Package 'joineR'

May 8, 2026

Type Package

Title Joint Modelling of Repeated Measurements and Time-to-Event Data

Version 1.2.8

Encoding UTF-8

Description Analysis of repeated measurements and time-to-event data via random effects joint models. Fits the joint models proposed by Henderson and colleagues <doi:10.1093/biostatistics/1.4.465> (single event time) and by Williamson and colleagues (2008) <doi:10.1002/sim.3451> (competing risks events time) to a single continuous repeated measure. The time-to-event data is modelled using a (cause-specific) Cox proportional hazards regression model with time-varying covariates. The longitudinal outcome is modelled using a linear mixed effects model. The association is captured by a latent Gaussian process. The model is estimated using am Expectation Maximization algorithm. Some plotting functions and the variogram are also included. This project is funded by the Medical Research Council (Grant numbers G0400615 and MR/M013227/1).

License GPL-3 | file LICENSE

URL https://github.com/graemeleehickey/joineR/

BugReports https://github.com/graemeleehickey/joineR/issues

LazyData true

ByteCompile true

Depends R (>= 3.6), survival

Imports graphics, lattice, MASS, nlme, statmod, stats, utils

Suggests knitr, rmarkdown, testthat, covr

VignetteBuilder knitr

RoxygenNote 7.1.2

NeedsCompilation no

Author Pete Philipson [aut] (ORCID: <https://orcid.org/0000-0001-7846-0208>), Ines Sousa [aut] (ORCID: <https://orcid.org/0000-0002-2712-1713>), Peter J. Diggle [aut] (ORCID: <https://orcid.org/0000-0003-3521-5020>), Paula Williamson [aut] (ORCID: <https://orcid.org/0000-0001-9802-6636>),

Ruwanthi Kolamunnage-Dona [aut] (ORCID:
<https://orcid.org/0000-0003-3886-6208>),
Robin Henderson [aut],
Graeme L. Hickey [aut, cre] (ORCID:
<https://orcid.org/0000-0002-4989-0054>),
Maria Sudell [ctb],
Medical Research Council [fnd] (Grant numbers: G0400615 and
MR/M013227/1)

Maintainer Graeme L. Hickey <graemeleehickey@gmail.com>

Repository CRAN

Date/Publication 2023-01-22 21:20:06 UTC

Contents

aids . . . 3
epileptic . . . 4
heart.valve . . . 5
joineR . . . 6
joint . . . 7
joint.object . . . 11
jointdata . . . 13
jointplot . . . 14
jointSE . . . 16
lines.jointdata . . . 18
liver . . . 19
mental . . . 20
plot.jointdata . . . 21
plot.vargm . . . 22
points.jointdata . . . 23
sample.jointdata . . . 24
simjoint . . . 25
subset.jointdata . . . 28
summary.joint . . . 29
summary.jointdata . . . 30
summarybal . . . 31
to.balanced . . . 32
to.unbalanced . . . 33
UniqueVariables . . . 34
variogram . . . 35

Index **37**

aids *AIDS drug trial data*

Description

This dataset describes a randomized clinical trial (Goldman et al., 1996) in which both survival and longitudinal data were collected to compare the efficacy and safety of two antiretroviral drugs, namely ddI (didanosine) and ddC (zalcitabine), in treating HIV-infected patients intolerant or failing zidovudine (AZT) therapy.

Usage

```
data(aids)
```

Format

A `data.frame` in the unbalanced format with 1405 longitudinal observations from 467 subjects. The columns are:

`id` integer: number for patient identification.

`time` numeric: time to death (or censoring).

`death` integer: event indicator. Coded as `0` = right-censoring, and `1` = death.

`obstime` numeric: measurement times for the repeated CD4 count measurements.

`CD4` numeric: CD4 cell counts measured at `obstime`.

`drug` factor: drug indicator. Coded as `ddI` = didanosine and `ddC` = zalcitabine.

`gender` factor: gender indicator. Coded as `male` and `female`.

`prevOI` factor: opportunistic infection indicator. Coded as `AIDS` = AIDS diagnosis at study entry, and `noAIDS` = no previous infection.

`AZT` factor: AZT intolerance/failure indicator. Coded as `intolerance` or `failure`.

Source

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

References

Goldman A, Carlin B, Crane L, Launer C, Korvick J, Deyton L, Abrams D. 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*. 1996; **11**: 161-169 URL: http://www.biostat.umn.edu/~brad/data.html.

See Also

`heart.valve`, `epileptic`, `mental`, `liver`.

`epileptic` *Dose calibration of anti-epileptic drugs data*

Description

The SANAD (Standard and New Anti-epileptic Drugs) study (Marson et al., 2007) is a randomized control trial of standard and new anti-epileptic drugs, comparing effects on longer term clinical outcomes. The data consists of longitudinal measurements of calibrated dose for the groups randomized to a standard drug (CBZ) and a new drug (LTG). The objective of the analysis is to investigate the effect of drug titration on the relative effects of LTG and CBZ on treatment failure (withdrawal of the randomized drug). There are several baseline covariates available, and also data on the time to withdrawal from randomized drug.

Usage

```
data(epileptic)
```

Format

This is a data frame in the unbalanced format, that is, with one row per observation. The data consists of columns for patient identifier, time of measurement, calibrated dose, baseline covariates, and survival data. The column names are identified as follows:

`id` integer: patient identifier.

`dose` numeric: calibrated dose.

`time` integer: timing of clinic visit at which dose recorded (days).

`with.time` integer: time of drug withdrawal/maximum follow up time (days).

`with.status` censoring indicator (`1` = withdrawal of randomized drug and `0` = not withdrawn from randomized drug/lost to follow up).

`with.status2` censoring indicator (`1` = withdrawal of randomized drug due to inadequate seizure control, (`2` = withdrawal of randomized drug due to unacceptable adverse effects, and `0` = not withdrawn from randomized drug/lost to follow up).

`with.status.uae` `1` = withdrawal due to unacceptable adverse effects, `0` = otherwise.

`with.status.isc` `1` = withdrawal due to inadequate seizure control, `0` = otherwise.

`treat` factor: randomized treatment (CBZ or LTG).

`age` numeric: age of patient at randomization (years).

`gender` factor: gender of patient. `F` = female, `M` = male.

`learn.dis` factor: learning disability.

Source

SANAD Trial Group, University of Liverpool

References

Marson AG, Appleton R, Baker GA, et al. A randomised controlled trial examining longer-term outcomes of standard versus new antiepileptic drugs. The SANAD Trial. *Health Tech Assess*. 2007; **11(37)**.

Marson AG, Al-Kharusi AM, Alwaidh M, et al. The SANAD study of effectiveness of carbamazepine, gabapentin, lamotrigine, oxcarbazepine, or topiramate for treatment of partial epilepsy: an unblinded randomised controlled trial. *Lancet*. 2007; **365**: 2007-2013.

Williamson PR, Kolamunnage-Dona R, Philipson P, Marson AG. Joint modelling of longitudinal and competing risks data. *Stats Med.* 2008; **27(30)**: 6426-6438.

See Also

`heart.valve`, `liver`, `mental`, `aids`.

`heart.valve` *Aortic valve replacement surgery data*

Description

This is longitudinal data on an observational study on detecting effects of different heart valves, differing on type of tissue, implanted in the aortic position. The data consists of longitudinal measurements on three different heart function outcomes, after surgery occurred. There are several baseline covariates available, and also survival data.

Usage

```
data(heart.valve)
```

Format

This is a data frame in the unbalanced format, that is, with one row per observation. The data consists in columns for patient identification, time of measurements, longitudinal multiple longitudinal measurements, baseline covariates, and survival data. The column names are identified as follows:

`num` number for patient identification.

`sex` gender of patient (`0` = Male and `1` = Female).

`age` age of patient at day of surgery (years).

`time` observed time point, with surgery date as the time origin (years).

`fuyrs` maximum follow up time, with surgery date as the time origin (years).

`status` censoring indicator (`1` = died and `0` = lost at follow up).

`grad` valve gradient at follow-up visit.

`log.grad` natural log transformation of `grad`.

`lvmi` left ventricular mass index (standardised) at follow-up visit.

`log.lvmi` natural log transformation of `lvmi`.

`ef` ejection fraction at follow-up visit.

`bsa` preoperative body surface area.

`lvh` preoperative left ventricular hypertrophy.

`prenyha` preoperative New York Heart Association (NYHA) classification (`1` = I/II and `3` = III/IV).

`redo` previous cardiac surgery.

`size` size of the valve (millimeters).

`con.cabg` concomitant coronary artery bypass graft.

`creat` preoperative serum creatinine (μmol/mL).

`dm` preoperative diabetes.

`acei` preoperative use of ace inhibitor.

`lv` preoperative left ventricular ejection fraction (LVEF) (`1` = good, `2` = moderate, and `3` = poor).

`emergenc` operative urgency (`0` = elective, `1` = urgent, and `3` = emergency).

`hc` preoperative high cholesterol (`0` = absent, `1` = present treated, and `2` = present untreated).

`sten.reg.mix` aortic valve haemodynamics (`1` = stenosis, `2` = regurgitation, `3` = mixed).

`hs` implanted aortic prosthesis type (`1` = homograft and `0` = stentless porcine tissue).

References

Lim E, Ali A, Theodorou P, Sousa I, Ashrafian H, Chamageorgakis T, Duncan M, Diggle P, Pepper J. A longitudinal study of the profile and predictors of left ventricular mass regression after stentless aortic valve replacement. *Ann Thorac Surg.* 2008; **85(6)**: 2026-2029.

See Also

`mental`, `liver`, `epileptic`, `aids`.

`joineR` *joineR*

Description

The joineR package implements methods for analyzing data from longitudinal studies in which the response from each subject consists of a time-sequence of repeated measurements and a possibly censored time-to-event outcome. The modelling framework for the repeated measurements is the linear model with random effects and/or correlated error structure (Laird and Ware, 1982). The model for the time-to-event outcome is a: Cox proportional hazards model with log-Gaussian frailty (Cox, 1972). A cause-specific hazards model is used when competing risks are present. Stochastic dependence is captured by allowing the Gaussian random effects of the linear model to be correlated with the frailty term of the Cox proportional hazards model. The methodology used to fit the model is described in Henderson et al. (2002) in the case of a single event time, and by Williamson et al. (2008) in the case of competing risks data. Both models exploit the general methodology proposed by Wulfsohn and Tsiatis (1997).

The package offers several types of functions for the analysis of joint data.

Data manipulation functions

There are several functions, including `jointdata`, `sample.jointdata`, `subset.jointdata`, `to.balanced`, `to.unbalanced`, and `UniqueVariables`, which offer the ability to construct a joint model dataset and manipulate it, e.g. take a sample according to a baseline covariate or outcome.

Plot functions

The plot function can be applied to `jointdata` and `vargm` (variogram) objects. In addition, `points` and `lines` can also be used with `jointplot` objects.

Model fitting functions

The primary function for fitting a joint model is `joint`. Standard errors can be estimated using `jointSE`.

Note

Further details on the package are given in the vignette. To access this, run `vignette("joineR")`.

References

Wulfsohn MS, Tsiatis AA. A joint model for survival and longitudinal data measured with error. *Biometrics.* 1997; **53(1)**: 330-339.

Henderson R, Diggle PJ, Dobson A. Joint modelling of longitudinal measurements and event time data. *Biostatistics.* 2000; **1(4)**: 465-480.

Cox DR. Regression models and life-tables. *J R Stat Soc Ser B Stat Methodol.* 1972; **34(2)**: 187-220.

Laird NM, Ware JH. Random-effects models for longitudinal data. *Biometrics.* 1982; **38(4)**: 963-974.

Williamson PR, Kolamunnage-Dona R, Philipson P, Marson AG. Joint modelling of longitudinal and competing risks data. *Stat Med.* 2008; **27**: 6426-6438.

`joint` *Fit joint model for survival and longitudinal data measured with error*

Description

This generic function fits a joint model with random latent association, building on the formulation described in Wulfsohn and Tsiatis (1997) while allowing for the presence of longitudinal and survival covariates, and three choices for the latent process. The link between the longitudinal and survival processes can be proportional or separate. When failure is attributable to 2 separate causes, a competing risks joint model is fitted as per Williamson et al. (2008).

Usage

```
joint(
  data,
  long.formula,
  surv.formula,
  model = c("intslope", "int", "quad"),
  sepassoc = FALSE,
  longsep = FALSE,
  survsep = FALSE,
  gpt,
  lgpt,
  max.it,
  tol,
  verbose = FALSE
)
```

Arguments

data	an object of class jointdata containing the variables named in the formulae arguments.
long.formula	a formula object with the response variable, and the covariates to include in the longitudinal sub-model.
surv.formula	a formula object with the survival time, censoring indicator and the covariates to include in the survival sub-model. The response must be a survival object as returned by the Surv function.
model	a character string specifying the type of latent association. This defaults to the intercept and slope version as seen in Wulfsohn and Tsiatis (1997). For association via the random intercept only, choose model = "int", whereas for a quadratic association, use model = "quad". Computing times are commensurate with the type of association structure chosen.
sepassoc	logical value: if TRUE then the joint model is fitted with separate association, see Details.
longsep	logical value: if TRUE, parameter estimates and log-likelihood from a separate linear mixed model analysis of the longitudinal data (see the lme function for details) are returned.
survsep	if TRUE, parameter estimates and log-likelihood from a separate analysis of the survival data using the Cox proportional hazards model are returned (see coxph function for details).
gpt	the number of quadrature points across which the integration with respect to the random effects will be performed. Defaults to gpt = 3 which produces stable estimates in most datasets.
lgpt	the number of quadrature points which the log-likelihood is evaluated over following a model fit. This defaults to lgpt = 10, though lgpt = 3 is often sufficient.

`max.it`	the maximum number of iterations of the EM algorithm that the function will perform. Defaults to `max.it = 200`, though more iterations may be necessary for large, complex data.
`tol`	the tolerance level before convergence of the algorithm is deemed to have occurred. Default value is `tol = 0.001`.
`verbose`	if `TRUE`, the parameter estimates at each iteration of the EM algorithm are printed. Default is `verbose = FALSE`.

Details

The `joint` function fits a joint model to survival and longitudinal data. The formulation is similar to Wulfsohn and Tsiatis (1997). A linear mixed effects model is assumed for the longitudinal data, namely

$$Y_i = X_{i1}(t_i)^T \beta_1 + D_i(t_i)^T U_i + \epsilon_i,$$

where U_i is a vector of random effects, $(U_{0i}, \ldots U_{qi})$ whose length depends on the model chosen, i.e. $q = 1$ for the random intercept model. D_i is the random effects covariate matrix, which will be time-dependent for all but the random intercept model. X_{i1} is the longitudinal design matrix for unit i, and t_i is the vector of measurement times for subject i. Measurement error is represented by ϵ_i.

The Cox proportional hazards model is adopted for the survival data, namely

$$\lambda(t) = \lambda_0(t) \exp\{X_{i2}(t)^T \beta_2 + D_i(t)(\gamma^T U_i)\}.$$

The parameter γ determines the level of association between the two processes. For the intercept and slope model with separate association we have

$$D_i(t)(\gamma^T U_i) = \gamma_0 U_{0i} + \gamma_1 U_{1i} t,$$

whereas under proportional association

$$D_i(t)(\gamma^T U_i) = \gamma(U_{0i} + U_{1i} t).$$

X_{i2} is the vector of survival covariates for unit i. The baseline hazard function is $\lambda_0(t)$.

The function uses an EM algorithm to estimate parameters in the joint model. Starting values are provided by calls to standard R functions `lme` and `coxph` for the longitudinal and survival components, respectively.

Value

A list containing the parameter estimates from the joint model and, if required, from either or both of the separate analyses. The combined log-likelihood from a separate analysis and the log-likelihood from the joint model are also produced as part of the fit.

Competing risks

If failure can be attributed to 2 causes, i.e. so-called competing risks events data, then a cause-specific hazards model is adopted, namely

$$\lambda_g(t) = \lambda_{0g}(t) \exp\{X_{i2}(t)^T \beta_2^{(g)} + D_i(t)(\gamma^T U_i)\},$$

where $g = 1, 2$ denotes the failure type, $\beta_2^{(g)}$ $(g = 1, 2)$ are cause-specific hazard parameters corresponding to the same covariates, and $\lambda_{0g}(t)$ are cause-specific baseline hazard functions. For this data, a proportional association structure is assumed (i.e. `sepassoc = FALSE`) and a random-intercepts and random-slopes model must be used (i.e. `model = "intslope"`). Note that the function only permits 2 failure types. The model is specified in full by Williamson et al. (2008). The function `joint()` automatically detects whether competing risks are present by counting the number of unique components in the event column on the event time data.

Separate models

Both `longsep` and `survsep` ignore any latent association (i.e. $\gamma = 0$) between the longitudinal and survival processes but their output can be used to compare with the results from the joint model. If interest is solely in the individual processes then the user should instead make use of the functions `lme` and `coxph` mentioned above. Furthermore, if interest is in the separate effect of each random effect (this is for intercept and slope or quadratic models only) upon the survival data, the user should set `sepassoc = TRUE`.

Note

Since numerical integration is required, it is advisable to check the stability of the maximum likelihood estimates with an increasing number of Gauss-Hermite quadrature points. `joint()` uses `gpt = 3` by default, as this has been adequate for many datasets. However, for certain datasets and models, this might be too small.

Author(s)

Pete Philipson

References

Wulfsohn MS, Tsiatis AA. A joint model for survival and longitudinal data measured with error. *Biometrics.* 1997; **53(1)**: 330-339.

Henderson R, Diggle PJ, Dobson A. Joint modelling of longitudinal measurements and event time data. *Biostatistics.* 2000; **1(4)**: 465-480.

Williamson PR, Kolamunnage-Dona R, Philipson P, Marson AG. Joint modelling of longitudinal and competing risks data. *Stat Med.* 2008; **27**: 6426-6438.

See Also

`lme`, `coxph`, `jointdata`, `jointplot`.

Examples

```
## Standard joint model

data(heart.valve)
heart.surv <- UniqueVariables(heart.valve,
                              var.col = c("fuyrs", "status"),
                              id.col = "num")
heart.long <- heart.valve[, c("num", "time", "log.lvmi")]
heart.cov <- UniqueVariables(heart.valve,
                             c("age", "hs", "sex"),
                             id.col = "num")
heart.valve.jd <- jointdata(longitudinal = heart.long,
                            baseline = heart.cov,
                            survival = heart.surv,
                            id.col = "num",
                            time.col = "time")
fit <- joint(data = heart.valve.jd,
             long.formula = log.lvmi ~ 1 + time + hs,
             surv.formula = Surv(fuyrs, status) ~ hs,
             model = "intslope")

## Competing risks joint model (same data as Williamson et al. 2008)

## Not run:
data(epileptic)
epileptic$interaction <- with(epileptic, time * (treat == "LTG"))
longitudinal <- epileptic[, c(1:3, 13)]
survival <- UniqueVariables(epileptic, c(4, 6), "id")
baseline <- UniqueVariables(epileptic, "treat", "id")
data <- jointdata(longitudinal = longitudinal,
                  survival = survival,
                  baseline = baseline,
                  id.col = "id",
                  time.col = "time")

fit2 <- joint(data = data,
              long.formula = dose ~ time + treat + interaction,
              surv.formula = Surv(with.time, with.status2) ~ treat,
              longsep = FALSE, survsep = FALSE,
              gpt = 3)
summary(fit2)

## End(Not run)
```

`joint.object` *Fitted* `joint` *object*

Description

An object returned by the `joint` function, inheriting from class `joint` and representing a fitted joint model for longitudinal and time-to-event (or competing risks) data.

Usage

```
joint.object
```

Format

An object of class NULL of length 0.

Value

A list with the following components.

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

- fixed longitudinal and survival sub-model fixed effects.
- random the BLUPs of the random effects.
- latent the latent association parameter(s) from the time-to-event sub-model.

sigama.z a numeric double for the residual standard error.

sigma.u the variance-covariance matrix of the random effects.

hazard a vector of the (centered) baseline hazards at each unique failure time.

log.lik the log-likelihood from the joint model fit and sub-model contributions.

numIter the number of EM algorithm iterations.

convergence a logical value of whether convergence was achieved or not.

model see joint for details.

sepassoc see joint for details.

sepests see joint for details.

compRisk a logical value indicating whether competing risks were detected or not.

sep.loglike the log-likelihood from the joint model fit (with association set to zero) and separately fitted sub-model contributions.

formulae a list of model formulae. See joint for details.

data a jointdata object. See joint for details.

call the model call. Can be used by update.

Author(s)

Graeme L. Hickey

See Also

joint.

`jointdata` *Creates an object of class* `jointdata`

Description

This function creates an object of class `jointdata`. This is an object with information on at least one of, longitudinal data or survival data. Moreover, it can also have data on baseline covariates.

Usage

```
jointdata(
  longitudinal = NA,
  survival = NA,
  baseline = NA,
  id.col = "ID",
  time.col = NA
)
```

Arguments

`longitudinal`	a data frame or matrix in the unbalanced format (one row per observation), with subject identification, time of measurements, and longitudinal measurements and/or time dependent covariates. This must be given if no `survival` argument is.
`survival`	a data frame or matrix with survival data for all the subjects. This must be given if no `longitudinal` argument is.
`baseline`	a data frame or matrix with baseline covariates, or non-time dependent covariates, for the same subjects as in `survival` and/or `longitudinal`. This has to be in the balanced format (one row per subject). By default an object of this class does not include baseline covariates.
`id.col`	an element of class `character` with the name identification of subject. This is to identify the subject identification in the data frames.
`time.col`	an element of class `character` with the time measurements identification. This is to identify the time column in the data frames.

Details

This function creates an object of class `jointdata`. This is a list with elements used in joint modelling, mainly longitudinal and/or survival data. The output has to have at least one of the data sets, longitudinal or survival. However, for joint modelling is necessary to have both data sets. Moreover, a third data frame is possible to be given as input, for the baseline (non-time dependent) covariates. The subject identification and time measurement column names are necessary.

Value

A list of length six. The first element is the vector of subjects identification. The second is, if exists a data frame of the longitudinal data. The third element of the list is, if exists a data frame of the survival data. The fourth element of the list is, if exists a data frame on the baseline covariates. The fifth is, if longitudinal data is given, the column name identification of longitudinal times. And the sixth and last element of the list is the column name identification of subjects.

Author(s)

Ines Sousa

Examples

```
data(heart.valve)
heart.surv <- UniqueVariables(heart.valve,
                              var.col = c("fuyrs", "status"),
                              id.col = "num")
heart.valve.jd <- jointdata(survival = heart.surv,
                            id.col = "num",
                            time.col = "time")
```

`jointplot` *Joint plot of longitudinal and survival data*

Description

This function views the longitudinal profile of each unit with the last longitudinal measurement prior to event-time (censored or not) taken as the end-point, referred to as time zero. In doing so, the shape of the profile prior to event-time can be inspected. This can be done over a user-specified number of time units.

Usage

```
jointplot(
  object,
  Y.col,
  Cens.col,
  lag,
  split = TRUE,
  col1,
  col2,
  xlab,
  ylab,
  gp1lab,
  gp2lab,
  smooth = 2/3,
  mean.profile = FALSE,
```

```
  mcol1,
  mcol2
)
```

Arguments

object	an object of class jointdata.
Y.col	an element of class character identifying the longitudinal response part of the jointdata object.
Cens.col	an element of class character identifying the survival status or censoring indicator part of the jointdata object.
lag	argument which specifies how many units in time we look back through. Defaults to the maximum observation time across all units.
split	logical argument which allows the profiles of units which *fail* and those which are *censored* to be viewed in separate panels of the same graph. This is the default option. Using split = FALSE will plot all profiles overlaid on a single plot.
col1	argument to choose the colour for the profiles of the *censored* units.
col2	argument to choose the colour for the profiles of the *failed* units.
xlab	an element of class character indicating the title for the x-axis.
ylab	an element of class character indicating the title for the x-axis.
gp1lab	an element of class character for the group corresponding to a censoring indicator of zero. Typically, the censored group.
gp2lab	an element of class character for the group corresponding to a censoring indicator of one. Typically, the group experiencing the event of interest.
smooth	the smoother span. This gives the proportion of points in the plot which influence the smooth at each value. Defaults to a value of 2/3. Larger values give more smoothness. See lowess for further details.
mean.profile	draw mean profiles if TRUE. Only applies to the split = TRUE case.
mcol1	argument to choose the colour for the mean profile of the units with a censoring indicator of zero.
mcol2	argument to choose the colour for the mean profile of the units with a censoring indicator of one.

Details

The function tailors the xyplot function to produce a representation of joint data with longitudinal and survival components.

Value

A lattice plot.

Note

If more than one cause of failure is present (i.e. competing risks data), then all failures are pooled together into a single failure type.

Author(s)

Pete Philipson

References

Wulfsohn MS, Tsiatis AA. A joint model for survival and longitudinal data measured with error. *Biometrics.* 1997; **53(1)**: 330-339.

See Also

xyplot, joint, jointdata.

Examples

```
data(heart.valve)
heart.surv <- UniqueVariables(heart.valve,
                              var.col = c("fuyrs", "status"),
                              id.col = "num")
heart.long <- heart.valve[, c("num", "time", "log.lvmi")]
heart.cov <- UniqueVariables(heart.valve,
                             c("age", "sex"),
                             id.col = "num")
heart.valve.jd <- jointdata(longitudinal = heart.long,
                            baseline = heart.cov,
                            survival = heart.surv,
                            id.col = "num",
                            time.col = "time")
jointplot(heart.valve.jd, Y.col = "log.lvmi",
          Cens.col = "status", lag = 5)
```

---

jointSE *Standard errors via bootstrap for a joint model fit*

Description

This function takes a model fit from a joint model and calculates standard errors, with optional confidence intervals, for the main longitudinal and survival covariates.

Usage

```
jointSE(fitted, n.boot, gpt, lgpt, max.it, tol, print.detail = FALSE)
```

Arguments

`fitted`	a list containing as components the parameter estimates obtained by fitting a joint model along with the respective formulae for the longitudinal and survival sub-models and the model chosen, see `joint` for further details.
`n.boot`	argument specifying the number of bootstrap samples to use in order to obtain the standard error estimates and confidence intervals. Note that at least `n.boot = 100` is required in order for the function to return non-zero confidence intervals.
`gpt`	the number of quadrature points across which the integration with respect to the random effects will be performed. Defaults to `gpt = 3` which produces stable estimates in most datasets.
`lgpt`	the number of quadrature points which the log-likelihood is evaluated over following a model fit. This defaults to `lgpt = 10`, though `lgpt = 3` is often sufficient.
`max.it`	the maximum number of iterations of the EM algorithm that the function will perform. Defaults to `max.it = 200`, though more iterations may be necessary for large, complex data.
`tol`	the tolerance level before convergence of the algorithm is deemed to have occurred. Default value is `tol = 0.001`.
`print.detail`	This argument determines the level of printing that is done during the bootstrapping. If `TRUE` then the parameter estimates from each bootstrap sample are output.

Details

Standard errors and confidence intervals are obtained by repeated fitting of the requisite joint model to bootstrap samples of the original longitudinal and survival data. It is rare that more than 200 bootstrap samples are needed for estimating a standard error. The number of bootstrap samples needed for accurate confidence intervals can be as large as 1000.

Value

An object of class `data.frame`.

Author(s)

Ruwanthi Kolamunnage-Dona and Pete Philipson

References

Wulfsohn MS, Tsiatis AA. A joint model for survival and longitudinal data measured with error. *Biometrics.* 1997; **53(1)**: 330-339.

Efron B, Tibshirani R. *An Introduction to the Bootstrap.* 2000; Boca Raton, FL: Chapman & Hall/CRC.

See Also

`lme`, `coxph`, `joint`, `jointdata`.

Examples

```
data(heart.valve)
heart.surv <- UniqueVariables(heart.valve,
                              var.col = c("fuyrs", "status"),
                              id.col = "num")
heart.long <- heart.valve[, c("num", "time", "log.lvmi")]
heart.cov <- UniqueVariables(heart.valve,
                             c("age", "hs", "sex"),
                             id.col = "num")
heart.valve.jd <- jointdata(longitudinal = heart.long,
                            baseline = heart.cov,
                            survival = heart.surv,
                            id.col = "num",
                            time.col = "time")
fit <- joint(heart.valve.jd,
             long.formula = log.lvmi ~ 1 + time + hs,
             surv.formula = Surv(fuyrs, status) ~ hs,
             model = "int")
jointSE(fitted = fit, n.boot = 1)
```

---

lines.jointdata *Add lines to an existing* jointdata *plot*

Description

Add lines to an existing plot of an object of class `jointdata`, for a longitudinal variable. It is possible to plot all the subjects in the data set, or just a selected `subset`. See `subset.jointdata`.

Usage

```
## S3 method for class 'jointdata'
lines(x, Y.col, ...)
```

Arguments

`x`	object of class `jointdata`.
`Y.col`	column number, or column name, of longitudinal variable to be plotted. Defaults to `Y.col = NA`, plotting all longitudinal variables.
`...`	other graphical arguments; see `plot`.

Value

A graphical device with a plot for longitudinal data.

Author(s)

Ines Sousa

See Also

Other functions are useful to be used with this such as `plot` and `points`.

Examples

```
data(heart.valve)
heart.surv <- UniqueVariables(heart.valve,
                              var.col = c("fuyrs", "status"),
                              id.col = "num")
heart.long <- heart.valve[, c(1, 4, 5, 7, 8, 9, 10, 11)]
heart.jd <- jointdata(longitudinal = heart.long,
                      survival = heart.surv,
                      id.col = "num",
                      time.col = "time")

# Randomly select a pair of subjects to plot profiles of
take <- sample(1:max(heart.jd$survival$num), 2)
heart.jd.1 <- subset(heart.jd, take[1])
heart.jd.2 <- subset(heart.jd, take[2])

plot(heart.jd.1, Y.col = 4)
lines(heart.jd.2, Y.col = 4, lty = 2)
```

---

liver *Liver cirrhosis drug trial data*

Description

This dataset gives the longitudinal observations of prothrombin index, a measure of liver function, for patients from a controlled trial into prednisone treatment of liver cirrhosis. Time-to-event information in the form of the event time and associated censoring indicator are also recorded along with a solitary baseline covariate - the allocated treatment arm in this instance. The data are taken from Andersen et al. (1993, p. 19) and were analyzed in Henderson et al. (2002). This is a subset of the full data where a number of variables were recorded both at entry and during the course of the trial.

Usage

```
data(liver)
```

Format

A `data.frame` in the unbalanced format with longitudinal observations from 488 subjects. The columns are:

`id` integer: number for patient identification.

`prothrombin` integer: prothrombin index measurement (%).

`time` numeric: time of prothrombin index measurement (years).

`treatment` integer: patient treatment indicator. Coded as `0` = placebo; `1` = prednisone.

`survival` numeric: patient survival time (years).

`cens` integer: censoring indicator. Coded as `1` = died; `0` = censored.

Source

Skrondal A, Rabe-Hesketh S. *Generalized Latent Variable Modeling: Multilevel, Longitudinal and Structural Equation Models.* Chapman & Hall/CRC. 2004. URL: http://www.gllamm.org/books/readme.html#14.6.

References

Andersen PK, Borgan O, Gill RD, Kieding N. *Statistical Models Based on Counting Processes*. New York: Springer. 2003.

Henderson R, Diggle PJ, Dobson A. Identification and efficacy of longitudinal markers for survival. *Biostatistics* 2002; **3**: 33-50.

See Also

`heart.valve`, `epileptic`, `mental`, `aids`.

---

mental *Mental health trial data*

Description

The data is obtained from a trial in which chronically ill mental health patients were randomized across two treatments: placebo and an active drug. A questionnaire instrument was used to assess each patient's mental state at weeks 0, 1, 2, 4, 6 and 8 post-randomisation, a high recorded score implying a severe condition. Some of the 100 patients dropped out of the study for reasons that were thought to be related to their mental state, and therefore potentially informative; others dropped out for reasons unrelated to their mental state.

Usage

```
data(mental)
```

Format

A balanced data set with respect to the times at which observations recorded. The data consists of the following variables on each patient:

`id` integer: patient identifier.

`Y.t0` integer: mental state assessment in week 0. Coded `NA` if missing.

`Y.t1` integer: mental state assessment in week 1. Coded `NA` if missing.

`Y.t2` integer: mental state assessment in week 2. Coded `NA` if missing.

`Y.t4` integer: mental state assessment in week 4. Coded `NA` if missing.

`Y.t6` integer: mental state assessment in week 6. Coded `NA` if missing.

`Y.t8` integer: mental state assessment in week 8. Coded `NA` if missing.

`treat` integer: treatment allocation. Coded as `0` = placebo; `1` = active drug.

`n.obs` integer: number of non-missing mental state assessments.

`surv.time` numeric: imputed dropout time in weeks. Coded as `surv.time` = `8.002` for completers.

`cens.ind` integer: censoring indicator. Coded as `0` = completer or non-informative dropout; `1` = potentially informative dropout.

Source

Peter J. Diggle (p.diggle@lancaster.ac.uk)

References

Henderson R, Diggle PJ, Dobson A. Joint modelling of longitudinal measurements and event time data. *Biostatistics.* 2000; **1(4)**: 465-480.

Diggle PJ, Farewell D, Henderson R. Longitudinal data with dropout: objectives, assumptions and a proposal (with Discussion). *Applied Statistics.* 2007; **56**: 499-550.

See Also

`heart.valve`, `liver`, `epileptic`.

`plot.jointdata` *Plot longitudinal data*

Description

Plot longitudinal data of an object of class `jointdata`, for a longitudinal variable. It is possible to plot all the subjects in the data set, or just a selected `subset`. See `subset.jointdata`.

Usage

```
## S3 method for class 'jointdata'
plot(x, Y.col, type, xlab, xlim = NULL, ylim = NULL, main = NA, pty, ...)
```

Arguments

`x`	object of class `jointdata`.
`Y.col`	column number, or column name, of longitudinal variable to be plotted. Defaults to `Y.col` = `NA`, plotting all longitudinal variables.
`type`	the type of line to be plotted, see `plot` for further details.
`xlab`	a title for the x-axis, see `title`.

xlim, ylim	numeric vectors of length 2, giving the x and y coordinates ranges, see plot.window for further details.
main	an overall title for the plot; see title.
pty	a character specifying the type of plot region to be used, see par for details.
...	other graphical arguments; see plot.

Value

A graphical device with a plot for longitudinal data.

Author(s)

Ines Sousa

See Also

lines and points.

Examples

```
data(heart.valve)
heart.surv <- UniqueVariables(heart.valve,
                              var.col = c("fuyrs", "status"),
                              id.col = "num")
heart.long <- heart.valve[, c(1, 4, 5, 7, 8, 9, 10, 11)]
heart.jd <- jointdata(longitudinal = heart.long,
                      survival = heart.surv,
                      id.col = "num",
                      time.col = "time")
plot(heart.jd, Y.col = "grad", col = "grey")
```

---

plot.vargm *Plots the empirical variogram for longitudinal data*

Description

Plots the empirical variogram for observed measurements, of an object of class vargm, obtained by using function variogram.

Usage

```
## S3 method for class 'vargm'
plot(x, smooth = FALSE, bdw = NULL, follow.time = NULL, points = TRUE, ...)
```

Arguments

x	object of class vargm obtained by using function. variogram
smooth	logical value to use a non-parametric estimator to calculate the variogram of all v_ijk. The default is FALSE, as it uses time averages.
bdw	bandwidth to use in the time averages. The default is NULL, because this is calculated automatically.
follow.time	the interval of time we want to construct the variogram for. When NULL this is the maximum of the data.
points	logical value if the points v_ijk should be plotted.
...	other graphical options as in par.

Value

A graphical device with the plot of empirical variogram.

Author(s)

Ines Sousa

Examples

```
data(mental)
mental.unbalanced <- to.unbalanced(mental, id.col = 1,
                                   times = c(0, 1, 2, 4, 6, 8),
                                   Y.col = 2:7,
                                   other.col = c(8, 10, 11))
names(mental.unbalanced)[3] <- "Y"
vgm <- variogram(indv = tail(mental.unbalanced[, 1], 30),
                 time = tail(mental.unbalanced[, 2], 30),
                 Y = tail(mental.unbalanced[, 3], 30))
plot(vgm)
```

---

points.jointdata *Add points to an existing* jointdata *plot*

Description

Add points to an existing plot of an object of class jointdata, for a longitudinal variable. It is possible plot all the subjects in the data set, or just a selected subset. See subset.jointdata.

Usage

```
## S3 method for class 'jointdata'
points(x, Y.col, ...)
```

Arguments

x	object of class jointdata.
Y.col	column number, or column name, of longitudinal variable to be plotted. Defaults to Y.col = NA, plotting all longitudinal variables.
...	other graphical arguments; see plot.

Value

A graphical device with a plot for longitudinal data. Other functions are useful to be used with this as plot and lines.

Author(s)

Ines Sousa

Examples

```
data(heart.valve)
heart.surv <- UniqueVariables(heart.valve,
                              var.col = c("fuyrs", "status"),
                              id.col = "num")
heart.long <- heart.valve[, c(1, 4, 5, 7, 8, 9, 10, 11)]
heart.jd <- jointdata(longitudinal = heart.long,
                      survival = heart.surv,
                      id.col = "num",
                      time.col = "time")

# Randomly select a pair of subjects to plot profiles of
take <- sample(1 : max(heart.jd$survival$num), 2)
heart.jd.1 <- subset(heart.jd, take[1])
heart.jd.2 <- subset(heart.jd, take[2])

plot(heart.jd.1, Y.col = "grad", type = "p")
points(heart.jd.2, Y.col = "grad", col = "blue", pch = 20)
```

---

sample.jointdata *Sample from a* jointdata *x*

Description

Generic function used to sampling a subset of data from an x of class jointdata, with a specific size of number of subjects.

Usage

```
sample.jointdata(x, size, replace = FALSE)
```

Arguments

x	an object of class jointdata.
size	number of subjects to include in the sampled subset.
replace	should sampling be with replacement? Default is replace = TRUE.

Value

An object of class jointdata, with data only on the subjects sampled.

Author(s)

Ines Sousa

See Also

sample, jointdata, UniqueVariables.

Examples

```
data(heart.valve)
heart.surv <- UniqueVariables(heart.valve,
                              var.col = c("fuyrs", "status"),
                              id.col = "num")
heart.valve.jd <- jointdata(survival = heart.surv,
                            id.col = "num",
                            time.col = "time")
sample.jointdata(heart.valve.jd, size = 10)
```

---

simjoint *Simulate data from a joint model*

Description

This function simulates longitudinal and time-to-event data from a joint model.

Usage

```
simjoint(
  n = 500,
  model = c("intslope", "int", "quad"),
  sepassoc = FALSE,
  ntms = 5,
  b1 = c(1, 1, 1, 1),
  b2 = c(1, 1),
  gamma = c(1, 0.1),
  sigu,
  vare = 0.01,
```

```
  theta0 = -3,
  theta1 = 1,
  censoring = TRUE,
  censlam = exp(-3),
  truncation = FALSE,
  trunctime = max(ntms),
  gridstep = 0.01
)
```

Arguments

n	the number of subjects to simulate data for.
model	a character string specifying the type of latent association. This defaults to the intercept and slope version as seen in Wulfsohn and Tsiatis (1997). For association via the random intercept only, choose model = "int", whereas for a quadratic association, use model = "quad". Computing times are commensurate with the type of association structure chosen.
sepassoc	logical value: if TRUE then the joint model is fitted with separate association, see Details.
ntms	the maximum number of (discrete) time points to simulate repeated longitudinal measurements at.
b1	a vector specifying the coefficients of the fixed effects in the longitudinal submodel. The order in each row is intercept, a continuous covariate, covariate, a binary covariate, the measurement time.
b2	a vector of length = 2 specifying the coefficients for the time-to-event baseline covariates, in the order of a continuous covariate and a binary covariate.
gamma	a vector of specifying the latent association parameter(s) for the longitudinal outcome. It must be of length 1 if sepassoc = FALSE.
sigu	a positive-definite matrix specifying the variance-covariance matrix. If model = "int", the matrix has dimension dim = c(1,1); if model = "intslope", the matrix has dimension dim = c(2, 2); else if model = "quad", the matrix has dimension dim = c(3, 3). If D = NULL (default), an identity matrix is assumed.
vare	a numeric value specifying the residual standard error.
theta0, theta1	parameters controlling the failure rate. See Details.
censoring	logical: if TRUE, includes an independent censoring time.
censlam	a scale (> 0) parameter for an exponential distribution used to simulate random censoring times for when censoring = TRUE.
truncation	logical: if TRUE, adds a truncation time for a maximum event time in the case of model = "int" or model = "intslope".
trunctime	a truncation time for use when truncation = TRUE. For model = "quad", trunctime is required, and defaults to max(ntms) if not specified.
gridstep	the step-size for the grid used to simulate event times when model = "quad". Default is gridstep = 0.01. See Details.

Details

The function `simjoint` simulates data from a joint model, similar to that performed in Henderson et al. (2000). It works by first simulating longitudinal data for all possible follow-up times using random draws for the multivariate Gaussian random effects and residual error terms. Data can be simulated assuming either random-intercepts only (`model = "int"`) in each of the longitudinal submodels; random-intercepts and random-slopes (`model = "intslope"`); or quadratic random effects structures (`model = "quad"`). The failure times are simulated from proportional hazards time-to-event models, using the following methodologies:

`model = "int"` The baseline hazard function is specified to be an exponential distribution with

$$\lambda_0(t) = \exp \theta_0.$$

Simulation is conditional on known time-independent effects, and the methodology of Bender et al. (2005) is used to simulate the failure time.

`model = "intslope"` The baseline hazard function is specified to be a Gompertz distribution with

$$\lambda_0(t) = \exp \theta_0 + \theta_1 t.$$

In the usual representation of the Gompertz distribution, θ_1 is the shape parameter, and the scale parameter is equivalent to $\exp(\theta_0)$. Simulation is conditional on on a predictable (linear) time-varying process, and the methodology of Austin (2012) is used to simulate the failure time.

`model="quad"` The baseline hazard function is specified as per `model="intslope"`. The integration technique used for the above two cases is complex in quadratic (and higher order) models, therefore we use a different approach. We note that hazard function can be written as

$$\lim_{dt \to 0} \lambda(t)dt = \lim_{dt \to 0} P[t \leq T \leq t + dt | T \geq t].$$

In the simulation routine the parameter `gridstep` acts as dt in that we choose a nominally small value, which multiplies the hazard and this scaled hazard is equivalent to the probability of having an event in the interval $(t, t + dt)$, or equivalently $(t, t+$`gridstep`$)$. A vector of possible times is set up for each individual, ranging from 0 to `trunctime` in increments of dt (or `gridstep`). The failure probability at each time is compared to an independent $U(0, 1)$ draw, and if the probability does not exceed the random draw then the survival time is set as `trunctime`, otherwise it is the generated time from the vector of candidate times. The minimum of these candidate times (i.e. when we deem the event to have first happened) is taken as the survival time.

Value

A list of 2 `data.frames`: one recording the requisite longitudinal outcomes data, and one recording the time-to-event data.

Author(s)

Pete Philipson

References

Austin PC. Generating survival times to simulate Cox proportional hazards models with time-varying covariates. *Stat Med.* 2012; **31(29)**: 3946-3958.

Bender R, Augustin T, Blettner M. Generating survival times to simulate Cox proportional hazards models. *Stat Med.* 2005; **24**: 1713-1723.

Henderson R, Diggle PJ, Dobson A. Joint modelling of longitudinal measurements and event time data. *Biostatistics.* 2000; **1(4)**: 465-480.

Examples

```
simjoint(10, sepassoc = TRUE)
```

`subset.jointdata` *Subsetting object of class* `jointdata`

Description

Returns an object of class `jointdata` which is a subset of an original object of class `jointdata`.

Usage

```
## S3 method for class 'jointdata'
subset(x, subj.subset, ...)
```

Arguments

`x`	an object of class `jointdata`.
`subj.subset`	vector of subject identifiers, to include in the data subset. This must be a unique vector of patient identifiers.
`...`	further arguments to be passed to or from other methods.

Value

An object of class `jointdata`, with data only on a subset of subjects.

Author(s)

Ines Sousa

Examples

```
data(heart.valve)
heart.surv <- UniqueVariables(heart.valve,
                              var.col = c("fuyrs", "status"),
                              id.col = "num")
heart.long <- heart.valve[, c(1, 4, 5, 7, 8, 9, 10, 11)]
heart.jd <- jointdata(longitudinal = heart.long,
                      survival = heart.surv,
                      id.col = "num",
                      time.col = "time")
take <- heart.jd$survival$num[heart.jd$survival$status == 0]
heart.jd.cens <- subset(heart.jd, take)
```

`summary.joint` *Summarise a random effects joint model fit*

Description

Generic function used to produce summary information from a fitted random effects joint model as represented by `object` of class `joint`.

Usage

```
## S3 method for class 'joint'
summary(object, variance = TRUE, ...)
```

Arguments

`object`	an object of class `joint`.
`variance`	should the variance components be output as variances or standard deviations? Defaults to `variance = TRUE`.
`...`	further arguments for the summary.

Value

An object inheriting from class `summary.joint` with all components included in `object` (see `joint` for a full description of the components) plus the following components:

`nobs`	the total number of (typically longitudinal) observations (i.e. rows in an unbalanced data set).
`ngrps`	the number of groups in the analyzed dataset, often individual subjects.

Author(s)

Pete Philipson

Examples

```
data(heart.valve)
heart.surv <- UniqueVariables(heart.valve,
                              var.col = c("fuyrs","status"),
                              id.col = "num")
heart.long <- heart.valve[, c("num", "time", "log.lvmi")]
heart.cov <- UniqueVariables(heart.valve,
                             c("age", "hs", "sex"),
                             id.col = "num")
heart.valve.jd <- jointdata(longitudinal = heart.long,
                            baseline = heart.cov,
                            survival = heart.surv,
                            id.col = "num",
                            time.col = "time")
fit <- joint(data = heart.valve.jd,
             long.formula = log.lvmi ~ 1 + time + hs,
             surv.formula = Surv(fuyrs,status) ~ hs,
             model = "intslope")
summary(fit)
```

`summary.jointdata` *Summarise a* `jointdata` *object*

Description

Generic function used to produce summaries of objects of class `jointdata`.

Usage

```
## S3 method for class 'jointdata'
summary(object, ...)
```

Arguments

`object`	an object of class `joint`.
`...`	further arguments for the summary.

Value

A list with five elements. Each summarises an element of the `jointdata` object:

`subjects`	Gives the number of subjects in the data set.
`longitudinal`	If longitudinal data is available, it gives the names and class, of the longitudinal variables.
`survival`	If survival data is available, it gives the number of subjects with failure and censored survival times.
`baseline`	If baseline covariates is available, it gives the names and class, of the baseline covariates.

`times` If longitudinal data is available, it gives the unique longitudinal time measurements, if it is a balanced study. In case of unbalanced study, it will only state it is an unbalanced study.

Author(s)

Ines Sousa

See Also

`jointdata`, `UniqueVariables`.

Examples

```
data(heart.valve)
heart.surv <- UniqueVariables(heart.valve,
                              var.col = c("fuyrs", "status"),
                              id.col = "num")
heart.valve.jd <- jointdata(survival = heart.surv,
                            id.col = "num",
                            time.col = "time")
summary(heart.valve.jd)
```

---

summarybal *Summary of a balanced longitudinal data set*

Description

For a balanced longitudinal data set a vector of the mean response and variances at defined time points is returned along with the correlation matrix of the responses across the time points.

Usage

```
summarybal(object, Y.col, times, use = "all.obs", na.rm, ...)
```

Arguments

`object` a longitudinal data set in the balanced format.

`Y.col` the column numbers of the longitudinal measurements at each design time point in the `object`. This does not have to be all of the longitudinal measurements taken and may be a subset instead.

`times` a vector of unique time points of the longitudinal measurements. This does not have to be all of the study time points and may be a subset instead, but should match the columns defined in `Y.col`.

`use` an optional character string giving a method for computing covariances in the presence of missing values. This must be (an abbreviation of) one of the strings `"all.obs"`, `"complete.obs"` or `"pairwise.complete.obs"`. Defaults to `use = "all.obs"`.

na.rm	logical. Should missing values be removed? By default, na.rm = FALSE.
...	further arguments for the summary.

Value

A list with three elements:

mean.vect	a matrix with the time points in the first column and the mean response vector as the second column.
variance	The vector of variances for the response at the time points.
cor.mtx	Containing the correlation matrix of the responses between each pair of time points.

Author(s)

Ines Sousa

See Also

to.balanced.

Examples

```
data(mental)
summarybal(mental, Y.col = 2:7, times = c(0, 1, 2, 4, 6, 8), na.rm = TRUE)
```

---

to.balanced *Transform data to the longitudinal balanced format*

Description

Transforms a longitudinal data set in the unbalanced format to the balanced format.

Usage

```
to.balanced(data, id.col, time.col, Y.col, other.col = NA)
```

Arguments

data	a data frame with longitudinal data in the unbalanced format. That is, in the format of 'one row per observation'.
id.col	a column number, or column name, in the data frame data, where the patient identifier is located.
time.col	a column number, or column name, in the data frame data, where the time measurements are.
Y.col	a vector of column numbers, or column names, of longitudinal variables, and/or time dependent covariates in the data frame data.

other.col — a vector of column numbers, or column names, of baseline covariates, and/or other subject level data, as for example, survival data. Default does not include other.col.

Value

A data frame with longitudinal data in the balanced format. The balanced format is considered in this context as the format where each row has data on each subject. Notice that in this format we will have multiple columns for the same longitudinal variable, each corresponding to the variable observed at each time point.

Author(s)

Ines Sousa

See Also

to.unbalanced.

Examples

```
simul <- data.frame(num = 1:10,
                    Y1.1 = rnorm(10), Y1.2 = rnorm(10),
                    Y2.1 = rnorm(10), Y2.2 = rnorm(10),
                    age = rnorm(10))
simul <- to.unbalanced(simul, id.col = 1, times = c(1, 2),
                       Y.col = 2:5, other.col = 6)
simul <- to.balanced(simul, id.col = "num", time.col = "time",
                     Y.col = c("Y1.1", "Y2.1"), other.col = "age")
```

---

to.unbalanced *Transform data to the longitudinal unbalanced format*

Description

Transforms a longitudinal data set in the balanced format to the unbalanced format.

Usage

```
to.unbalanced(data, id.col, times, Y.col, other.col = NA)
```

Arguments

data — a data frame with longitudinal data in the balanced format. That is, in the format of 'one row per subject'. data, where the patient identifications is.

id.col — a column number, or column name, in the data frame data, where the patient identifier is located.

times	a vector with the unique time points where the patients are observed. This is the study design time points in a balanced data set.
Y.col	a vector of column numbers, or column names, of longitudinal variables, and/or time dependent covariates in the data frame data.
other.col	a vector of column numbers, or column names, of baseline covariates, and/or other subject level data, as for example, survival data. Default does not include other.col.

Value

A data frame with longitudinal data in the unbalanced format. The unbalanced format is considered in this context as the format where each row has data on each subject observation.

Author(s)

Ines Sousa

See Also

to.balanced.

Examples

```
simul <- data.frame(num = 1:10,
                    Y1.1 = rnorm(10), Y1.2 = rnorm(10),
                    Y2.1 = rnorm(10), Y2.2 = rnorm(10),
                    age = rnorm(10))
to.unbalanced(simul, id.col = 1, times = c(1, 2), Y.col = 2:5,
              other.col = 6)
```

UniqueVariables *Extracts the unique non-time dependent variables per patient, from an unbalanced data set*

Description

This function extracts a set of unique variables within a patient, returning a data frame with columns, patient identification and variables selected. Each row corresponds to the data for each individual.

Usage

```
UniqueVariables(data, var.col, id.col = "ID")
```

Arguments

data	data frame, or matrix, with at least a column of patient identification and a covariate column.
var.col	vector of column names or column numbers, of the variables (non-time dependent). Cannot have mix of numbers and column names.
id.col	column name or column number of the patient identification.

Details

This function can be used, when longitudinal data is in the unbalanced format, and it is necessary, for example, to extract the set of unique baseline covariates, or any non-time dependent variables, that in the unbalanced format, are repeated for each observation row. Also, if the original data frame has survival data, this can also be used to extract the survival information from the original data set.

Value

A data frame with patient identification and covariates selected. Each row corresponds to the data for each individual. Note that, this can be only used for non-time dependent covariates. If extracting unique time dependent covariates, the function gives an error, because it can't select what is the unique covariate.

Author(s)

Ines Sousa

Examples

```
data(heart.valve)
heart.cov <- UniqueVariables(heart.valve,
                             c(2, 3, 5, 6, 12:25),
                             id.col = "num")
```

---

variogram *Empirical variogram for longitudinal data*

Description

Calculates the variogram for observed measurements, with two components, the total variability in the data, and the variogram for all time lags in all individuals.

Usage

```
variogram(indv, time, Y)
```

Arguments

indv	vector of individual identification, as in the longitudinal data, repeated for each time point.
time	vector of observation time, as in the longitudinal data.
Y	vector of observed measurements. This can be a vector of longitudinal data, or residuals after fitting a model for the mean response.

Details

The empirical variogram in this function is calculated from observed half-squared-differences between pairs of measurements, $v_ijk = 0.5 * (r_ij - r_ik)^2$ and the corresponding time differences $u_ijk = t_ij - t_ik$. The variogram is plotted for averages of each time lag for the v_ijk for all i.

Value

An object of class `vargm` and `list` with two elements. The first `svar` is a matrix with columns for all values (u_ijk, v_ijk), and the second `sigma2` is the total variability in the data.

Note

There is a function `plot.vargm` which should be used to plot the empirical variogram.

Author(s)

Ines Sousa

Examples

```
data(mental)
mental.unbalanced <- to.unbalanced(mental, id.col = 1,
                                   times = c(0, 1, 2, 4, 6, 8),
                                   Y.col = 2:7,
                                   other.col = c(8, 10, 11))
names(mental.unbalanced)[3] <- "Y"

vgm <- variogram(indv = tail(mental.unbalanced[, 1], 30),
                 time = tail(mental.unbalanced[, 2], 30),
                 Y = tail(mental.unbalanced[, 3], 30))
```

Index

∗ **aplot**
  lines.jointdata, 18
  points.jointdata, 23
∗ **datagen**
  simjoint, 25
∗ **datasets**
  aids, 3
  epileptic, 4
  heart.valve, 5
  liver, 19
  mental, 20
∗ **data**
  jointdata, 13
  sample.jointdata, 24
  subset.jointdata, 28
  to.balanced, 32
  to.unbalanced, 33
  UniqueVariables, 34
∗ **dplot**
  jointplot, 14
  plot.jointdata, 21
  plot.vargm, 22
∗ **htest**
  jointSE, 16
∗ **list**
  jointdata, 13
∗ **manip**
  to.balanced, 32
  to.unbalanced, 33
  UniqueVariables, 34
∗ **methods**
  summary.joint, 29
  summary.jointdata, 30
  summarybal, 31
∗ **models**
  joint, 7
  jointSE, 16
  variogram, 35
∗ **smooth**
  variogram, 35
∗ **survival**
  joint, 7
  joint.object, 11
  jointdata, 13
  jointSE, 16
  simjoint, 25

aids, 3, *5, 6*, *20*

coxph, *8–10*, *17*

epileptic, *3*, 4, *6*, *20, 21*

heart.valve, *3*, *5*, 5, *20, 21*

joineR, 6
joint, 7, *12*, *16, 17*, *29*
joint.object, 11
jointdata, *10*, *12*, 13, *16, 17*, *25*, *31*
jointplot, *10*, 14
jointSE, 16

lines, *22*, *24*
lines.jointdata, 18
liver, *3*, *5, 6*, 19, *21*
lme, *8–10*, *17*
lowess, *15*

mental, *3*, *5, 6*, *20*, 20

par, *22, 23*
plot, *18, 19*, *21, 22*, *24*
plot.jointdata, 21
plot.vargm, 22, *36*
plot.window, *22*
points, *19*, *22*
points.jointdata, 23

sample, *25*
sample.jointdata, 24

simjoint, 25
subset.jointdata, *18*, *21*, *23*, 28
summary.joint, 29
summary.jointdata, 30
summarybal, 31
Surv, *8*

title, *21, 22*
to.balanced, *32*, 32, *34*
to.unbalanced, *33*, 33

UniqueVariables, *25*, *31*, 34
update, *12*

variogram, *22, 23*, 35

xyplot, *15, 16*