Package 'simrec'

November 20, 2020

Title Simulation of Recurrent Event Data for Non-Constant Baseline Hazard

Version 1.0.0

Date 2020-11-10

Description Simulation of recurrent event data for non-constant baseline hazard in the total time model with risk-free intervals and possibly a competing event. Possibility to cut the data to an interim data set. Data can be plotted. Details about the method can be found in Jahn-Eimermacher, A. et al. (2015) <doi:10.1186/s12874-015-0005-2>.

Depends R (>= 2.10)

Imports graphics, stats

VignetteBuilder knitr

Suggests knitr, rmarkdown, DT

URL https://github.com/federicomarini/simrec

License GPL-2

LazyData true

RoxygenNote 7.1.1

Encoding UTF-8

NeedsCompilation no

Author Katharina Ingel [aut],
Antje Jahn-Eimermacher [aut],
Stella Preussler [aut],
Federico Marini [aut, cre] (<https://orcid.org/0000-0003-3252-7758>)

Maintainer Federico Marini <marinif@uni-mainz.de>

Repository CRAN

Date/Publication 2020-11-20 09:10:02 UTC

R topics documented:

`simrec` *simrec*

Description

This function allows simulation of recurrent event data following the multiplicative intensity model described in Andersen and Gill [1] with the baseline hazard being a function of the total/calendar time. To induce between-subject-heterogeneity a random effect covariate (frailty term) can be incorporated. Data for individual i are generated according to the intensity process

$$Y_i(t) * \lambda_0(t) * Z_i * exp(\beta^t X_i),$$

where X_i defines the covariate vector and β the regression coefficient vector. $\lambda_0(t)$ denotes the baseline hazard, being a function of the total/calendar time t, and $Y_i(t)$ the predictable process that equals one as long as individual i is under observation and at risk for experiencing events. Z_i denotes the frailty variable with $(Z_i)_i$ iid with $E(Z_i) = 1$ and $Var(Z_i) = \theta$. The parameter θ describes the degree of between-subject-heterogeneity. Data output is in the counting process format.

Usage

```
simrec(
  N,
  fu.min,
  fu.max,
  cens.prob = 0,
  dist.x = "binomial",
  par.x = 0,
  beta.x = 0,
  dist.z = "gamma",
  par.z = 0,
  dist.rec,
  par.rec,
  pfree = 0,
  dfree = 0
)
```

Arguments

N	Number of individuals
fu.min	Minimum length of follow-up.
fu.max	Maximum length of follow-up. Individuals length of follow-up is generated from a uniform distribution on [fu.min,fu.max]. If fu.min=fu.max, then all individuals have a common follow-up.
cens.prob	Gives the probability of being censored due to loss to follow-up before fu.max. For a random set of individuals defined by a B(N,cens.prob)-distribution, the time to censoring is generated from a uniform distribution on [0,fu.max]. Default is cens.prob=0, i.e. no censoring due to loss to follow-up.
dist.x	Distribution of the covariate(s) X. If there is more than one covariate, dist.x must be a vector of distributions with one entry for each covariate. Possible values are "binomial" and "normal", default is dist.x="binomial".
par.x	Parameters of the covariate distribution(s). For "binomial",par.x is the probability for $x = 1$. For "normal", par.x=c(μ, σ) where μ is the mean and σ is the standard deviation of a normal distribution. If one of the covariates is defined to be normally distributed, par.x must be a list, e.g. dist.x <-c("binomial","normal") and par.x <-list(0.5,c(1,2)). Default is par.x=0, i.e. $x = 0$ for all individuals.
beta.x	Regression coefficient(s) for the covariate(s) x. If there is more than one covariate, beta.x must be a vector of coefficients with one entry for each covariate. simrec generates as many covariates as there are entries in beta.x. Default is beta.x=0, corresponding to no effect of the covariate x.
dist.z	Distribution of the frailty variable Z with $E(Z) = 1$ and $Var(Z) = \theta$. Possible values are "gamma" for a Gamma distributed frailty and "lognormal" for a lognormal distributed frailty. Default is dist.z="gamma".
par.z	Parameter θ for the frailty distribution: this parameter gives the variance of the frailty variable Z. Default is par.z=0, which causes $Z = 1$, i.e. no frailty effect.
dist.rec	Form of the baseline hazard function. Possible values are "weibull" or "gompertz" or "lognormal" or "step".
par.rec	Parameters for the distribution of the event data. If dist.rec="weibull" the hazard function is $$\lambda_0(t) = \lambda * \nu * t^{\nu-1},$$ where $\lambda > 0$ is the scale and $\nu > 0$ is the shape parameter. Then par.rec=c(λ, ν). A special case of this is the exponential distribution for $\nu = 1$.\ If dist.rec="gompertz", the hazard function is $$\lambda_0(t) = \lambda * exp(\alpha t),$$ where $\lambda > 0$ is the scale and $\alpha \in (-\infty, +\infty)$ is the shape parameter. Then par.rec=c(λ, α).\ If dist.rec="lognormal", the hazard function is $$\lambda_0(t) = [(1/(\sigma t)) * \phi((ln(t) - \mu)/\sigma)]/[\Phi((-ln(t) - \mu)/\sigma)],$$ where ϕ is the probability density function and Φ is the cumulative distribution function of the standard normal distribution, $\mu \in (-\infty, +\infty)$ is a location parameter and $\sigma > 0$ is a shape parameter. Then par.rec=c(μ, σ). Please note,

that specifying `dist.rec="lognormal"` together with some covariates does not specify the usual lognormal model (with covariates specified as effects on the parameters of the lognormal distribution resulting in non-proportional hazards), but only defines the baseline hazard and incorporates covariate effects using the proportional hazard assumption.\ If `dist.rec="step"` the hazard function is

$$\lambda_0(t) = a, t <= t_1, and \lambda_0(t) = b, t > t_1$$

. Then `par.rec=c(`a, b, t_1`)`.

`pfree` Probability that after experiencing an event the individual is not at risk for experiencing further events for a length of `dfree` time units. Default is `pfree=0`.

`dfree` Length of the risk-free interval. Must be in the same time unit as `fu.max`. Default is `dfree=0`, i.e. the individual is continously at risk for experiencing events until end of follow-up.

Details

Data are simulated by extending the methods proposed by Bender et al [2] to the multiplicative intensity model.

Value

The output is a data.frame consisting of the columns:

`id` An integer number for identification of each individual

`x` or `x.V1,x.V2,...` - depending on the covariate matrix. Contains the randomly generated value of the covariate(s) X for each individual.

`z` Contains the randomly generated value of the frailty variable Z for each individual.

`start` The start of interval `[start,stop]`, when the individual starts to be at risk for a next event.

`stop` The time of an event or censoring, i.e. the end of interval `[start,stop]`.

`status` An indicator of whether an event occured at time `stop` (`status=1`) or the individual is censored at time `stop` (`status=0`).

`fu` Length of follow-up period `[0,fu]` for each individual.

For each individual there are as many lines as it experiences events, plus one line if being censored. The data format corresponds to the counting process format.

Author(s)

Katharina Ingel, Stella Preussler, Antje Jahn-Eimermacher. Institute of Medical Biostatistics, Epidemiology and Informatics (IMBEI), University Medical Center of the Johannes Gutenberg-University Mainz, Germany

References

1. Andersen P, Gill R (1982): Cox's regression model for counting processes: a large sample study. The Annals of Statistics 10:1100-1120
2. Bender R, Augustin T, Blettner M (2005): Generating survival times to simulate Cox proportional hazards models. Statistics in Medicine 24:1713-1723
3. Jahn-Eimermacher A, Ingel K, Ozga AK, Preussler S, Binder H (2015): Simulating recurrent event data with hazard functions defined on a total time scale. BMC Medical Research Methodology 15:16

See Also

simreccomp

Examples

```
### Example:
### A sample of 10 individuals

N <- 10

### with a binomially distributed covariate with a regression coefficient
### of beta=0.3, and a standard normally distributed covariate with a
### regression coefficient of beta=0.2,

dist.x <- c("binomial", "normal")
par.x <- list(0.5, c(0, 1))
beta.x <- c(0.3, 0.2)

### a gamma distributed frailty variable with variance 0.25

dist.z <- "gamma"
par.z <- 0.25

### and a Weibull-shaped baseline hazard with shape parameter lambda=1
### and scale parameter nu=2.

dist.rec <- "weibull"
par.rec <- c(1, 2)

### Subjects are to be followed for two years with 20% of the subjects
### being censored according to a uniformly distributed censoring time
### within [0,2] (in years).

fu.min <- 2
fu.max <- 2
cens.prob <- 0.2

### After each event a subject is not at risk for experiencing further events
### for a period of 30 days with a probability of 50%.

dfree <- 30 / 365
```

```
pfree <- 0.5

simdata <- simrec(
  N, fu.min, fu.max, cens.prob, dist.x, par.x, beta.x, dist.z, par.z,
  dist.rec, par.rec, pfree, dfree
)
# print(simdata)  # only run for small N!
```

`simrec-pkg` *simrec*

Description

Simulation of recurrent event data for non-constant baseline hazard (total-time model)

Details

Simulation of recurrent event data for non-constant baseline hazard in the total time model with risk-free intervalls and possibly a competing event. The simrec package enables to cut the data to an interim data set, and provides functionality to plot.

Author(s)

Katharina Ingel, Stella Preussler, Antje Jahn-Eimermacher, Federico Marini

Maintainer: Antje Jahn-Eimermacher <jahna@uni-mainz.de>

`simreccomp` *simreccomp*

Description

This function allows simulation of time-to-event-data that follow a multistate-model with recurrent events of one type and a competing event. The baseline hazard for the cause-specific hazards are here functions of the total/calendar time. To induce between-subject-heterogeneity a random effect covariate (frailty term) can be incorporated for the recurrent and the competing event. Data for the recurrent events of the individual i are generated according to the cause-specific hazards

$$\lambda_{0r}(t) * Z_{ri} * exp(\beta_r^t X_i),$$

where X_i defines the covariate vector and β_r the regression coefficient vector. $\lambda_{0r}(t)$ denotes the baseline hazard, being a function of the total/calendar time t and Z_{ri} denotes the frailty variables with $(Z_{ri})_i$ iid with $E(Z_{ri}) = 1$ and $Var(Z_{ri}) = \theta_r$. The parameter θ_r describes the degree of between-subject-heterogeneity for the recurrent event. Analougously the competing event is generated according to the cause-specific hazard conditionally on the frailty variable and covariates:

$$\lambda_{0c}(t) * Z_{ci} * exp(\beta_c^t X_i)$$

Data output is in the counting process format.

Usage

```
simreccomp(
  N,
  fu.min,
  fu.max,
  cens.prob = 0,
  dist.x = "binomial",
  par.x = 0,
  beta.xr = 0,
  beta.xc = 0,
  dist.zr = "gamma",
  par.zr = 0,
  a = NULL,
  dist.zc = NULL,
  par.zc = NULL,
  dist.rec,
  par.rec,
  dist.comp,
  par.comp,
  pfree = 0,
  dfree = 0
)
```

Arguments

N	Number of individuals
fu.min	Minimum length of follow-up.
fu.max	Maximum length of follow-up. Individuals length of follow-up is generated from a uniform distribution on [fu.min,fu.max]. If fu.min=fu.max, then all individuals have a common follow-up.
cens.prob	Gives the probability of being censored due to loss to follow-up before fu.max. For a random set of individuals defined by a B(N,cens.prob)-distribution, the time to censoring is generated from a uniform distribution on [0,fu.max]. Default is cens.prob=0, i.e. no censoring due to loss to follow-up.
dist.x	Distribution of the covariate(s) X. If there is more than one covariate, dist.x must be a vector of distributions with one entry for each covariate. Possible values are "binomial" and "normal", default is dist.x="binomial".
par.x	Parameters of the covariate distribution(s). For "binomial", par.x is the probability for $x = 1$. For "normal", par.x=c(μ, σ) where μ is the mean and σ is the standard deviation of a normal distribution. If one of the covariates is defined to be normally distributed, par.x must be a list, e.g. dist.x <-c("binomial","normal") and par.x <-list(0.5,c(1,2)). Default is par.x=0, i.e. $x = 0$ for all individuals.
beta.xr	Regression coefficient(s) for the covariate(s) x corresponding to the recurrent events. If there is more than one covariate, beta.xr must be a vector of coefficients with one entry for each covariate. simreccomp generates as many

covariates as there are entries in `beta.xr`. Default is `beta.xr=0`, corresponding to no effect of the covariate x on the recurrent events.

`beta.xc` Regression coefficient(s) for the covariate(s) x corresponding to the competing event. If there is more than one covariate, `beta.xc` must be a vector of coefficients with one entry for each covariate. Default is `beta.xc=0`, corresponding to no effect of the covariate x on the competing event.

`dist.zr` Distribution of the frailty variable Z_r for the recurent events with $E(Z_r) = 1$ and $Var(Z_r) = \theta_r$. Possible values are `"gamma"` for a Gamma distributed frailty and `"lognormal"` for a lognormal distributed frailty. Default is `dist.zr="gamma"`.

`par.zr` Parameter θ_r for the frailty distribution: this parameter gives the variance of the frailty variable Z_r. Default is `par.zr=0`, which causes $Z_r = 1$, i.e. no frailty effect for the recurrent events.

`a` Alternatively, the frailty distribution for the competing event can be computed through the distribution of the frailty variable Z_r by $Z_c = Z_r * *a$. Default is `a=NULL`.

`dist.zc` Distribution of the frailty variable Z_c for the competing event with $E(Z_c) = 1$ and $Var(Z_c) = \theta_c$. Possible values are `"gamma"` for a Gamma distributed frailty and `"lognormal"` for a lognormal distributed frailty. Default is `dist.zc=NULL`.

`par.zc` Parameter θ_c for the frailty distribution: this parameter gives the variance of the frailty variable Z_c. Default is `par.zc=NULL`.

`dist.rec` Form of the baseline hazard function for the recurrent events. Possible values are `"weibull"` or `"gompertz"` or `"lognormal"` or `"step"`.

`par.rec` Parameters for the distribution of the recurrent event data. If `dist.rec="weibull"` the hazard function is

$$\lambda_0(t) = \lambda * \nu * t^{\nu-1},$$

where $\lambda > 0$ is the scale and $\nu > 0$ is the shape parameter. Then `par.rec=c(`λ, ν`)`. A special case of this is the exponential distribution for $\nu = 1$. If `dist.rec="gompertz"`, the hazard function is

$$\lambda_0(t) = \lambda * exp(\alpha t),$$

where $\lambda > 0$ is the scale and $\alpha \in (-\infty, +\infty)$ is the shape parameter. Then `par.rec=c(`λ, α`)`. If `dist.rec="lognormal"`, the hazard function is

$$\lambda_0(t) = [(1/(\sigma t)) * \phi((ln(t) - \mu)/\sigma)]/[\Phi((-ln(t) - \mu)/\sigma)],$$

where ϕ is the probability density function and Φ is the cumulative distribution function of the standard normal distribution, $\mu \in (-\infty, +\infty)$ is a location parameter and $\sigma > 0$ is a shape parameter. Then `par.rec=c(`μ, σ`)`. Please note, that specifying `dist.rec="lognormal"` together with some covariates does not specify the usual lognormal model (with covariates specified as effects on the parameters of the lognormal distribution resulting in non-proportional hazards), but only defines the baseline hazard and incorporates covariate effects using the proportional hazard assumption. If `dist.rec="step"` the hazard function is

$$\lambda_0(t) = a, t <= t_1, and \lambda_0(t) = b, t > t_1$$

. Then `par.rec=c(`a, b, t_1`)`.

dist.comp	Form of the baseline hazard function for the competing event. Possible values are "weibull" or "gompertz" or "lognormal" or "step" .
par.comp	Parameters for the distribution of the competing event data. For more details see par.rec.
pfree	Probability that after experiencing an event the individual is not at risk for experiencing further events for a length of dfree time units. Default is pfree=0.
dfree	Length of the risk-free interval. Must be in the same time unit as fu.max. Default is dfree=0, i.e. the individual is continously at risk for experiencing events until end of follow-up.

Value

The output is a data.frame consisting of the columns:

id	An integer number for identification of each individual
x	or x.V1,x.V2,... - depending on the covariate matrix. Contains the randomly generated value of the covariate(s) X for each individual.
zr	Contains the randomly generated value of the frailty variable Z_r for each individual.
zc	Contains the randomly generated value of the frailty variable Z_c for each individual.
start	The start of interval [start,stop], when the individual starts to be at risk for a next event.
stop	The time of an event or censoring, i.e. the end of interval [start,stop].
status	An indicator of whether an event occured at time stop (status=1), the individual is censored at time stop (status=0) or the competing event occured at time stop (status=2).
fu	Length of follow-up period [0,fu] for each individual.

For each individual there are as many lines as it experiences events, plus one line if being censored. The data format corresponds to the counting process format.

Author(s)

Katharina Ingel, Stella Preussler, Antje Jahn-Eimermacher. Institute of Medical Biostatistics, Epidemiology and Informatics (IMBEI), University Medical Center of the Johannes Gutenberg-University Mainz, Germany

See Also

simrec

Examples

```
### Example:
### A sample of 10 individuals

N <- 10

### with a binomially distributed covariate and a standard normally distributed covariate
### with regression coefficients of beta.xr=0.3 and beta.xr=0.2, respectively,
### for the recurrent events,
### as well as regression coefficients of beta.xc=0.5 and beta.xc=0.25, respectively,
### for the competing event.

dist.x <- c("binomial", "normal")
par.x <- list(0.5, c(0, 1))
beta.xr <- c(0.3, 0.2)
beta.xc <- c(0.5, 0.25)

### a gamma distributed frailty variable for the recurrent event with variance 0.25
### and for the competing event with variance 0.3,

dist.zr <- "gamma"
par.zr <- 0.25

dist.zc <- "gamma"
par.zc <- 0.3

### alternatively the frailty variable for the competing event can be computed via a:
a <- 0.5

### Furthermore a Weibull-shaped baseline hazard for the recurrent event with shape parameter
### lambda=1 and scale parameter nu=2,

dist.rec <- "weibull"
par.rec <- c(1, 2)

### and a Weibull-shaped baseline hazard for the competing event with shape parameter lambda=1
### and scale parameter nu=2

dist.comp <- "weibull"
par.comp <- c(1, 2)

### Subjects are to be followed for two years with 20% of the subjects
### being censored according to a uniformly distributed censoring time
### within [0,2] (in years).

fu.min <- 2
fu.max <- 2
cens.prob <- 0.2

### After each event a subject is not at risk for experiencing further events
### for a period of 30 days with a probability of 50%.
```

```
dfree <- 30 / 365
pfree <- 0.5

simdata1 <- simreccomp(
  N = N, fu.min = fu.min, fu.max = fu.max, cens.prob = cens.prob,
  dist.x = dist.x, par.x = par.x, beta.xr = beta.xr, beta.xc = beta.xc,
  dist.zr = dist.zr, par.zr = par.zr, a = a,
  dist.rec = dist.rec, par.rec = par.rec, dist.comp = dist.comp, par.comp = par.comp,
  pfree = pfree, dfree = dfree
)

simdata2 <- simreccomp(
  N = N, fu.min = fu.min, fu.max = fu.max, cens.prob = cens.prob,
  dist.x = dist.x, par.x = par.x, beta.xr = beta.xr, beta.xc = beta.xc,
  dist.zr = dist.zr, par.zr = par.zr, dist.zc = dist.zc, par.zc = par.zc,
  dist.rec = dist.rec, par.rec = par.rec, dist.comp = dist.comp, par.comp = par.comp,
  pfree = pfree, dfree = dfree
)

simdata1
simdata2
```

`simreccompPlot` *simreccompPlot*

Description

This function allows plotting of recurrent event data with a competing event.

Usage

```
simreccompPlot(
  data,
  id = "id",
  start = "start",
  stop = "stop",
  status = "status"
)
```

Arguments

`data`	A data set of recurrent event data to be plotted. The input-data must include columns corresponding to: `id` (patient-ID), `start` (= beginning of an interval where the patient is at risk for an event), `stop` (= end of the interval due to an event or censoring), `status` (= an indicator of the patient status at `stop` with = 0 censoring, 1 = event, 2 = competing event)
`id`	the name of the `id` column, default is `"id"`
`start`	the name of the `start` column, default is `"start"`

stop	the name of the stop column, default is "stop"
status	the name of the status column, default is "status"

Value

The output is a plot of the data with a bullet indicating a recurrent event, an x indicating the competing event and a circle indicating censoring.

Author(s)

Katharina Ingel, Stella Preussler, Antje Jahn-Eimermacher. Institute of Medical Biostatistics, Epidemiology and Informatics (IMBEI), University Medical Center of the Johannes Gutenberg-University Mainz, Germany

See Also

simrec, simreccomp, simrecPlot

Examples

```
### Example:
### First simulate a sample of 10 individuals (for more details see the help of \code{simreccomp})
N <- 10
dist.x <- c("binomial", "normal")
par.x <- list(0.5, c(0, 1))
beta.xr <- c(0.3, 0.2)
beta.xc <- c(0.5, 0.25)
dist.zr <- "gamma"
par.zr <- 0.25
dist.zc <- "gamma"
par.zc <- 0.3
dist.rec <- "weibull"
par.rec <- c(1, 2)
dist.comp <- "weibull"
par.comp <- c(1, 2)
fu.min <- 2
fu.max <- 2
cens.prob <- 0.2
dfree <- 30 / 365
pfree <- 0.5
simdata <- simreccomp(
  N = N, fu.min = fu.min, fu.max = fu.max, cens.prob = cens.prob,
  dist.x = dist.x, par.x = par.x, beta.xr = beta.xr, beta.xc = beta.xc,
  dist.zr = dist.zr, par.zr = par.zr, dist.zc = dist.zc, par.zc = par.zc,
  dist.rec = dist.rec, par.rec = par.rec, dist.comp = dist.comp, par.comp = par.comp,
  pfree = pfree, dfree = dfree
)
simreccompPlot(simdata)
```

simrecint *simrecint*

Description

With this function previously simulated data (for example simulated by the use of `simrec` or `simreccomp`) can be cut to an interim data set. The simulated data must be in patient time (i.e. time since the patient entered the study), and must be in the counting process format. Furthermore the dataset must have the variables `id`, `start`, `stop` and `status`, like data simulated by the use of `simrec` or `simreccomp`. Then for every individual additionally a recruitment time is generated in study time (i.e. time since start of the study), which is uniformly distributed on `[0,tR]`. The timing of the interim analysis `tI` is set in study time and data are being cut to all data, that are available at the interim analysis. For further explanations on study time and patient time see the vignette. If you only wish to simulate a recruitment time, `tI` can be set to `tR + fu.max` or something bigger.

Usage

```
simrecint(data, N, tR, tI)
```

Arguments

`data`	Previously generated data (in patient time), that shall be cut to interim data
`N`	Number of individuals, for which `data` was generated
`tR`	Length of the recruitment period (in study time)
`tI`	Timing of the interim analysis (in study time)

Value

The output is a data.frame consisting of the columns, that were put into, and additionally the following columns:

`rectime`	The recruitment time for each individual (in study time).
`interimtime`	The time of the interim analysis `tI` (in study time).
`stop_study`	The stopping time for each event in study time.

Individuals that are not already recruited at the interim analysis are left out here.

Author(s)

Katharina Ingel, Stella Preussler, Antje Jahn-Eimermacher. Institute of Medical Biostatistics, Epidemiology and Informatics (IMBEI), University Medical Center of the Johannes Gutenberg-University Mainz, Germany

See Also

simrec, simreccomp

Examples

```
### Example - see example for simrec
library(simrec)
N <- 10
dist.x <- c("binomial", "normal")
par.x <- list(0.5, c(0, 1))
beta.x <- c(0.3, 0.2)
dist.z <- "gamma"
par.z <- 0.25
dist.rec <- "weibull"
par.rec <- c(1, 2)
fu.min <- 2
fu.max <- 2
cens.prob <- 0.2

simdata <- simrec(
  N, fu.min, fu.max, cens.prob, dist.x, par.x, beta.x, dist.z,
  par.z, dist.rec, par.rec
)

### Now simulate for each patient a recruitment time in [0,tR=2]
### and cut data to the time of the interim analysis at tI=1:

simdataint <- simrecint(simdata, N = N, tR = 2, tI = 1)
# print(simdataint)  # only run for small N!
```

`simrecPlot` *simrecPlot*

Description

This function allows plotting of recurrent event data.

Usage

```
simrecPlot(data, id = "id", start = "start", stop = "stop", status = "status")
```

Arguments

`data`	A data set of recurrent event data to be plotted. The input-data must include columns corresponding to: `id` (patient-ID), `start` (= beginning of an interval where the patient is at risk for an event), `stop` (= end of the interval due to an event or censoring), `status` (= an indicator of the patient status at `stop` with = 0 censoring, 1 = event)
`id`	the name of the `id` column, default is `"id"`
`start`	the name of the `start` column, default is `"start"`
`stop`	the name of the `stop` column, default is `"stop"`
`status`	the name of the `status` column, default is `"status"`

Value

The output is a plot of the data with a bullet indicating a recurrent event and a circle indicating censoring.

Author(s)

Katharina Ingel, Stella Preussler, Antje Jahn-Eimermacher. Institute of Medical Biostatistics, Epidemiology and Informatics (IMBEI), University Medical Center of the Johannes Gutenberg-University Mainz, Germany

See Also

simrec, simreccomp, simreccompPlot

Examples

```
### Example:
### First simulate a sample of 10 individuals (for more details see the help of \code{simrec})
N <- 10
dist.x <- c("binomial", "normal")
par.x <- list(0.5, c(0, 1))
beta.x <- c(0.3, 0.2)
dist.z <- "gamma"
par.z <- 0.25
dist.rec <- "weibull"
par.rec <- c(1, 2)
fu.min <- 2
fu.max <- 2
cens.prob <- 0.2
dfree <- 30 / 365
pfree <- 0.5
simdata <- simrec(
  N, fu.min, fu.max, cens.prob, dist.x, par.x, beta.x,
  dist.z, par.z, dist.rec, par.rec, pfree, dfree
)
simrecPlot(simdata)
```

Index