Interval censoring with R

Version 0.1
7 September 2006

Martyn Plummer International Agency for Research on Cancer
plummer@iarc.fr

Bendix Carstensen Steno Diabetes Center, Gentofte, Denmark
bxc@steno.dk
http://www.biostat.ku.dk/~bxc

Contents

1 Introduction

Interval censored data have been around for ages. Analytical procedures have almost all relied on parametric models because the likelihood contribution for an observation of a disease-free interval (e, w) and an interval (w, d) from last date seen well, w, to first day seen diseased d, with an underlying cumulative hazard function Λ is:

$$\exp\bigl(-(\Lambda(w) - \Lambda(e))\bigr)\Bigl(1 - \exp\bigl(-(\Lambda(d) - \Lambda(w))\bigr)\Bigr)$$

which with suitable parametric assumptions for the intensity gives a tractable expression to use for likelihood maximization.

Becker [1] observed that for panel data where all persons are seen a fixed set of dates (where some may miss an appointment), the likelihood reduces to a binomial model with log-link.

Carstensen [2] and Farrington [3] provided in two very similar papers a recipe for how to get things implemented in the at that time lingua franca of practical computing, GLIM. However this has to our knowledge never been disseminated in any practical form to the biostatistical community. The consequence has been that no one has used these methods in routine analysis of the abundant interval-censored data.

The purpose of this note is to describe an implementation of the methods in R with some worked examples that shows how data should be set up and how results are output from R and how they should be reported in practical analyses.

2 Data

To describe interval censored observations three dates are needed:

1. Date first seen well.

2. Date last seen well.

3. Date first seen ill.

This means that we infer that no event has taken place between the first two events.

In practical coding of data we accept that either of the two first ones are missing, in which case we set them both to the same value. If the date first seen ill is missing, we have a person which has experience no event.

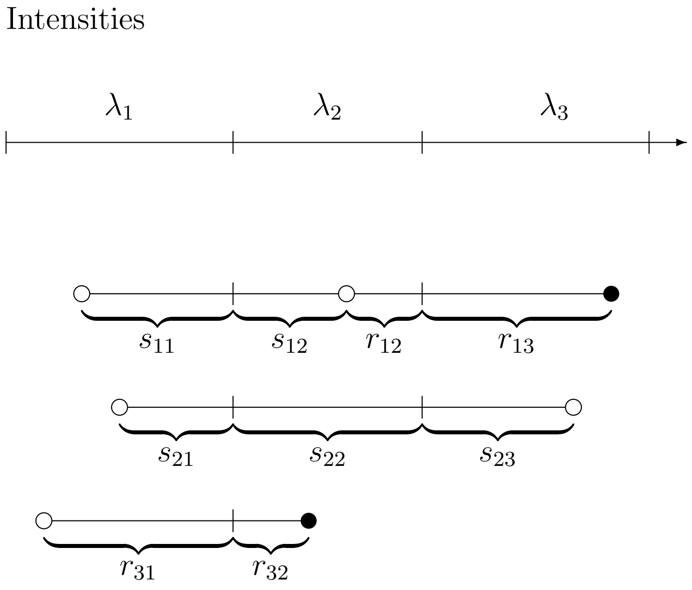
For each person there can be covariates attached too.

These are the data needed for the analyses described below.

3 Algorithms

3.1 One timescale, no covariates

Consider a person, p, where we know date of entry, t_{pe} (the first date seen without event), a last date known without the event, t_{pw} (which is the censoring date for those not known to experience the event), and for people for whom the event has occurred, an earliest date known to be after the event, t_{pd}.



Variables

p	y_{pj}	x_{1pj}	x_{2pj}	x_{3pj}
1	1	$-s_{11}$	$-s_{12}$	0
1	0	0	$-r_{12}$	$-r_{13}$
2	1	$-s_{21}$	$-s_{22}$	$-s_{23}$
3	0	$-r_{31}$	$-r_{32}$	0

Figure 1: *Construction of covariates x_{ipj} for the piecewise constant intensity model. Circles represent the last date seen before the 'event', dots the first date seen after.*

If the rate is constant equal to λ the likelihood contribution for this person is:

$$\begin{aligned} & \Pr(\text{no event from } t_{pe} \text{ to } t_{pw}\ ) \\ \times\ & \{1 - \Pr(\text{no event from } t_{pw} \text{ to } t_{pd}\ |\text{no event till } t_{pw})\} \\ =\ & \exp\Big(-\lambda(t_{pw} - t_{pe})\Big) \times \Big\{1 - \exp\Big(-\lambda(t_{pd} - t_{pw})\Big)\Big\} \end{aligned} \tag{1}$$

This can easily be recognized as the likelihood from two independent Bernoulli observations, 1 and 0, with means $\mathrm{e}^{-\lambda(t_{pw}-t_{pe})}$ and $\mathrm{e}^{-\lambda(t_{pd}-t_{pw})}$, respectively, i.e. the likelihood from a generalized linear model with binomial error and logarithmic link function, so this case easily solved by software that fits this kind of model.

Now suppose the time scale is divided into intervals $I_1 = (t_0, t_1), I_2 = (t_1, t_2), \ldots$, and put $\ell_i = t_i - t_{i-1}$. Assume that the intensity of events is constant within each I_i, equal to λ_i, say. The dates t_{pe}, t_{pw}, t_{pd} need not be in the set $\{t_0, t_1, \ldots\}$.

Then let s_{pi} be the time in I_i during which it is known that person p has not experienced an event (that is $I_i \cap (t_{pe}, t_{pw})$) and let r_{pi} be the time in I_i during which an event may have occurred (that is $I_i \cap (t_{pw}, t_{pd})$), (see figure 1). In most cases s_{pi} and r_{pi} are either 0 or ℓ_i, and $s_{pi} + r_{pi} \leq \ell_i$.

Under this model the contribution to the likelihood from person p will be:

$$\begin{aligned} & \Pr(\text{no event from } t_{pe} \text{ to } t_{pw}\ ) \\ \times\ & \{1 - \Pr(\text{no event from } t_{pw} \text{ to } t_{pd}\ |\text{no event till } t_{pw})\} \\ =\ & \exp\Big(-\sum_i \lambda_i s_{pi}\Big) \times \Big\{1 - \exp\Big(-\sum_i \lambda_i r_{pi}\Big)\Big\} \end{aligned} \tag{2}$$

Again we see that we have a likelihood corresponding to two independent Bernoulli trials with outcome 1 and 0, and that the model is a generalised liner model with binomial error and logarithmic link.

3.1.1 Keeping the λs positive

However the log-link with the binomial family is notoriously unstable, and moreover there is nothing in the setup above that keeps the λs from going negative. That can however be fixed by noting that:

$$\exp\Big(-\sum_i \lambda_i s_{pi}\Big) = \prod_i \exp\Big(-\lambda_i s_{pi}\Big)$$

which in coding terms mean replacing the single 1-observation with linear predictor $-\sum_i \lambda_i s_{pi}$ by several 1-observations with the different linear predictors $\lambda_i s_{pi}$.

3.1.2 `expand.data`

This data-expansion needed for this is handled by the function `expand.data`, which is called by all the fitting functions, so the user need not bother about this.

`expand.data` does the data-expansion based on three times for each person: `first.well`, `last.well` and `first.ill`. If any of the two first is missing (`NA`), that is set to value of the other; corresponding to person 3 in figure 1. If `first.ill` is missing, no event has occurred; corresponding to person 2 in figure 1. Furthermore `expand.data` need a specification of the breakpoints between the intervals where the underlying hazard is assumed constant.

The output from `expand.data` will be two data-frames, `rates.frame` with the covariates needed to estimate the underlying rates (the λs), and `cov.frame` with the columns needed to fit the model specified the `formula` argument.

3.1.3 Sensible starting values

The model that is being fitted is a model with a link that does not guarantee a probability in the range $(0, 1)$, so starting values are needed. In order to get sensible starting values, it is recommended to compute an overall estimate of the event rate as the total number of events divided by the total follow-up-time, and use this as starting value for all the baseline rates in the model.

3.2 `fit.add`

This is what is implemented in the R-function `fit.add`, which requires input of the intervals where the underlying rates are constant, given as interval endpoints in the vector `breaks`, in order to pass this on the `expand.data`.

`fit.add` fits the baseline rates in the prespecified intervals. These are put in the same function because the actual model-matrix used in the fitting depends on the specification of the intervals where the baseline rate is assumed constant.

The function also optionally fits and additive excess risk model, i.e. a model where we assume that covariates act additively on the rate-scale.

The contributions to the linear predictor will in this case not be

$$\sum_i \lambda_i x_{pi} \qquad \text{but} \qquad \sum_i \Big(\lambda_i + \sum_k \beta_k z_{pk}\Big) x_{pi} = \sum_i \lambda_i x_{pi} + \sum_k \beta_k \Big(z_{pk} \sum_i x_{pi}\Big)$$

which means that the covariates in the binomial model with log-link should be x_{pi} (corresponding to the λs) and $z_{pk} \sum_i x_{pi}$ (corresponding to the βs), the latter ones giving the excess rate estimates associated with the covariates.

3.2.1 `fit.mult`

This function fits the multiplicative relative risk model.

The likelihood for the data under the multiplicative relative risk model (Cox model, proportional hazards model) is constructed by replacing the terms λ_i in the likelihood for the simple case above by the terms $\lambda_i \exp(\sum_k \beta_k z_k)$, which leads to a Bernoulli likelihood for independent observations with success probability (mean):

$$\begin{aligned} \mu_p(\mathbf{z}) &= \exp\Big\{\textstyle\sum_i \lambda_i x_{ip} \exp(\sum_k \beta_k z_{kp})\Big\} \\ &= \exp\Big\{-\exp(\ln[-\textstyle\sum_i \lambda_i x_{ip}] + \sum_k \beta_k z_{kp})\Big\} \end{aligned} \tag{3}$$

For fixed β's this is a generalized linear model; the parameters are the λ's and the covariates $x_i \exp(\sum_k \beta_k z_k)$, the error-distribution is Bernoulli and the link is logarithmic. For fixed λ's it is also a generalised linear model; the parameters are the β's, the covariates z_k and the error-distribution Bernoulli, the link log−log and the offset $\ln(-\sum_i \lambda_i x_i)$.

This suggests the following fitting algorithm:

1. Fit a model as given in section 2, to obtain initial estimates of the λ's.
2. Fix the λ's, and fit a model with covariates z_k, log−log-link and offset $\ln(-\sum_i \lambda_i x_i)$ to obtain estimates of the β's.
3. Fix the β's, form the covariates $x_i \exp(\sum_k \beta_k z_k)$, and fit a model with these covariates and log-link.
4. Repeat 2. and 3. until convergence.

If this algorithm converges to a point, it will be a stationary point of the likelihood function.

3.3 Confidence intervals for estimates

The additive excess risk model has the rates as parameters, and therefore symmetric confidence intervals based on estimated standard errors form the model fit are likely to be wrong, unless inference is based on massive amounts of data.

For the multiplicative rate ratio model where one set of parameters is estimated while another set is held fixed, we will get variances of the parameter estimates in one of the sets (the β's or the λ's) that are estimates of the *conditional* variances given the value of the maximum likelihood estimates of the parameters in the other set, and as such underestimate the marginal variances, since conditional variances are never greater than marginal variances.

Therefore we have provided bootstrapped confidence intervals for the parameters in both models. This is done by resampling the persons in the original dataframe, expanding it and refitting the model 1000 times.

4 EM-algorithm for interval censored data

The EM-algorithm for interval censored data just consists in imputing the interval censored event times with the expected event times under the current estimates. This is because the log-likelihood for a piecewise constant hazard ($D\log(\lambda) - \lambda Y$) is linear in the follow-up time.

For an interval censored time (u, s) the expected time of event given a hazard function $\lambda(t)$ should be based on the density function, conditional on being in the interval.

The density is:

$$\lambda(t)\exp(-\Lambda(t))$$

so the conditional density given observation in the interval (u, s) is

$$\frac{\lambda(t)\exp(-(\Lambda(t)-\Lambda(u)))}{\exp(-\Lambda(u))-\exp(-\Lambda(s))}$$

4.1 Calculation of the expected event time

The idea is to lay out a number of equally spaced time-points over the interval (u, s) and then compute the average of these timepoints weighted by the conditional density evaluated at these points. Since it is a weighted average, the normalizing factor is irrelevant, so the weights should just be $\lambda(t)\exp(-(\Lambda(t)-\Lambda(u)))$.

This is implemented in the function `E.time`, which accepts either an intensity or a cumulative intensity supplied either as a function or a table of times and corresponding values.

And this is as far as the implementation of the EM-machinery has come. Although the function accepts vectors of entry and exit times, the practical applicability of this feature is going to be limited, since in practise the intensity function will be different for each person in a study.

The code is as follows:

```
E.time <-
function( e, x, lambda=NULL, Lambda=NULL, N=100 )
{
# Estimates the expected event time given that the event has taken
# place between e and x, and that the intensity is lambda.
# lambda or Lambda can be given either as a function or as a two-colum
# structure of times and corresponding values.
if( length( e ) != length( x ) ) stop( "e and x must have same length!" )
if( N < 2 ) stop( "N must be at least 2!" )
if( !is.null( lambda ) )
  {
  if( !is.function( lambda ) )
    {
    if( dim( lambda )[2] !=2 ) stop( "lambda must be a 2-column structure" )
    lhelp <- approxfun( lambda )
    } else lhelp <- lambda
  # The intensity at midpoints
  lamb <- function( tt ) lhelp( tt[-1] - diff( tt )/ 2 )
  # The cumulative intensity at midpoints:
  # The first point is approxomating the integral over the first
  # half of the first interval
  Lamb <- function( tt ) cumsum( c( lhelp( (3*tt[1]+tt[2])/4 )/2,
                                    lhelp( tt[-c(1,length(tt))] ) ) )
  } else
if( !is.null( Lambda ) )
  {
  if( !is.function( Lambda ) )
    {
    if( dim( Lambda )[2] !=2 ) stop( "Lambda must be a 2-column structure" )
    Lhelp <- function( tt ) approxfun( Lambda )( tt ) -
                            approxfun( Lambda )( tt[1] )
    } else Lhelp <- function( tt ) Lambda( tt ) - Lambda( tt[1] )
  lamb <- function( tt ) diff( Lhelp( tt ) ) / diff( tt )
  Lamb <- function( tt ) Lhelp( tt[-1] - diff( tt )/2 )
  } else
stop( "Either lambda or Lambda must be supplied!" )
# Here is the real business:
# Expected time between two single times
scalar.xt <- function( io )
  {
  tt <- seq( from=io[1], to=io[2], length=N )
  ll <- lamb( tt )
  LL <- Lamb( tt )
  ff <- ll * exp( -LL )
  sum( (tt[-1]-diff(tt))*ff ) / sum( ff )
  }
# - and use that on the matrix
apply( cbind( e, x ), 1, scalar.xt )
}
```

References

[1] NG Becker and M Melbye. Use of a log-linear model to compute the empirical survival curve from interval-censored data, with application to data on tests for hiv positivity. *Australian Journal of Statistics*, 33(2):125–133, 1991.

[2] B Carstensen. Regression models for interval censored survival data: application to HIV infection in Danish homosexual men. *Stat Med*, 15(20):2177–2189, Oct 1996.

[3] CP Farrington. Interval censored survival data: a generalized linear modelling approach. *Statistics in Medicine*, 15(3):283–292, Feb 1996.

5 The IC-package (so far)

E.time *Calculates the expected event time for an interval censored observation*

Description

If last time without event (e) and first tiame after event (x) and a representation of the hazard or integrated hazard are given the function computes the conditional expectation of the event time given that the event has taken place between e and x.

Usage

```
 E.time(e, x, lambda = NULL, Lambda = NULL, N = 100)
```

Arguments

e	Numerical vector. Last time without event.
x	Numerical vector. First time after event.
lambda	The hazard function. Can either be a 2-coumn matrix of times and hazrds, or a function with one argument that returns the hazard a given time.
Lambda	do., but for the integrated hazard
N	The number of points used for the calculation of the density

Details

The function generates N equally spaced points between e and x, and computes the intensity lambda and the integrated intensity Lambda at the mipoints between these points. The conditional probability density lambda * exp(-Lambda) is then evaluated at these points, and used as weights in calculating the weighted avarage of the timepoints as the expected time of event. The expression lambda * exp(- Lambda) is not a proper density, but this is immaterial as it is only used as weights.

Value

A vector of the same lenght as e.

Author(s)

Bendix Carstensen, Steno Diabetes Center, ⟨bxc@steno.dk⟩

Examples

```
library( survival )
data( lung )

m0 <- coxph( Surv( time, status==2 ) ~ 1, data=lung )
# Here is the baseline integrated hazard from the Cox model
Bh <- basehaz( m0 )[,2:1]

# Intervals where event took place
ni <- 20
e <- runif( ni, 50, 500 )
x <- e + runif( ni, 50, 200 )

# Show the results and compare with the midpoints of the intervals
cbind( last.well=e, first.ill=x,
       expected=E.time( e, x, Lambda=Bh ),
       midpoint=(e+x)/2 )
```

`expand.data` *Function to expand data for regression analysis of interval censored data.*

Description

This is a utility function.

The original records with `first.well`, `last.well` and `first.ill` are expanded to multiple records; several for each interval where the person is known to be well and one where the person is known to fail. At the same time columns for the covariates needed to estimate rates and the response variable are generated.

Usage

```
expand.data(fu, formula, breaks, data)
```

Arguments

`fu` A 3-column matrix with `first.well`, `last.well` and `first.ill` in each row.

`formula` Model fromula, used to derive the model matrix.

`breaks` Defines the intervals in which the baseline rate is assumed constant. All follow-up before the first and after the last break is discarded.

`data` Datafrem in which `fu` and `formula` is interpreted.

Value

Returns a list with three components

`rates.frame` Dataframe of covariates for estimation of the baseline rates — one per interval defined by `breaks`.

`cov.frame` Dataframe for estimation of the covariate effects. A data-framed version of the designmatrix from `formula`.

`y` Response vector.

Author(s)

Martyn Plummer, ⟨plummer@iarc.fr⟩

References

B Carstensen: Regression models for interval censored survival data: application to HIV infection in Danish homosexual men. Statistics in Medicine, 15(20):2177-2189, 1996.

See Also

`Icens` `fit.mult` `fit.add`

Examples

`fit.add` *Fit an addive excess risk model to interval censored data.*

Description

Utility function.

The model fitted assumes a piecewise constant inensity for the baseline, and that the covariates act additively on the rate scale.

Usage

```
fit.add( y, rates.frame, cov.frame, start )
```

Arguments

`y` Binary vector of outcomes

`rates.frame` Dataframe expanded from the original data by `expand.data`, cooresponding to covariates for the rate parameters.

`cov.frame` do., but covariates corresponding to the `formula` argument of `Icens`

`start` Starting values for the rate parameters. If not supplied, then starting values are generated.

Value

A `glm` object from a binomial model with log-link function.

Author(s)

Martyn Plummer, ⟨plummer@iarc.fr⟩

References

B Carstensen: Regression models for interval censored survival data: application to HIV infection in Danish homosexual men. Statistics in Medicine, 15(20):2177-2189, 1996.

CP Farrington: Interval censored survival data: a generalized linear modelling approach. Statistics in Medicine, 15(3):283-292, 1996.

See Also

`Icens` `fit.mult`

Examples

```
data( HIV.dk )
```

`fit.baseline` *Fit a piecewise contsnt intesity model for interval censored data.*

Description

Utility function

Fits a binomial model with logaritmic link, with `y` as outcome and covariates in `rates.frame` to estimate rates in the inttervals between `breaks`.

Usage

```
fit.baseline( y, rates.frame, start )
```

Arguments

`y`	Binary vector of outcomes
`rates.frame`	Dataframe expanded from the original data by `expand.data`
`start`	Starting values for the rate parameters. If not supplied, then starting values are generated.

Value

A `glm` object, with binomial error and logaritmic link.

Author(s)

Martyn Plummer, ⟨plummer@iarc.fr⟩

References

put references to the literature/web site here

See Also

`fit.add` `fit.mult`

Examples

`fit.mult` *Fits a multiplicative relative risk model to interval censored data.*

Description

Utility function.

The model fitted assumes a piecewise constant baseline rate in intervals specified by the argument `breaks`, and a multiplicative relative risk function.

Usage

```
fit.mult( y, rates.frame, cov.frame, start )
```

Arguments

`y`	Binary vector of outcomes
`rates.frame`	Dataframe expanded from the original data by `expand.data`, cooresponding to covariates for the rate parameters.
`cov.frame`	do., but covariates corresponding to the `formula` argument of `Icens`
`start`	Starting values for the rate parameters. If not supplied, then starting values are generated.

Details

The model is fitted by alternating between two generalized linear models where one estimates the underlying rates in the intervals, and the other estimates the log-relative risks.

Value

A list with three components:

`rates` A glm object from a binomial model with log-link, estimating the baseline rates.

`cov` A glm object from a binomial model with complementary log-log link, estimating the log-rate-ratios

`niter` Nuber of iterations, a scalar

Author(s)

Martyn Plummer, ⟨plummer@iarc.fr⟩, Bendix Carstensen, ⟨bxc@steno.dk⟩

References

B Carstensen: Regression models for interval censored survival data: application to HIV infection in Danish homosexual men. Statistics in Medicine, 15(20):2177-2189, 1996.

CP Farrington: Interval censored survival data: a generalized linear modelling approach. Statistics in Medicine, 15(3):283-292, 1996.

See Also

`Icens` `fit.add`

Examples

```
data( HIV.dk )
```

`hivDK` *hivDK: seroconversion in a cohort of Danish men*

Description

Data from a survey of HIV-positivity of a cohort of Danish men followed by regular tests from 1983 to 1989.

Usage

```
data(hivDK)
```

Format

A data frame with 297 observations on the following 7 variables.

`id` ID of the person

`entry` Date of entry to the study. Date variable.

`well` Date last seen seronegative. Date variable.

`ill` Date first seen seroconverted. Date variable.

`bth` Year of birth minus 1950.

`pyr` Annual number of sexual partners.

`us` Indicator of wheter the person has visited the USA.

Source

Mads Melbye, Statens Seruminstitut.

References

Becker N.G. and Melbye M.: Use of a log-linear model to compute the empirical survival curve from interval-censored data, with application to data on tests for HIV-positivity, Australian Journal of Statistics, 33, 125–133, 1990.

Melbye M., Biggar R.J., Ebbesen P., Sarngadharan M.G., Weiss S.H., Gallo R.C. and Blattner W.A.: Seroepidemiology of HTLV-III antibody in Danish homosexual men: prevalence, transmission and disease outcome. British Medical Journal, 289, 573–575, 1984.

Examples

```
data(hivDK)
str(hivDK)
```

`Icens` *Fits a regression model to interval censored data.*

Description

The models fitted assumes a piecewise constant baseline rate in intervals specified by the argument `breaks`, and for the covariates either a multiplicative relative risk function (default) or an additive excess risk function.

Usage

```
Icens( first.well, last.well, first.ill,
       formula, model.type=c("MRR","AER"), breaks,
       boot=FALSE, alpha=0.05, keep.sample=FALSE,
       data )
```

Arguments

`first.well` Time of entry to the study, i.e. the time first seen without event. Numerical vector.

`last.well` Time last seen without event. Numerical vector.

`first.ill` Time first seen with event. Numerical vector.

`formula` Model formula for the log relative risk.

`model.type` Which model should be fitted.

`breaks` Breakpoints between intervals in which the underlying timescale is assumed constant. Any observation outside the range of `breaks` is discarded.

`boot` Should bootstrap be performed to produce confidence intervals for parameters. If a number is given this will be the number of bootsrap samples. The default is 1000.

`alpha` 1 minus the confidence level.

`keep.sample` Should the bootstrap sample of the parameter values be returned?

`data` Data frame in which the times and formula are interpreted.

Details

The model is fitted by calling either `fit.mult` or `fit.add`.

Value

An object of class `"Icens"`: a list with three components:

`rates` A glm object from a binomial model with log-link, estimating the baseline rates, and the excess risk if `"AER"` is specfied.

`cov` A glm object from a binomial model with complementary log-log link, estimating the log-rate-ratios. Only if `"MRR"` is specfied.

`niter` Nuber of iterations, a scalar

`boot.ci` If `boot=TRUE`, a 3-column matrix with estimates and 1-`alpha` confidence intervals for the parameters in the model.

`sample` A matrix of the parameterestimates from the bootstrapping. Rows refer to parameters, columns to bootstrap samples.

Author(s)

Martyn Plummer, ⟨plummer@iarc.fr⟩, Bendix Carstensen, ⟨bxc@steno.dk⟩

References

B Carstensen: Regression models for interval censored survival data: application to HIV infection in Danish homosexual men. Statistics in Medicine, 15(20):2177-2189, 1996.

CP Farrington: Interval censored survival data: a generalized linear modelling approach. Statistics in Medicine, 15(3):283-292, 1996.

See Also

`fit.add` `fit.mult`

Examples

```
data( HIV.dk )
```

`plotevent` *Plot Equivalence Classes*

Description

For interval censored data, segments of times between last.well and first.ill are plotted for each conversion in the data. It also plots the equivalence classes.

Usage

```
plotevent(last.well, first.ill, data)
```

Arguments

`last.well`	Time at which the individuals are last seen negative for the event
`first.ill`	Time at which the individuals are first seen positive for the event
`data`	Data with a transversal shape

Details

last.well and first.ill should be written as character in the function.

Value

Graph

Author(s)

Delphine Maucort-Boulch, Bendix Carstensen, Martyn Plummer

References

Carstensen B. Regression models for interval censored survival data: application to HIV infection in Danish homosexual men.Stat Med. 1996 Oct 30;15(20):2177-89.

Lindsey JC, Ryan LM. Tutorial in biostatistics methods for interval-censored data.Stat Med. 1998 Jan 30;17(2):219-38.

See Also

`Icens`