", type = "Subject",
        scale = "cumulative-Hazard")
```

```
))
```

JM *Joint Modelling of Longitudinal and Time-to-Event Data in R*

Description

This package fits shared parameter models for the joint modelling of normal longitudinal responses and event times under a maximum likelihood approach. Various options for the survival model and optimization/integration algorithms are provided.

Details

Package:	JM
Type:	Package
Version:	0.4-0
Date:	2009-08-20
License:	GPL

The package has a single model-fitting function called `jointModel`, which accepts as main arguments a linear mixed effects object fit returned by function `lme()` of package **nlme**, and a survival object fit returned by either function `coxph()` or function `survreg()` of package **survival**. In addition, the `method` argument of `jointModel()` specifies the type of the survival submodel to be fitted and the type of the numerical integration technique; available options are:

`"ph-GH"` the time-dependent version of a proportional hazards model with unspecified baseline hazard function. The Gauss-Hermite integration rule is used to approximate the required integrals. (This option corresponds to the joint model proposed by Wulfsohn and Tsiatis, 1997)

`"weibull-PH-GH"` the Weibull model under the proportional hazards formulation. The Gauss-Hermite integration rule is used to approximate the required integrals.

`"weibull-AFT-GH"` the Weibull model under the accelerated failure time formulation. The Gauss-Hermite integration rule is used to approximate the required integrals.

`"piecewise-PH-GH"` a proportional hazards model with a piecewise constant baseline risk function. The Gauss-Hermite integration rule is used to approximate the required integrals.

`"ch-GH"` an additive log cumulative hazard model, in which the log cumulative baseline hazard is approximated using B-splines. The Gauss-Hermite integration rule is used to approximate the required integrals.

`"ch-Laplace"` an additive log cumulative hazard model, in which the log cumulative baseline hazard is approximated using B-splines. A fully exponential Laplace approximation method is used to approximate the required integrals (Rizopoulos et al., 2009).

Author(s)

Dimitris Rizopoulos

Maintainer: Dimitris Rizopoulos <d.rizopoulos@erasmusmc.nl>

References

Henderson, R., Diggle, P. and Dobson, A. (2000) Joint modelling of longitudinal measurements and event time data. *Biostatistics* **1**, 465–480.

Rizopoulos, D., Verbeke, G. and Lesaffre, E. (2009) Fully exponential Laplace approximation for the joint modelling of survival and longitudinal data. *Journal of the Royal Statistical Society, Series B* **71**, 637–654.

Rizopoulos, D., Verbeke, G. and Molenberghs, G. (2009) Multiple-imputation-based residuals and diagnostic plots for joint models of longitudinal and survival outcomes. *Biometrics*, to appear (doi: 10.1111/j.1541-0420.2009.01273.x).

Tsiatis, A. and Davidian, M. (2004) Joint modeling of longitudinal and time-to-event data: an overview. *Statistica Sinica* **14**, 809–834.

Wulfsohn, M. and Tsiatis, A. (1997) A joint model for survival and longitudinal data measured with error. *Biometrics* **53**, 330–339.

See Also

`jointModel`

---

jointModel *Joint Models for Longitudinal and Survival Data*

Description

This function fits shared parameter models for the joint modelling of normal longitudinal responses and time-to-event data under a maximum likelihood approach. Various options for the survival model are available.

Usage

```
jointModel(lmeObject, survObject, timeVar,
    method = c("weibull-AFT-GH", "weibull-PH-GH",
    "piecewise-PH-GH", "ch-GH", "ph-GH", "ch-Laplace"),
    init = NULL, control = list(), ...)
```

Arguments

`lmeObject`	an object inheriting from class `lme` (see also **Note**).
`survObject`	an object inheriting from class `coxph` or class `survreg`. In the call to `coxph()` or `survreg()`, you need to specify the argument `x = TRUE` such that the design matrix is contained in the object fit. See **Examples**.
`timeVar`	a character string indicating the time variable in the linear mixed effects model.
`method`	a character string specifying the type of joint model to fit. See **Details**.
`init`	a list of user-specified initial values. The initial values list must have the following components:

betas the vector of fixed effects for the linear mixed effects model.

sigma the measurement error standard deviation for the linear mixed effects model.

D the variance-covariance matrix of the random effects.

gammas the vector of baseline covariates for the survival model. For `method = "ch-GH"` or `method = "ch-Laplace"` this vector should first contain initial values for the sorted B-spline coefficients used to model the log cumulative baseline hazard.

alpha the association parameter.

sigma.t the scale parameter for the Weibull survival model; required only when `method = "weibull-AFT-GH"` or `method = "weibull-PH-GH"`.

lambda0 a vector of the baseline hazard values at the sorted unique event times; required only when `method = "ph-GH"`.

If the user-specified list of initial values does not contain some of these components or contains components not of the appropriate length, the default initial values are used instead.

`control` a list of control values with components:

only.EM logical; if `TRUE` only the EM algorithm is used in the optimization, otherwise if convergence has not been achieved a quasi-Newton algorithm is initiated. Default is `FALSE` except for `method = "ph-GH"` for which only the EM algorithm is available.

iter.EM the number of EM iterations. Default is 50 except for `method = "ph-GH"` for which the default is 200.

iter.qN the number of quasi-Newton iterations. Default is 150.

optimizer a character string indicating which optimizer to use; options are "optim" (default) and "nlminb".

tol1 tolerance value for convergence in the parameters; see **Details**. Default is 1e-03.

tol2 tolerance value for convergence in the parameters; see **Details**. Default is 1e-04.

tol3 tolerance value for convergence in the log-likelihood; see **Details**. Default is `sqrt(.Machine$double.eps)`.

numeriDeriv a character string indicating which type of numerical derivative to use to compute the Hessian matrix; options are "fd" (default) denoting the forward difference approximation, and "cd" denoting the central difference approximation.

eps.Hes tolerance value used in the numerical derivative method. Default is 1e-06; if you choose `numeriDeriv = "cd"` a larger value (e.g., 1e-04) is suggested.

parscale the `parscale` control argument for `optim()`, or the `scale` argument for `nlminb()`. It should be a numeric vector of length equal to the number of parameters. Default is 0.01 for all parameters.

step.max tolerance value for the maximum step size in the Newton-Raphson algorithm used to update the parameters of the survival submodel for `method = "ch-Laplace"`. Default is 0.1.

backtrackSteps the number of backtrack steps to use when updating the parameters of the survival submodel under `method = "ch-Laplace"`.

knots a numeric vector of the knots positions for the piecewise constant baseline risk function of for the log times used in the B-splines approximation of the log cumulative baseline hazard; therefore, this argument is relevant only when `method = "piecewise-PH-GH"` or when `method = "ch-GH"` or `method = "ch-Laplace"`. For `method = "piecewise-PH-GH"` default is to place equally-spaced `lng.in.kn` knots (see below) in the quantiles of the observed event times. For `method = "ch-GH"` and `method = "ch-Laplace"` default is to place `lng.in.kn` knots (see below) at the quantiles of the uncensored log survival times.

lng.in.kn the number of internal knots; relevant only when when `method = "piecewise-PH-GH"` where it denotes the number of internal knots for the piecewise constant baseline risk function or when `method = "ch-GH"` or `method = "ch-Laplace"` where it denotes the number of internal knots for B-splines approximation of the log cumulative baseline hazard. Default is 6 when `method = "piecewise-PH-GH"` and 3 otherwise.

ord a positive integer denoting the order of the B-splines used to approximate the log cumulative hazard (default is 4); relevant only when `method = "ch-GH"` or `method = "ch-Laplace"`.

GHk the number of Gauss-Hermite quadrature points used to approximate the integrals over the random effects. For `method = "weibull-AFT-GH"`, `method = "weibull-PH-GH"` and `method = "ch-GH"` default is 21 for one- or two-dimensional integration and 11 otherwise. For `method = "ph-GH"` default is 21 for one-dimensional integration, 11 for two-dimensional integration and 9 otherwise.

GKk the number of Gauss-Kronrod points used to approximate the integral involved in the calculation of the survival function. Two options are available, namely 7 or 15. For `method = "weibull-PH-GH"` and `method = "weibull-AFT-GH"` 15 are used, whereas for `method = "piecewise-PH-GH"` 7.

verbose logical; if `TRUE` the parameter estimates and the log-likelihood value are printed during the optimization procedure. Default is `FALSE`.

... options passed to the `control` argument.

Details

The `jointModel` function fits joint models for longitudinal and survival data. For the longitudinal responses the linear mixed effects model represented by the `lmeObject` is assumed. For the survival times three options are available. In particular, let t_i denote the time-to-event for the ith sample unit, x_i denote the vector of baseline covariates in `survObject`, with associated parameter vector γ, $W_i(t)$ the value of the longitudinal outcome at time point t (i.e., $W_i(t)$ equals the fixed effects part + random effects part of the linear mixed effects model for sample unit i), and α the association parameter. Then, for `method = "weibull-AFT-GH"` a time-dependent Weibull model under the accelerated failure time formulation is assumed. For `method = "weibull-PH-GH"` a time-dependent relative risk model is postulated with a Weibull baseline risk function. For `method = "piecewise-PH-GH"` a time-dependent relative risk model is postulated with

a piecewise constant baseline risk function. For `method = "ch-GH"` and `method = "ch-Laplace"` an additive model on the log cumulative hazard scale is assumed (see Rizopoulos et al., 2009 for more info). Finally, for `method = "ph-GH"` a time-dependent relative risk model is assumed where the baseline risk function is left unspecified (Wulfsohn and Tsiatis, 1997).

For all survival models except for the time-dependent proportional hazards model, the optimization algorithm starts with EM iterations, and if convergence is not achieved, it switches to quasi-Newton iterations (i.e., BFGS in `optim()` or `nlminb()`, depending on the value of the `optimizer` control argument). For the time-dependent proportional hazards model only the EM algorithm is used. During the EM iterations, convergence is declared if either of the following two conditions is satisfied: (i) $L(\theta^{it}) - L(\theta^{it-1}) < tol_3\{|L(\theta^{it-1})| + tol_3\}$, or (ii) $\max\{|\theta^{it} - \theta^{it-1}|/(|\theta^{it-1}| + tol_1)\} < tol_2$, where θ^{it} and θ^{it-1} is the vector of parameter values at the current and previous iterations, respectively, and $L(.)$ is the log-likelihood function. The values for tol_1, tol_2 and tol_3 are specified via the `control` argument. During the quasi-Newton iterations, the default convergence criteria of either `optim()` or `nlminb()` are used.

The required integrals are approximated using the Gauss-Hermite quadrature rule for `method = "weibull-AFT-GH"`, `method = "weibull-PH-GH"`, `method = "piecewise-PH-GH"`, `method = "ch-GH"` and `method = "ph-GH"`, whereas for `method = "ch-Laplace"` the fully exponential Laplace approximation described in Rizopoulos et al. (2008) is used. This last option is more suitable when high-dimensional random effects vectors are considered (e.g., when modelling nonlinear subject-specific trajectories with splines or high-order polynomials).

Value

See `jointModelObject` for the components of the fit.

Note

1. The `lmeObject` should represent a linear mixed model with a simple random effects structure, i.e., only the `pdDiag()` class is currently allowed.

2. The `lmeObject` should not contain any within-group correlation structure (i.e., `correlation` argument of `lme()`) or within-group heteroscedasticity structure (i.e., `weights` argument of `lme()`).

3. It is assumed that the linear mixed effects model `lmeObject` and the survival model `survObject` have been fitted to the same subjects. Moreover, it is assumed that the ordering of the subjects is the same for both `lmeObject` and `survObject`, i.e., that the first line in the data frame containing the event times corresponds to the first set of lines identified by the grouping variable in the data frame containing the repeated measurements, and so on.

4. In the `print` and `summary` generic functions for class `jointModel`, the estimated coefficients (and standard errors for the `summary` generic) for the event process are augmented with the element "Assoct" that corresponds to the association parameter α (see **Details**).

5. The standard errors returned by the `summary` generic function for class `jointModel` when `method = "ph-GH"` are based on the profile score vector (i.e., given the NPMLE for the unspecified baseline hazard). Hsieh et al. (2006) have noted that these standard errors are underestimated.

6. For more examples of `jointModel()` check the following link: http://wiki.r-project.org/rwiki/doku.php?id=packages:cran:jm.

Author(s)

Dimitris Rizopoulos ⟨d.rizopoulos@erasmusmc.nl⟩

References

Henderson, R., Diggle, P. and Dobson, A. (2000) Joint modelling of longitudinal measurements and event time data. *Biostatistics* **1**, 465–480.

Hsieh, F., Tseng, Y.-K. and Wang, J.-L. (2006) Joint modeling of survival and longitudinal data: Likelihood approach revisited. *Biometrics* **62**, 1037–1043.

Rizopoulos, D., Verbeke, G. and Lesaffre, E. (2009) Fully exponential Laplace approximation for the joint modelling of survival and longitudinal data. *Journal of the Royal Statistical Society, Series B* **71**, 637–654.

Rizopoulos, D., Verbeke, G. and Molenberghs, G. (2009) Multiple-imputation-based residuals and diagnostic plots for joint models of longitudinal and survival outcomes. *Biometrics*, to appear (doi: 10.1111/j.1541-0420.2009.01273.x).

Tsiatis, A. and Davidian, M. (2004) Joint modeling of longitudinal and time-to-event data: an overview. *Statistica Sinica* **14**, 809–834.

Wulfsohn, M. and Tsiatis, A. (1997) A joint model for survival and longitudinal data measured with error. *Biometrics* **53**, 330–339.

See Also

`jointModelObject`, `anova.jointModel`, `coef.jointModel`, `fixef.jointModel`, `ranef.jointModel`, `fitted.jointModel`, `residuals.jointModel`, `plot.jointModel`

Examples

```
# linear mixed model fit (random intercepts)
fitLME <- lme(log(serBilir) ~ drug * year, random = ~ 1 | id, data = pbc2)
# survival regression fit
fitSURV <- survreg(Surv(years, status2) ~ drug, data = pbc2.id, x = TRUE)
# joint model fit, under the (default) Weibull model
fitJOINT <- jointModel(fitLME, fitSURV, timeVar = "year")
fitJOINT
summary(fitJOINT)

# linear mixed model fit (random intercepts + random slopes)
fitLME <- lme(log(serBilir) ~ drug * year, random = ~ year | id, data = pbc2)
# survival regression fit
fitSURV <- survreg(Surv(years, status2) ~ drug, data = pbc2.id, x = TRUE)
# joint model fit, under the (default) Weibull model
fitJOINT <- jointModel(fitLME, fitSURV, timeVar = "year")
fitJOINT
summary(fitJOINT)

# linear mixed model fit
fitLME <- lme(sqrt(CD4) ~ obstime * drug - drug,
    random = ~ 1 | patient, data = aids)
# cox model fit
```

```
fitCOX <- coxph(Surv(Time, death) ~ drug, data = aids.id, x = TRUE)
# joint model fit, under the additive log cumulative hazard model
fitJOINT <- jointModel(fitLME, fitCOX,
    timeVar = "obstime", method = "ch-GH")
fitJOINT
summary(fitJOINT)
```

`jointModelObject` *Fitted jointModel Object*

Description

An object returned by the `jointModel` function, inheriting from class `jointModel` and representing a fitted joint model for longitudinal and time-to-event data. Objects of this class have methods for the generic functions `anova`, `coef`, `fitted`, `fixed.effects`, `logLik`, `plot`, `print`, `random.effects`, `residuals`, `summary`, and `vcov`.

Value

The following components must be included in a legitimate `jointModel` object.

`coefficients` a list with the estimated coefficients. The components of this list are:

betas the vector of fixed effects for the linear mixed effects model.

sigma the measurement error standard deviation for the linear mixed effects model.

gammas the vector of baseline covariates for the survival model. For `method = "ch-GH"` and `method = "ch-Laplace"` this vector contains as first elements the sorted B-spline coefficients used to model the log cumulative baseline hazard.

alpha the association parameter.

sigma.t the scale parameter for the Weibull survival model; returned only when `method = "weibull-GH"`.

lambda0 a two-column numeric matrix with the first column containing the estimated baseline hazard values, and the second the unique sorted event times; returned only when `method = "ph-GH"`.

D the variance-covariance matrix of the random effects.

`Hessian` the Hessian matrix evaluated at the estimated parameter values.

`logLik` the log-likelihood value.

`EB` a list with components:

post.b the estimated random effects values.

post.vb the estimated variance for the random effects estimates.

Zb the estimated random effects part of the linear predictor for the longitudinal outcome (i.e., Z is the design matrix for the random effects b).

Ztimeb the estimated random effects part of the linear predictor for the survival outcome (i.e., evaluated at the observed event times).

Ztime2b the estimated random effects part of the linear predictor for the survival outcome (i.e., for the ith sample unit is evaluated at all event times that are less or equal to the ith observed event time); returned only when `method = "ph-GH"`.

`knots` the numeric vector of the knots positions for the log times used in the B-splines approximation of the log cumulative baseline hazard; returned only when `method = "ch-GH"` or `method = "ch-Laplace"`.

`iters` the number of iterations in the optimization algorithm.

`convergence` convergence identifier: 0 corresponds to successful convergence, whereas 1 to a problem (i.e., when 1, usually more iterations are required).

`n` the number of sample units.

`N` the total number of repeated measurements for the longitudinal outcome.

`ni` a vector with the number of repeated measurements for each sample unit.

`d` a numeric vector with 0 denoting censored observation and 1 events.

`id` the grouping vector for the longitudinal responses.

`x` a list with the design matrices for the longitudinal and event processes.

`y` a list with the response vectors for the longitudinal and event processes.

`data.id` a `data.frame` containing the variables for the linear mixed effects model at the time of the event.

`method` the value of the `method` argument.

`termsY` the `terms` component of the `lmeObject`.

`termsT` the `terms` component of the `survObject`.

`formYx` the formula for the fixed effects part of the longitudinal model.

`formYz` the formula for the random effects part of the longitudinal model.

`formT` the formula for the survival model.

`timeVar` the value of the `timeVar` argument

`control` the value of the `control` argument.

`call` the matched call.

Author(s)

Dimitris Rizopoulos ⟨d.rizopoulos@erasmusmc.nl⟩

See Also

`jointModel`

pbc2 *Mayo Clinic Primary Biliary Cirrhosis Data*

Description

Followup of 312 randomised patients with primary biliary cirrhosis, a rare autoimmune liver disease, at Mayo Clinic.

Format

A data frame with 1945 observations on the following 20 variables.

id patients identifier; in total there are 312 patients.

years number of years between registration and the earlier of death, transplantion, or study analysis time.

status a factor with levels `alive`, `transplanted` and `dead`.

drug a factor with levels `placebo` and `D-penicil`.

age at registration in years.

sex a factor with levels `male` and `female`.

year number of years between enrollment and this visit date, remaining values on the line of data refer to this visit.

ascites a factor with levels `No` and `Yes`.

hepatomegaly a factor with levels `No` and `Yes`.

spiders a factor with levels `No` and `Yes`.

edema a factor with levels `No edema` (i.e., no edema and no diuretic therapy for edema), `edema no diuretics` (i.e., edema present without diuretics, or edema resolved by diuretics), and `edema despite diuretics` (i.e., edema despite diuretic therapy).

serBilir serum bilirubin in mg/dl.

serChol serum cholesterol in mg/dl.

albumin albumin in gm/dl.

alkaline alkaline phosphatase in U/liter.

SGOT SGOT in U/ml.

platelets platelets per cubic ml / 1000.

prothrombin prothrombin time in seconds.

histologic histologic stage of disease.

status2 a numeric vector with the value 1 denoting if the patient was transplanted or dead, and 0 if the patient was alive.

Note

The data frame `pbc2.id` contains the first measurement for each patient. This data frame is used to fit the survival model.

Source

http://lib.stat.cmu.edu/datasets/pbcseq

References

Fleming, T. and Harrington, D. (1991) *Counting Processes and Survival Analysis*. Wiley, New York.

Therneau, T. and Grambsch, P. (2000) *Modeling Survival Data: Extending the Cox Model*. Springer-Verlag, New York.

Examples

```
summary(pbc2.id)
```

---

plot *Plot Diagnostics for Joint Models*

Description

Produces a variety of plots for fitted joint models.

Usage

```
## S3 method for class 'jointModel':
plot(x, which = 1:4, caption = c("Residuals vs Fitted",
  "Normal Q-Q", "Marginal Survival", "Marginal Cumulative Hazard",
  "Marginal log Cumulative Hazard", "Baseline Hazard",
  "Cumulative Baseline Hazard", "Subject-specific Survival",
  "Subject-specific Cumulative Hazard",
  "Subject-specific log Cumulative Hazard"), survTimes = NULL,
  main = "",
  ask = prod(par("mfcol")) < length(which) && dev.interactive(),
  ..., ids = NULL, add.smooth = getOption("add.smooth"),
  add.qqline = TRUE, add.KM = FALSE, cex.caption = 1)
```

Arguments

x	an object inheriting from class jointModel.
which	which types of plots to produce, specify a subset of the numbers 1:10.
caption	captions to appear above the plots defined by argument which.
survTimes	a vector of survival times for which the survival, cumulative hazard or log cumulative hazard will be computed. Default is seq(minT, maxT, length = 15), where minT and maxT are the minimum and maximum observed survival times, respectively.
main	a character string specifying the title in the plot.

ask	logical; if TRUE, the user is asked before each plot, see par(ask=.).
...	other parameters to be passed through to plotting functions.
ids	a numeric vector specifying which subjects, the subject-specific plots will include; default is all subjects.
add.smooth	logical; if TRUE a smooth line is superimposed in the "Residuals vs Fitted" plot.
add.qqline	logical; if TRUE a qq-line is superimposed in the "Normal Q-Q" plot.
add.KM	logical; if TRUE the Kaplan-Meier estimate of the survival function is superimposed in the "Marginal Survival" plot.
cex.caption	magnification of captions.

Note

The plots of the baseline hazard and the cumulative baseline hazard are only produced when the joint model has been fitted using method = "ph-GH".

Author(s)

Dimitris Rizopoulos ⟨d.rizopoulos@erasmusmc.nl⟩

See Also

jointModel

Examples

```
# linear mixed model fit
fitLME <- lme(log(serBilir) ~ drug * year, random = ~ 1 | id, data = pbc2)
# survival regression fit
fitSURV <- survreg(Surv(years, status2) ~ drug, data = pbc2.id, x = TRUE)
# joint model fit, under the (default) Weibull model
fitJOINT <- jointModel(fitLME, fitSURV, timeVar = "year")

plot(fitJOINT, 3, add.KM = TRUE, col = "red", lwd = 2)

par(mfrow = c(2, 2))
plot(fitJOINT)
```

---

plot.survfitJM *Plot Method for survfitJM Objects*

Description

Produces plots of conditional probabilities of survival.

Usage

```
## S3 method for class 'survfitJM':
plot(x, estimator = c("both", "mean", "median"),
    which = NULL, fun = NULL, conf.int = TRUE, add.last.time.axis.tick = FALSE,
    include.y = FALSE, main = NULL, xlab = NULL, ylab = NULL, lty = NULL,
    col = NULL, lwd = NULL, pch = NULL, ask = NULL, legend = FALSE, ...)
```

Arguments

x	an object inheriting from class survfitJM.
estimator	character string specifying, whether to include in the plot the mean of the conditional probabilities of survival, the median or both. The mean and median are taken as estimates of these conditional probabilities over the M replications of the Monte Carlo scheme described in survfitJM.
which	a numeric or character vector specifying for which subjects to produce the plot. If a character vector, then is should contain a subset of the values of the idVar variable of the newdata argument of survfitJM.
fun	a vectorized function defining a transformation of the survival curve. For example with fun=log the log-survival curve is drawn.
conf.int	logical; if TRUE, then a pointwise confidence interval is included in the plot.
add.last.time.axis.tick	logical; if TRUE, a tick is added in the x-axis for the last available time point for which a longitudinal measurement was available.
include.y	logical; if TRUE, two plots are produced per subject, i.e., the plot of conditional probabilities of survival and a scatterplot of his longitudinal measurements.
main	a character string specifying the title in the plot.
xlab	a character string specifying the x-axis label in the plot.
ylab	a character string specifying the y-axis label in the plot.
lty	what types of lines to use.
col	which colors to use.
lwd	the thickness of the lines.
pch	the type of points to use.
ask	logical; if TRUE, the user is asked before each plot, see par(ask=.).
legend	logical; if TRUE, a legend is included in the plot.
...	extra graphical parameters passed to plot().

Author(s)

Dimitris Rizopoulos ⟨d.rizopoulos@erasmusmc.nl⟩

See Also

survfitJM

Examples

```
# linear mixed model fit
fitLME <- lme(sqrt(CD4) ~ obstime + obstime:drug,
    random = ~ 1 | patient, data = aids)
# cox model fit
fitCOX <- coxph(Surv(Time, death) ~ drug, data = aids.id, x = TRUE)

# joint model fit
fitJOINT <- jointModel(fitLME, fitCOX,
    timeVar = "obstime", method = "weibull-PH-GH")

# sample of the patients who are still alive
ND <- aids[aids$patient == "141", ]
ss <- survfitJM(fitJOINT, newdata = ND, idVar = "patient", M = 50)
plot(ss)
plot(ss, include.y = TRUE, add.last.time.axis.tick = TRUE, legend = TRUE)
```

---

prothro *Prednisone versus Placebo in Liver Cirrhosis Patients*

Description

A randomized trial on 488 liver cirrhosis patients

Format

Two data frames with the following variable.

id patients identifier; in total there are 467 patients.

pro prothrobin measurements.

time for data frame `prothro` the time points at which the prothrobin measurements were taken; for data frame `prothros` the time to death or censoring.

death a numeric vector with 0 denoting censoring and 1 death.

treat randomized treatment; a factor with levels "placebo" and "prednisone".

Source

http://www.gllamm.org/books/readme.html#14.6,

References

Andersen, P. K., Borgan, O., Gill, R. D. and Keiding, N. (1993). *Statistical Models Based on Counting Processes*. New York: Springer.

Examples

```
summary(prothros)
```

ranef *Random Effects Estimates for Joint Models*

Description

Extracts the random effects estimates from a fitted joint model.

Usage

```
## S3 method for class 'jointModel':
ranef(object, postVar = FALSE, ...)
```

Arguments

object	an object inheriting from class jointModel.
postVar	logical; if TRUE the variance-covariance matrix of these estimates is also returned. See **Details** for more info.
...	additional arguments; currently none is used.

Details

For methods "ch-GH", "weibull-GH", "ph-GH" the ranef() method returns the posterior means (and posterior variances if postVar = TRUE), whereas for method "ch-Laplace" it returns the posterior modes (and the inverse of the negative of the second-order derivative of the log-posterior with respect to the random effects, if postVar = TRUE).

Value

a numeric matrix with rows denoting the individuals and columns the random effects (e.g., intercepts, slopes, etc.). If postVar = TRUE, the numeric matrix has an extra attribute "postVar".

Author(s)

Dimitris Rizopoulos ⟨d.rizopoulos@erasmusmc.nl⟩

See Also

coef.jointModel, fixef.jointModel

Examples

```
# linear mixed model fit
fitLME <- lme(log(serBilir) ~ drug * year, random = ~ 1 | id, data = pbc2)
# survival regression fit
fitSURV <- survreg(Surv(years, status2) ~ drug, data = pbc2.id, x = TRUE)

# joint model fit, under the (default) Weibull model
fitJOINT <- jointModel(fitLME, fitSURV, timeVar = "year")
ranef(fitJOINT)
```

`residuals` *Residuals for Joint Models*

Description

Calculates residuals for joint models.

Usage

```
## S3 method for class 'jointModel':
residuals(object, process = c("Longitudinal", "Event"),
  type = c("Marginal", "Subject", "stand-Marginal",
  "stand-Subject", "Martingale", "CoxSnell", "AFT"),
  MI = FALSE, M = 50, time.points = NULL, return.data = FALSE,
  ...)
```

Arguments

`object`	an object inheriting from class `jointModel`.
`process`	for which model (i.e., linear mixed model or survival model) to calculate residuals.
`type`	what type of residuals to calculate. See **Details**.
`MI`	logical; if `TRUE` multiple-imputation-based residuals are calculated.
`M`	integer denoting how many imputations to use for the MI residuals.
`time.points`	for fixed visit times, this should be a numeric vector with the unique times points at which longitudinal measurements are supposed to be taken; if `NULL`, then the code attempts to extract these unique time points using the design matrix for the fixed effects of the longitudinal model and the value of the `timeVar` argument of `jointModel`. For random visit times, this should be an object of class `weibull.frailty` that represents the fit of Weibull model with Gamma frailties for the visiting process. The user may also augment the object `weibull.frailty` with the following two attributes: `"prev.y"` denoting the variable name for the previous longitudinal responses, and `"tmax"` denoting the end of the study.
`return.data`	logical; if `TRUE` and `MI = TRUE` and fixed visit times are considered, then the multiply imputed data sets are returned.
`...`	additional arguments; currently none is used.

Details

When `process = "Longitudinal"`, residuals are calculated for the longitudinal outcomes. In particular, if `type = "Marginal"` these are $e_{ij} = y_{ij} - x_{ij}^T \hat{\beta}$, whereas for `type = "Subject"`,

$e_{ij} = y_{ij} - x_{ij}^T \hat{\beta} - z_{ij}^T b_i$, where i denotes the subject and j the measurement, y_{ij} the longitudinal responses, x_{ij}^T and z_{ij}^T the corresponding rows of the fixed and random effects design matrices, respectively, and β and b_i denote the fixed effects and random effects components. If `type = "stand-Marginal"` or `type = "stand-Subject"`, the above defined residuals are divided by the estimated standard deviation of the corresponding error term. If `MI = TRUE`, multiple-imputation-based residuals are calculated for the longitudinal process; for more information regarding these residuals, check Rizopoulos et al. (2009).

When `process = "Event"`, residuals are calculated for the survival outcome. Martingale residuals are available for all options for the survival submodel (for the different options of survival submodel, check the `method` argument of `jointModel`). Cox-Snell residuals (Cox and Snell, 1968) are available for the Weibull model and the additive log cumulative hazard model. AFT residuals are only available for the Weibull model.

Value

If `MI = FALSE`, a numeric vector of residual values. Otherwise a list with components:

`fitted.values`
: the fitted values for the observed data.

`residuals`
: the residuals for the observed data.

`fitted.valsM`
: the fitted values for the missing data.

`resid.valsM`
: the multiply imputed residuals for the missing longitudinal responses.

`mean.resid.valsM`
: the average of the multiply imputed residuals for the missing longitudinal responses; returned only if fixed visit times are considered.

`dataM`
: if `return.data = TRUE` and fixed visit times are considered, then it returns the data set with the simulated response values for the longitudinal outcome, for each of the multiple imputations.

Note

The multiple-imputation-based residuals are not available for joint models with `method = "ph-GH"`.

Author(s)

Dimitris Rizopoulos ⟨d.rizopoulos@erasmusmc.nl⟩

References

Cox, D. and Snell, E. (1968) A general definition of residuals. *Journal of the Royal Statistical Society, Series B* **30**, 248–275.

Rizopoulos, D., Verbeke, G. and Molenberghs, G. (2009) Multiple-imputation-based residuals and diagnostic plots for joint models of longitudinal and survival outcomes. *Biometrics*, to appear (doi: 10.1111/j.1541-0420.2009.01273.x).

See Also

fitted.jointModel

Examples

```
# linear mixed model fit
fitLME <- lme(sqrt(CD4) ~ obstime * drug - drug,
    random = ~ 1 | patient, data = aids)
# cox model fit
fitCOX <- coxph(Surv(Time, death) ~ drug, data = aids.id, x = TRUE)

# joint model fit, under the additive log cumulative hazard model
fitJOINT <- jointModel(fitLME, fitCOX,
    timeVar = "obstime", method = "ch-GH")

# residuals for the longitudinal outcome
head(cbind(
    "Marginal" = residuals(fitJOINT),
    "std-Marginal" = residuals(fitJOINT, type = "stand-Marginal"),
    "Subject" = residuals(fitJOINT, type = "Subject"),
    "std-Subject" = residuals(fitJOINT, type = "stand-Subject")
))

# residuals for the survival outcome
head(cbind(
    "Martingale" = residuals(fitJOINT, process = "Event", type = "Martingale"),
    "CoxSnell" = residuals(fitJOINT, process = "Event", type = "CoxSnell")
))
```

---

summary.weibull.frailty

Summary Method for weibull.frailty Objects

Description

Summarizes the fit of a Weibull model with Gamma frailties

Usage

```
## S3 method for class 'weibull.frailty':
summary(object, sand.se = FALSE, ...)
```

Arguments

object	an object inheriting from class weibull.frailty.
sand.se	logical; if TRUE, sandwich standard errors are also produced.
...	additional arguments; currently none is used.

Author(s)

Dimitris Rizopoulos ⟨d.rizopoulos@erasmusmc.nl⟩

See Also

weibull.frailty

Examples

```
fit <- weibull.frailty(Surv(time, status) ~ age + sex, kidney)
summary(fit)
summary(fit, TRUE)
```

---

survfitJM *Prediction in Joint Models*

Description

This function computes the conditional probability of surviving later times than the last observed time for which a longitudinal measurement was available.

Usage

```
survfitJM(object, newdata, idVar = "id", survTimes = NULL,
        last.time = NULL, M = 200, CI.levels = c(0.025, 0.975), scale = 1.6)
```

Arguments

object	an object inheriting from class jointModel.
newdata	a data frame that contains the longitudinal and covariate information for the subjects for which prediction of survival probabilities is required. The names of the variables in this data frame must be the same as in the data frames that were used to fit the linear mixed effects model (using lme()) and the survival model (using coxph() or survreg()) that were supplied as the two first argument of jointModel. In addition, this data frame should contain a variable that identifies the different subjects (see also argument idVar).
idVar	the name of the variable in newdata that identifies the different subjects.
survTimes	a numeric vector of times for which prediction survival probabilities are to be computed.
last.time	a numeric vector or character string. This specifies the known time at which each of the subjects in newdat was known to be alive. If NULL, then this is automatically taken as the last time each subject provided a longitudinal measurement. If a numeric vector, then it is assumed to contain this last time point for each subject. If a character string, then it should be a variable in the data frame newdata.
M	integer denoting how many Monte Carlo samples to use – see **Details**.

`CI.levels` a numeric vector of length two that specifies which quantiles to use for the calculation of confidence interval for the predicted probabilities – see **Details**.

`scale` a numeric value that controls the acceptance rate of the Metropolis-Hastings algorithm – see **Details**.

Details

Based on a fitted joint model (represented by `object`), and a history of longitudinal responses $\tilde{y}_i(t) = \{y_i(s), 0 \leq s \leq t\}$ and a covariates vector x_i (stored in `newdata`), this function provides estimates of $Pr(T_i > u | T_i > t, \tilde{y}_i(t), x_i)$, i.e., the conditional probability of surviving time u given that subject i, with covariate information x_i, has survived up to time t and has provided longitudinal the measurements $\tilde{y}_i(t)$.

To estimate $Pr(T_i > u | T_i > t, \tilde{y}_i(t), x_i)$ a Monte Carlo procedure is followed with the following steps:

Step 1: Simulate new parameter values, say θ^*, from $N(\hat{\theta}, C(\hat{\theta}))$, where $\hat{\theta}$ are the MLEs and $C(\hat{\theta})$ their large sample covariance matrix, which are extracted from `object`.

Step 2: Simulate random effects values, say b_i^*, from their posterior distribution given survival up to time t, the vector of longitudinal responses $\tilde{y}_i(t)$ and θ^*. This is achieved using a Metropolis-Hastings algorithm with independent proposals from a properly centered and scaled multivariate t distribution. The `scale` argument controls the acceptance rate for this algorithm.

Step 3 Using θ^* and b_i^*, compute $Pr(T_i > u | T_i > t, b_i^*, x_i; \theta^*)$.

Step 4: Repeat Steps 1-3 `M` times.

Based on the `M` estimates of the conditional probabilities, we compute useful summary statistics, such as their mean, median, and quantiles (to produce a confidence interval).

Value

A list of class `survfitJM` with components:

`summaries` a list with elements numeric matrices with numeric summaries of the predicted probabilities for each subect.

`survTimes` a copy of the `survTimes` argument.

`last.time` a numeric vector with the time of the last available longitudinal measurement of each subject.

`obs.times` a list with elements numeric vectors denoting the timings of the longitudinal measurements for each subject.

`y` a list with elements numeric vectors denoting the longitudinal responses for each subject.

`full.results` a list with elements numeric matrices with predicted probabilities for each subject in each replication of the Monte Carlo scheme described above.

`success.rate` a numeric vector with the success rates of the Metropolis-Hastings algorithm described above for each subject.

`scale` a copy of the `scale` argument.

Note

Predicted probabilities are not computed for joint models with `method = "ch-GH"`, `method = "ch-Laplace"` and `method = "ph-GH"`.

Author(s)

Dimitris Rizopoulos ⟨d.rizopoulos@erasmusmc.nl⟩

See Also

`jointModel`, `plot.survfitJM`

Examples

```
# linear mixed model fit
fitLME <- lme(sqrt(CD4) ~ obstime + obstime:drug,
    random = ~ 1 | patient, data = aids)
# cox model fit
fitCOX <- coxph(Surv(Time, death) ~ drug, data = aids.id, x = TRUE)

# joint model fit
fitJOINT <- jointModel(fitLME, fitCOX,
    timeVar = "obstime", method = "weibull-PH-GH")

# sample of the patients who are still alive
ND <- aids[aids$patient == "141", ]
ss <- survfitJM(fitJOINT, newdata = ND, idVar = "patient", M = 50)
ss
```

`weibull.frailty` *Weibull Model with Gamma Frailties*

Description

Fits a Weibull model with Gamma frailties for multivariate survival data under maximum likelihood

Usage

```
weibull.frailty(formula = formula(data), data = parent.frame(),
    id = "id", subset, na.action, init, control = list())
```

Arguments

`formula`	an object of class `formula`: a symbolic description of the model to be fitted. The response must be a survival object as returned by function `Surv()`.
`data`	an optional data frame containing the variables specified in the model.

`id` either a character string denoting a variable name in `data` or a numeric vector specifying which event times belong to the same cluster (e.g., hospital, patient, etc.).

`subset` an optional vector specifying a subset of observations to be used in the fitting process.

`na.action` what to do with missing values.

`init` a numeric vector of length $p + 3$ of initial values. The first p elements should correspond to the regression coefficients for the covariates, and the last 3 to log-scale, log-shape, and log-frailty-variance, respectively. See **Details**.

`control` a list of control values with components:

optimizer a character string indicating which optimizer to use; options are "optim" (default) and "nlminb".

parscale the `parscale` control argument for `optim()`, or the `scale` argument for `nlminb()`. It should be a numeric vector of length equal to the number of parameters. Default is 0.01 for all parameters.

maxit the maximum number of iterations. Default is 500.

numeriDeriv a character string indicating which type of numerical derivative to use to compute the Hessian matrix; options are "fd" denoting the forward difference approximation, and "cd" (default) denoting the central difference approximation.

eps.Hes tolerance value used in the numerical derivative method. Default is 1e-03.

Details

The fitted model is defined as follows:

$$\lambda(t_i|\omega_i) = \lambda_0(t_i)\omega_i \exp(x_i^T\beta),$$

where i denotes the subject, $\lambda(\cdot)$ denotes the hazard function, conditionally on the frailty ω_i, x_i is a vector of covariates with corresponding regression coefficients β, and $\lambda_0(\cdot)$ is the Weibull baseline hazard defined as $\lambda_0(t) = shape * scale * t^{shape-1}$. Finally, for the frailties we assume $\omega_i \sim Gamma(\eta, \eta)$, with η^{-1} denoting the unknown variance of ω_i's.

Value

an object of class `weibull.frailty` with components:

`coefficients` a list with the estimated coefficients values. The components of this list are: `betas`, `scale`, `shape`, and `var.frailty`, and correspond to the coefficients with the same name.

`hessian` the hessian matrix at convergence. For the shape, scale, and var-frailty parameters the Hessian is computed on the log scale.

`logLik` the log-likelihood value.

`control` a copy of the `control` argument.

`y` an object of class `Surv` containing the observed event times and the censoring indicator.

x	the design matrix of the model.
id	a numeric vector specifying which event times belong to the same cluster.
nam.id	the value of argument id, if that was a character string.
terms	the term component of the fitted model.
data	a copy of data or the created model.frame.
call	the matched call.

Note

weibull.frailty() currently supports only right-censored data.

Author(s)

Dimitris Rizopoulos ⟨d.rizopoulos@erasmusmc.nl⟩

Examples

```
weibull.frailty(Surv(time, status) ~ age + sex, kidney)
```

Index