

Package ‘survsim’

October 14, 2022

Type Package

Title Simulation of Simple and Complex Survival Data

Version 1.1.8

Date 2021-12-13

Encoding UTF-8

Maintainer David Moríña Soler <dmorina@ub.edu>

Description Simulation of simple and complex survival data including recurrent and multiple events and competing risks. See Moríña D, Navarro A. (2014) <[doi:10.18637/jss.v059.i02](https://doi.org/10.18637/jss.v059.i02)> and Moríña D, Navarro A. (2017) <[doi:10.1080/03080181.2017.1380000](https://doi.org/10.1080/03080181.2017.1380000)>

Depends R (>= 2.13.1), eha, statmod

License GPL (>= 2)

NeedsCompilation no

Author David Moríña Soler [aut, cre] (<<https://orcid.org/0000-0001-5949-7443>>),
Albert Navarro [aut]

Repository CRAN

Date/Publication 2021-12-14 15:30:08 UTC

R topics documented:

survsim-package	2
accum	4
crisk.sim	6
mult.ev.sim	8
rec.ev.sim	11
simple.surv.sim	14

Index	17
--------------	-----------

Description

Simulation of cohorts in a context of simple and complex survival analysis, multiple events and recurrent events including several covariates, individual heterogeneity and periods at risk before and after the initial time of follow-up.

Distribution	Survival function	Density function	Parametrization
Weibull	$\exp(-\lambda t^p)$	$\lambda p t^{p-1} \exp(-\lambda t^p)$	$\lambda = \exp(-p\beta_0)$
Log-normal	$1 - \Theta((\log(t) - \mu)/\sigma)$	$(1/(t\sigma\sqrt{2\pi}))\exp((-1/(2\sigma^2))(\log(t) - \mu)^2)$	$\mu = \beta_0$
Log-logistic	$1/(1 + (\lambda t)^{1/\gamma})$	$\lambda^{1/\gamma} t^{(1/\gamma)-1}/(\gamma(1 + (\lambda t)^{1/\gamma})^2)$	$\lambda = \exp(-\beta_0)$

Distribution	Time
Weibull	$t = (-\ln u/\lambda)^{1/p}$
Log-normal	$t = \exp(\beta_0 + \gamma(\log(u) - \log(1 - u)))$
Log-logistic	$t = \exp(\beta_0 + \sigma\Theta^{-1}(u))$

Where Θ is the standard normal cumulative distribution.

In order to simulate censored survival data, two survival distributions are required, one for the uncensored survival times that would be observed if the follow-up had been sufficiently long to reach the event and another representing the censoring mechanism. The uncensored survival distribution, T'_i , for $i = 1, \dots, n$ subjects, could be generated to depend on a set of covariates with a specified relationship with survival, which represents the true prognostic importance of each covariate (Burton, 2006). The package allows to simulate times by means of using Weibull (and exponential as a particular case), log-normal and log-logistic distributions, as such is showed in previous table. To induce individual heterogeneity or within-subject correlation we generate Z_i , a random effect covariate that follows a particular distribution (Uniform or Normal).

$$t_i = t'_i \cdot z_i$$

When $z_i = 1$, for all subjects, we are in the case of individual homogeneity and the survival times are completely specified by the covariates. Random non-informative right censoring, C_i , can be generated in a similar manner to the uncensored survival times, T'_i , by assuming a particular distribution for the censoring times (previous table), but without including any covariates nor individual heterogeneity. The observation times, Y'_i , incorporating both events and censored observations are calculated for each case by combining the uncensored survival times, T_i , and the censoring times, C_i . If the uncensored survival time for an observation is less than or equal to the censored time, then the event is considered to be observed and the observation time equals the uncensored survival time, otherwise the event is considered censored and the observation time equals the censored time. In other words, once simulated t_i and c_i , we can define $Y'_i = \min(t_i, c_i)$ as the observation time

with δ_i an indicator of non-censoring, i.e. $\delta_i = I(t_i \leq c_i)$. While all y'_i start at 0, the package allows create dynamic cohorts. We can generate entry times higher than 0 adding a t_0 value corresponding with an uniform distribution in $[0, t_{maxfollow-up}]$. We can also simulate subjects at risk before of the initial time of follow-up ($y'_i = 0$), by including an uniform distribution for t_0 between $[-t_{maxold}, 0)$ for a fixed percentage of subjects. Then:

$$y_i = y'_i + t_0$$

where t_0 follows a uniform distribution in $[0, t_{maxfollow-up}]$ if entry time is 0 or more and t_0 is uniform distributed in $[-t_{maxold}, 0)$ if entry time is less than 0. Therefore, t_0 represents the initial point of the episode, y_i the endpoint and y'_i is the lenght. Note that $y'_i + t_0$ can be higher than $t_{maxfollow-up}$, and in this case y_i will be set at $t_{maxfollow-up}$ and $\delta_i = 0$. The observations corresponding to the subjects at risk before of the initial time of follow-up have t_0 negative, then the initial point of the episode will be set at 0. y_i may also be negative, in this case this episode will not be included in the simulated data, as long as this episode won't be observed in practice.

Details

Package:	survsim
Type:	Package
Version:	1.1.8
Date:	2021-12-13
License:	GPL version 2 or newer
LazyLoad:	yes

The package provide a tool for simulation of cohorts in a simple single-event context through the function `simple.surv.sim`, in a recurrent event context with the function `rec.ev.sim`, in a multiple event context with the function `mult.ev.sim` and in a competing risks context with the function `crisk.sim`, and it also allows the user to generate aggregated data from the simulated cohort, by means of the function `accum`.

Author(s)

David Moraña, (Universitat de Barcelona) and Albert Navarro (Universitat Autònoma de Barcelona)
 Maintainer: David Moraña Soler <dmorina@ub.edu>

References

- Kelly PJ, Lim LL. Survival analysis for recurrent event data: an application to childhood infectious diseases. *Stat Med* 2000 Jan 15;**19**(1):13-33.
- Bender R, Augustin T, Blettner M. Generating survival times to simulate Cox proportional hazards models. *Stat Med* 2005 Jun 15;**24**(11):1713-1723.
- Metcalf C, Thompson SG. The importance of varying the event generation process in simulation studies of statistical methods for recurrent events. *Stat Med* 2006 Jan 15;**25**(1):165-179.
- Burton A, Altman DG, Royston P, Holder RL. The design of simulation studies in medical statistics. *Stat Med* 2006 Dec 30;**25**(24):4279-4292.

Beyersmann J, Latouche A, Buchholz A, Schumacher M. Simulating competing risks data in survival analysis. *Stat Med* 2009 Jan 5; **28**(1):956-971.

Reis RJ, Utzet M, La Rocca PF, Nedel FB, Martin M, Navarro A. Previous sick leaves as predictor of subsequent ones. *Int Arch Occup Environ Health* 2011 Jun; **84**(5):491-499.

Navarro A, Moriña D, Reis R, Nedel FB, Martin M, Alvarado S. Hazard functions to describe patterns of new and recurrent sick leave episodes for different diagnoses. *Scand J Work Environ Health* 2012 Jan 27.

Moriña D, Navarro A. The R package *survsim* for the simulation of simple and complex survival data. *Journal of Statistical Software* 2014 Jul; **59**(2):1-20.

accum

Aggregate data from a simulated cohort.

Description

Aggregate the observed number of events suffered by a subject, the time of follow-up, the duration of all the observed episodes and the real number of events suffered in all subject history.

Usage

```
accum(data)
```

Arguments

data	An object of class <code>mult.ev.data.sim</code> , if the individual cohort has been simulated in a multiple event situation or an object of class <code>rec.ev.data.sim</code> , if the individual cohort has been simulated in a recurrent event situation. Note that, although the routine will work, it's probably not much useful in other contexts than recurrent event situation.
------	--

Details

The output contains `z` and `real.ep.accum` because they can be interesting when analyzing several aspects as missing data or individual heterogeneity, although those variables cannot be observed in a real cohort.

Value

An object of class `sim.ev.agg.data`. It is a data frame with a row for each subject in `data`, and the following columns

nid	an integer number that identifies the subject.
old	real value indicating the time that the individual was at risk before the beginning of the follow-up.
risk.bef	Boolean indicating if the subject was at risk before the beginning of the follow-up time or not.

z	individual heterogeneity, generated according to the specified distribution.
x	value of each covariate randomly generated for each subject in the cohort.
obs.ep.accum	aggregated number of episodes suffered by an individual since the beginning of subject's follow-up time.
real.ep.accum	aggregated number of episodes suffered by an individual from the beginning of subject's history.
time.accum	global time of follow-up for each individual.
long.accum	global time not at risk within the follow-up time, corresponding to the sum of times between the end of an event and the beginning of the next.

Author(s)

David Moriña, Universitat de Barcelona and Albert Navarro, Universitat Autònoma de Barcelona

References

- Kelly PJ, Lim LL. Survival analysis for recurrent event data: an application to childhood infectious diseases. *Stat Med* 2000 Jan 15;**19**(1):13-33.
- Bender R, Augustin T, Blettner M. Generating survival times to simulate Cox proportional hazards models. *Stat Med* 2005 Jun 15;**24**(11):1713-1723.
- Metcalf C, Thompson SG. The importance of varying the event generation process in simulation studies of statistical methods for recurrent events. *Stat Med* 2006 Jan 15;**25**(1):165-179.
- Reis RJ, Utzet M, La Rocca PF, Nedel FB, Martin M, Navarro A. Previous sick leaves as predictor of subsequent ones. *Int Arch Occup Environ Health* 2011 Jun;**84**(5):491-499.
- Navarro A, Moriña D, Reis R, Nedel FB, Martin M, Alvarado S. Hazard functions to describe patterns of new and recurrent sick leave episodes for different diagnoses. *Scand J Work Environ Health* 2012 Jan 27.
- Moriña D, Navarro A. The R package *survsim* for the simulation of simple and complex survival data. *Journal of Statistical Software* 2014 Jul; **59**(2):1-20.

See Also

[rec.ev.sim](#), [mult.ev.sim](#), [crisk.sim](#), [survsim](#), [simple.surv.sim](#)

Examples

```
### A cohort with 500 subjects, with a maximum follow-up time of 1825 days and
### just a covariate, following a Bernoulli distribution, and a corresponding
### beta of -0.4, -0.5, -0.6 and -0.7 for each episode, in a context of recurrent
### events.

sim.data <- rec.ev.sim(n=500, foltime=1825, dist.ev=c('lnorm','llogistic', 'weibull',
'weibull'),anc.ev=c(1.498, 0.924, 0.923, 1.051),beta0.ev=c(7.195, 6.583, 6.678, 6.430)
, ,anc.cens=c(1.272, 1.218, 1.341, 1.484),beta0.cens=c(7.315, 6.975, 6.712, 6.399),
z=list(c("unif",0.8,1.2)),beta=list(c(-0.4,-0.5,-0.6,-0.7)), x=list(c("bern", 0.5)),
lambda=c(2.18,2.33,2.40,3.46),priskb=0.5,max.old=730)
```

```
### Aggregated data

accum.data <- accum(sim.data)

head(accum.data)
```

crisk.sim *Generate a cohort in a competing risks context*

Description

Simulation of cohorts in a context of competing risks survival analysis including several covariates, individual heterogeneity and periods at risk prior and after the start of follow-up.

Competing risks analysis considers time-to-first-event and the event type, possibly subject to right censoring (Beyersmann et al., 2009)

Usage

```
crisk.sim(n, foltime, dist.ev, anc.ev, beta0.ev, dist.cens="weibull",
anc.cens, beta0.cens, z=NULL, beta=NA, x=NA, nsit)
```

Arguments

n	integer value indicating the desired size of the cohort to be simulated.
foltime	real number that indicates the maximum time of follow-up of the simulated cohort.
dist.ev	vector of arbitrary size indicating the time to event distributions, with possible values weibull for the Weibull distribution, lnorm for the log-normal distribution and llogistic for the log-logistic distribution.
anc.ev	vector of arbitrary size of real components containing the ancillary parameters for the time to event distributions.
beta0.ev	vector of arbitrary size of real components containing the β_0 parameters for the time to event distributions.
dist.cens	string indicating the time to censoring distributions, with possible values weibull for the Weibull distribution, lnorm for the log-normal distribution, llogistic for the log-logistic distribution and unif for the uniform distribution. If no distribution is introduced, the time to censoring is assumed to follow a Weibull distribution.
anc.cens	real number containing the ancillary parameter for the time to censoring distribution or the maximum in case of uniform distributed time to censoring.
beta0.cens	real number containing the β_0 parameter for the time to censoring distribution or the minimum in case of uniform distributed time to censoring.

<code>z</code>	list of vectors with three elements containing information relative to a random effect used in order to introduce individual heterogeneity. Each vector in the list refers to a possible competing risk, so the number of vectors must be equal to <code>nsit</code> or equal to 1 if the same random effect will be used for all the causes. The first element indicates the distribution: "unif" states for a uniform distribution, "gamma" states for a gamma distribution, "exp" states for an exponential distribution, "weibull" states for a Weibull distribution and "invgauss" states for an inverse gaussian distribution. The second and third elements indicates the minimum and maximum in the case of a uniform distribution (both must be positive) and the parameters in the case of the rest of distributions. Note that just one parameter is needed in the case of exponential distribution. Its default value is NULL, indicating that no individual heterogeneity is introduced.
<code>beta</code>	list of vectors indicating the effect of the corresponding covariate. The number of vectors in <code>beta</code> must match the number of covariates, and the length of each vector must match the number of events considered. Its default value is NA, indicating that no covariates are included.
<code>x</code>	list of vectors indicating the distribution and parameters of any covariate that the user needs to introduce in the simulated cohort. The possible distributions are "normal" for normal distribution, "unif" for uniform distribution and "bern" for Bernoulli distribution. Its default value is NA, indicating that no covariates are included. The number of vectors in <code>x</code> must match the number of vectors in <code>beta</code> . Each vector in <code>x</code> must contain the name of the distribution and the parameter(s), which are: the probability of success in the case of a Bernoulli distribution; the mean and the variance in the case of a normal distribution; and the minimum and maximum in the case of a uniform distribution.
<code>nsit</code>	Number of different events that a subject can suffer. It must match the number of distributions specified in <code>dist.ev</code> .

Details

In order to get the function to work properly, the length of the vectors containing the parameters of the time to event and the number of distributions indicated in the parameter `dist.ev` must be the same.

Value

An object of class `mult.ev.data.sim`. It is a data frame containing the events suffered by the corresponding subjects. The columns of this data frame are detailed below

<code>nid</code>	an integer number that identifies the subject.
<code>cause</code>	cause of the event corresponding to the follow-up time of the individual.
<code>time</code>	time until the corresponding event happens (or time to subject drop-out).
<code>status</code>	logical value indicating if the corresponding event has been suffered or not.
<code>start</code>	time at which the follow-up time begins for each event.
<code>stop</code>	time at which the follow-up time ends for each event.
<code>z</code>	Individual heterogeneity generated according to the specified distribution.
<code>x</code>	value of each covariate randomly generated for each subject in the cohort.

Author(s)

David Morfiña, Universitat de Barcelona and Albert Navarro, Universitat Autònoma de Barcelona

References

Beyersmann J, Latouche A, Buchholz A, Schumacher M. Simulating competing risks data in survival analysis. *Stat Med* 2009 Jan 5;28(1):956-971.

See Also

[survsim-package](#), [accum](#), [rec.ev.sim](#), [mult.ev.sim](#), [simple.surv.sim](#)

Examples

```
### A cohort with 50 subjects, with a maximum follow-up time of 100 days and two
### covariates, following Bernoulli distributions, and a corresponding beta of
### 0.1698695 and 0.0007010932 for each event for the first covariate and a
### corresponding beta of 0.3735146 and 0.5591244 for each event for the
### second covariate. Notice that the time to censorship is assumed to follow a
### log-normal distribution.

sim.data <- crisk.sim(n=50, foltime=100, dist.ev=c("lnorm","lnorm"),
anc.ev=c(1.479687, 0.5268302),beta0.ev=c(3.80342, 2.535374),dist.cens="lnorm",
anc.cens=1.242733,beta0.cens=5.421748,z=list(c("unif", 0.8,1.2), c("unif", 0.9, 1.5)),
beta=list(c(0.1698695,0.0007010932),c(0.3735146,0.5591244)),
x=list(c("bern", 0.381), c("bern", 0.564)), nsit=2)

summary(sim.data)
```

mult.ev.sim

Generate a cohort with multiple events

Description

Simulation of cohorts in a context of multiple event survival analysis including several covariates, individual heterogeneity and periods at risk prior and after the start of follow-up.

Multiple event data occurs when each subject can have more than one event of entirely different natures (Kelly, 2000). Examples of this type of event are the occurrence of tumours at different sites in the body or multiple sequelae after a surgery.

We can obtain the observation time of the k -th event in the i -th subject, y_{ik} , in the same manner that we can simulate k simple independent survival data. Notice that, in multiple-type events, T_{ik} and C_i are mutually independent and, furthermore, the failure in each event is independent of the others (within each subject all y_{ik} are independents for all k).

Usage

```
mult.ev.sim(n, foltime, dist.ev, anc.ev, beta0.ev, dist.cens="weibull",
anc.cens, beta0.cens, z=NULL, beta=NA, x=NA, nsit)
```


Arguments

n	integer value indicating the desired size of the cohort to be simulated.
foltime	real number that indicates the maximum time of follow-up of the simulated cohort.
dist.ev	vector of arbitrary size indicating the time to event distributions, with possible values <code>weibull</code> for the Weibull distribution, <code>lnorm</code> for the log-normal distribution and <code>llogistic</code> for the log-logistic distribution.
anc.ev	vector of arbitrary size of real components containing the ancillary parameters for the time to event distributions.
beta0.ev	vector of arbitrary size of real components containing the β_0 parameters for the time to event distributions.
dist.cens	string indicating the time to censoring distributions, with possible values <code>weibull</code> for the Weibull distribution, <code>lnorm</code> for the log-normal distribution, <code>llogistic</code> for the log-logistic distribution and <code>unif</code> for the uniform distribution. If no distribution is introduced, the time to censoring is assumed to follow a Weibull distribution.
anc.cens	real number containing the ancillary parameter for the time to censoring distribution or the maximum in case of uniform distributed time to censoring.
beta0.cens	real number containing the β_0 parameter for the time to censoring distribution or the minimum in case of uniform distributed time to censoring.
z	list of vectors with three elements containing information relative to a random effect used in order to introduce individual heterogeneity. Each vector in the list refers to a possible event, so the number of vectors must be equal to <code>nsit</code> or equal to 1 if the same random effect will be used for all the events. The first element indicates the distribution: "unif" states for a uniform distribution, "gamma" states for a gamma distribution, "exp" states for an exponential distribution, "weibull" states for a Weibull distribution and "invgauss" states for an inverse gaussian distribution. The second and third elements indicates the minimum and maximum in the case of a uniform distribution (both must be positive) and the parameters in the case of the rest of distributions. Note that just one parameter is needed in the case of exponential distribution. Its default value is <code>NULL</code> , indicating that no individual heterogeneity is introduced.
beta	list of vectors indicating the effect of the corresponding covariate. The number of vectors in <code>beta</code> must match the number of covariates, and the length of each vector must match the number of events considered. Its default value is <code>NA</code> , indicating that no covariates are included.
x	list of vectors indicating the distribution and parameters of any covariate that the user needs to introduce in the simulated cohort. The possible distributions are "normal" for normal distribution, "unif" for uniform distribution and "bern" for Bernoulli distribution. Its default value is <code>NA</code> , indicating that no covariates are included. The number of vectors in <code>x</code> must match the number of vectors in <code>beta</code> . Each vector in <code>x</code> must contain the name of the distribution and the parameter(s), which are: the probability of success in the case of a Bernoulli distribution; the mean and the variance in the case of a normal distribution; and the minimum and maximum in the case of a uniform distribution.

`nsit` Number of different events that a subject can suffer. It must match the number of distributions specified in `dist.ev`.

Details

In order to get the function to work properly, the length of the vectors containing the parameters of the time to event and the number of distributions indicated in the parameter `dist.ev` must be the same.

Value

An object of class `mult.ev.data.sim`. It is a data frame containing the events suffered by the corresponding subjects. The columns of this data frame are detailed below

<code>nid</code>	an integer number that identifies the subject.
<code>ev.num</code>	number of the event corresponding to the follow-up time of the individual.
<code>time</code>	time until the corresponding event happens (or time to subject drop-out).
<code>status</code>	logical value indicating if the corresponding event has been suffered or not.
<code>start</code>	time at which the follow-up time begins for each event.
<code>stop</code>	time at which the follow-up time ends for each event.
<code>z</code>	Individual heterogeneity generated according to the specified distribution.
<code>x</code>	value of each covariate randomly generated for each subject in the cohort.

Author(s)

David Moríña, Universitat de Barcelona and Albert Navarro, Universitat Autònoma de Barcelona

References

Kelly PJ, Lim LL. Survival analysis for recurrent event data: an application to childhood infectious diseases. *Stat Med* 2000 Jan 15; **19**(1):13-33.

Bender R, Augustin T, Blettner M. Generating survival times to simulate Cox proportional hazards models. *Stat Med* 2005 Jun 15; **24**(11):1713-1723.

Metcalf C, Thompson SG. The importance of varying the event generation process in simulation studies of statistical methods for recurrent events. *Stat Med* 2006 Jan 15; **25**(1):165-179.

Moríña D, Navarro A. The R package `survsim` for the simulation of simple and complex survival data. *Journal of Statistical Software* 2014 Jul; **59**(2):1-20.

See Also

[survsim-package](#), [accum](#), [rec.ev.sim](#), [crisk.sim](#), [simple.surv.sim](#)

Examples

```
### A cohort with 1000 subjects, with a maximum follow-up time of 3600 days and two
### covariates, following a Bernoulli and uniform distribution respectively, and a
### corresponding beta of -0.4, -0.5, -0.6 and -0.7 for each event for the first
### covariate and a corresponding beta of 0, 0, 0 and 1 for each event for the
### second covariate. Notice that the time to censorship is assumed to follow a
### Weibull distribution, as no other distribution is stated and random effect is
### the same for all events.

sim.data <- mult.ev.sim(n=1000, foltime=3600, dist.ev=c('llogistic','weibull',
'weibull','weibull'),anc.ev=c(0.69978200185280, 0.79691659193027,
0.82218416457321, 0.85817155198598),beta0.ev=c(5.84298525742252, 5.94362650803287,
5.78182528904637, 5.4686522339755),,anc.cens=1.17783687569519,
beta0.cens=7.39773677281100,z=list(c("unif", 0.8,1.2)), beta=list(c(-0.4,-0.5,-0.6,-0.7),
c(0,0,0,1)), x=list(c("bern", 0.5), c("unif", 0.7, 1.3)), nsit=4)

summary(sim.data)
```

```
rec.ev.sim
```

Generate a cohort with recurrent events

Description

Simulation of cohorts in a context of recurrent event survival analysis including several covariates, individual heterogeneity and periods at risk before and after the initial time of follow-up.

Recurrent event data is a type of multiple event where the subject can experience repeated occurrences of the same type (Kelly, 2000), for example repeated asthma attacks or sick leave episodes. In practice, the hazard of an recurrent event can vary depending on the number of previous occurrences, in terms of shape and intensity (Reis, 2011; Navarro, 2012). However, simulations based on a mixture of distributions with different baseline hazard rates are quite rare (Bender, 2005; Metcalfe, 2006).

In a recurrent data context, each subject can present different number of episodes. We talk of episodes (or occurrences) rather than events since each occurrence is a new episode of the same event. This package supposes that there exists one different and independent Y_k distribution depending on k , the number of episode at risk. The simulating process for each Y_k is the same than for the multiple events situation (see [mult.ev.sim](#)), but in this case, obviously, a subject cannot be at risk for the k -th episode if he/she hadn't had the $k - 1$ -th.

Usage

```
rec.ev.sim(n, foltime, dist.ev, anc.ev, beta0.ev, dist.cens=rep("weibull",
length(beta0.cens)), anc.cens, beta0.cens, z=NULL, beta=NA, x=NA, lambda=NA,
max.ep=Inf, priskb=0, max.old=0)
```

Arguments

n	integer value indicating the desired size of the cohort to be simulated.
foltime	real number that indicates the maximum time of follow-up of the simulated cohort.
dist.ev	vector of arbitrary size indicating the time to event distributions, with possible values <code>weibull</code> for the Weibull distribution, <code>lnorm</code> for the log-normal distribution and <code>llogistic</code> for the log-logistic distribution. If a subject suffers more episodes than specified distributions, the last distribution specified here is used to generate times corresponding to posterior episodes.
anc.ev	vector of arbitrary size of real components containing the ancillary parameters for the time to event distributions.
beta0.ev	vector of arbitrary size of real components containing the β_0 parameters for the time to event distributions.
dist.cens	string indicating the time to censoring distributions, with possible values <code>weibull</code> for the Weibull distribution, <code>lnorm</code> for the log-normal distribution, <code>llogistic</code> for the log-logistic distribution and <code>unif</code> for the uniform distribution. If no distribution is introduced, the time to censoring is assumed to follow a Weibull distribution.
anc.cens	real number containing the ancillary parameter for the time to censoring distribution or the maximum in case of uniform distributed time to censoring.
beta0.cens	real number containing the β_0 parameter for the time to censoring distribution or the minimum in case of uniform distributed time to censoring.
z	list of vectors with three elements containing information relative to a random effect used in order to introduce individual heterogeneity. Each vector in the list refers to a possible episode, so the number of vectors must be equal to <code>nsit</code> or equal to 1 if the same random effect will be used for all the episodes. The first element indicates the distribution: <code>"unif"</code> states for a uniform distribution, <code>"gamma"</code> states for a gamma distribution, <code>"exp"</code> states for an exponential distribution, <code>"weibull"</code> states for a Weibull distribution and <code>"invgauss"</code> states for an inverse gaussian distribution. The second and third elements indicates the minimum and maximum in the case of a uniform distribution (both must be positive) and the parameters in the case of the rest of distributions. Note that just one parameter is needed in the case of exponential distribution. Its default value is <code>NULL</code> , indicating that no individual heterogeneity is introduced.
beta	list of vectors indicating the effect of the corresponding covariate. The number of vectors in <code>beta</code> must match the number of covariates, and the length of each vector must match the number of events considered. Its default value is <code>NA</code> , indicating that no covariates are included.
x	list of vectors indicating the distribution and parameters of any covariate that the user need to introduce in the simulated cohort. The possible distributions are <code>"normal"</code> for a normal distribution, <code>"unif"</code> for a uniform distribution and <code>"bern"</code> for a Bernoulli distribution. Its default value is <code>NA</code> , indicating that no covariates are included. The number of vectors in <code>x</code> must match the number of vectors in <code>beta</code> . Each vector in <code>x</code> must contain the name of the distribution and the parameter(s), which are: the probability of success in the case of a Bernoulli

	distribution; the mean and the variance in the case of a normal distribution; and the minimum and maximum in the case of a uniform distribution.
lambda	real number indicating the mean duration of each event or discontinuous risk time, assumed to follow a zero-truncated Poisson distribution. Its default value is NA, corresponding to the case where the duration of each event or discontinuous risk time is unnecessary information for the user.
max.ep	integer value that matches the maximum permitted number of episodes per subject. Its default value is Inf, i.e. the number of episodes per subject is no limited.
priskb	proportion of subjects at risk prior to the start of follow-up, defaults to 0.
max.old	maximum time at risk prior to the start of follow-up.

Details

In order to get the function to work properly, the length of the vectors containing the parameters of the time to event and time to censor distributions and the number of distributions indicated in the parameter `dist` must be the same. Finally, `priskb` and `max.old` must be positive numbers, with `priskb` being between 0 and 1. Notice that large values of `max.old` can result in the routine taking a long time to simulate a cohort with the specified size.

Value

An object of class `rec.ev.data.sim`. It is a data frame containing the episodes suffered by the corresponding subjects. The columns of the data frame are detailed below

nid	an integer number that identifies the subject.
real.episode	number of the episode corresponding to the real history of the individual.
obs.episode	number of the episode corresponding to the follow-up time of the individual.
time	time until the corresponding event happens (or time to subject drop-out), regarding the beginning of the follow-up time.
status	logical value indicating if the episode corresponds to an event or a drop-out.
start	time at which an episode starts, taking the beginning of follow-up as the origin of the time scale.
stop	time at which an episode ends, taking the beginning of follow-up as the origin of the time scale.
time2	time until the corresponding event happens (or time to subject drop-out), in calendar time.
start2	time at which an episode starts, where the time scale is calendar time.
stop2	time at which an episode ends, where the time scale is calendar time.
old	real value indicating the time that the individual was at risk before the beginning of follow-up.
risk.bef	factor that indicates if an individual was at risk before the beginning of follow-up or not.
long	time not at risk immediately after an episode.
z	Individual heterogeneity generated according to the specified distribution.
x	value of each covariate randomly generated for each subject in the cohort.

Author(s)

David Moriña, Universitat de Barcelona and Albert Navarro, Universitat Autònoma de Barcelona

References

Kelly PJ, Lim LL. Survival analysis for recurrent event data: an application to childhood infectious diseases. *Stat Med* 2000 Jan 15;**19**(1):13-33.

Bender R, Augustin T, Blettner M. Generating survival times to simulate Cox proportional hazards models. *Stat Med* 2005 Jun 15;**24**(11):1713-1723.

Metcalf C, Thompson SG. The importance of varying the event generation process in simulation studies of statistical methods for recurrent events. *Stat Med* 2006 Jan 15;**25**(1):165-179.

Reis RJ, Utzet M, La Rocca PF, Nedel FB, Martin M, Navarro A. Previous sick leaves as predictor of subsequent ones. *Int Arch Occup Environ Health* 2011 Jun;**84**(5):491-499.

Navarro A, Moriña D, Reis R, Nedel FB, Martin M, Alvarado S. Hazard functions to describe patterns of new and recurrent sick leave episodes for different diagnoses. *Scand J Work Environ Health* 2012 Jan 27.

Moriña D, Navarro A. The R package survsim for the simulation of simple and complex survival data. *Journal of Statistical Software* 2014 Jul; **59**(2):1-20.

See Also

[survsim-package](#), [accum](#), [mult.ev.sim](#), [simple.surv.sim](#), [crisk.sim](#)

Examples

```
### A cohort with 500 subjects, with a maximum follow-up time of 1825 days and
### just a covariate, following a Bernoulli distribution, and a corresponding
### beta of -0.4, -0.5, -0.6 and -0.7 for each episode. Note that random effect is
### the same for all events.
```

```
sim.data <- rec.ev.sim(n=500, foltime=1825, dist.ev=c('lnorm','llogistic',
'weibull','weibull'),anc.ev=c(1.498, 0.924, 0.923, 1.051),beta0.ev=c(7.195,
6.583, 6.678, 6.430),,anc.cens=c(1.272, 1.218, 1.341, 1.484),
beta0.cens=c(7.315, 6.975, 6.712, 6.399), z=list(c("unif", 0.8,1.2)),
beta=list(c(-0.4,-0.5,-0.6,-0.7)), x=list(c("bern", 0.5)),
lambda=c(2.18,2.33,2.40,3.46), priskb=0.5, max.old=730)
```

```
summary(sim.data)
```

simple.surv.sim

Generate a cohort with single-event survival times

Description

Simulation of cohorts in a context of standard survival analysis including several covariates and individual heterogeneity.

Usage

```
simple.surv.sim(n, foltime, dist.ev, anc.ev, beta0.ev, dist.cens="weibull",
  anc.cens, beta0.cens, z=NULL, beta=NA, x=NA)
```

Arguments

n	integer value indicating the desired size of the cohort to be simulated.
foltime	real number that indicates the maximum time of follow-up of the simulated cohort.
dist.ev	time to event distributions, with possible values <code>weibull</code> for the Weibull distribution, <code>lnorm</code> for the log-normal distribution and <code>llogistic</code> for the log-logistic distribution.
anc.ev	ancillary parameter for the time to event distribution.
beta0.ev	β_0 parameter for the time to event distribution.
dist.cens	string indicating the time to censoring distributions, with possible values <code>weibull</code> for the Weibull distribution, <code>lnorm</code> for the log-normal distribution, <code>llogistic</code> for the log-logistic distribution and <code>unif</code> for the uniform distribution. If no distribution is introduced, the time to censoring is assumed to follow a Weibull distribution.
anc.cens	real number containing the ancillary parameter for the time to censoring distribution or the maximum in case of uniform distributed time to censoring.
beta0.cens	real number containing the β_0 parameter for the time to censoring distribution or the minimum in case of uniform distributed time to censoring.
z	vector with three elements that contains information relative to a random effect used in order to introduce individual heterogeneity. The first element indicates the distribution: <code>"unif"</code> states for a uniform distribution, <code>"gamma"</code> states for a gamma distribution, <code>"exp"</code> states for an exponential distribution, <code>"weibull"</code> states for a Weibull distribution and <code>"invgauss"</code> states for an inverse gaussian distribution. The second and third elements indicates the minimum and maximum in the case of a uniform distribution (both must be positive) and the parameters in the case of the rest of distributions. Notice that that just one parameter is needed in the case of exponential distribution. Its default value is <code>NULL</code> , indicating that no individual heterogeneity is introduced.
beta	list of elements indicating the effect of the corresponding covariate. The number of vectors in <code>beta</code> must match the number of covariates. Its default value is <code>NA</code> , indicating that no covariates are included.
x	list of vectors indicating the distribution and parameters of any covariate that the user needs to introduce in the simulated cohort. The possible distributions are <code>"normal"</code> for normal distribution, <code>"unif"</code> for uniform distribution and <code>"bern"</code> for Bernoulli distribution. Its default value is <code>NA</code> , indicating that no covariates are included. The number of vectors in <code>x</code> must match the number of vectors in <code>beta</code> . Each vector in <code>x</code> must contain the name of the distribution and the parameter(s), which are: the probability of success in the case of a Bernoulli distribution; the mean and the variance in the case of a normal distribution; and the minimum and maximum in the case of a uniform distribution.

Value

An object of class `simple.surv.sim`. It is a data frame containing the events suffered by the corresponding subjects. The columns of this data frame are detailed below

<code>nid</code>	an integer number that identifies the subject.
<code>status</code>	logical value indicating if the corresponding event has been suffered or not.
<code>start</code>	time at which the follow-up time begins for each event.
<code>stop</code>	time at which the follow-up time ends for each event.
<code>z</code>	Individual heterogeneity generated according to the specified distribution.
<code>x</code>	value of each covariate randomly generated for each subject in the cohort.

Author(s)

David Moríña, Universitat de Barcelona and Albert Navarro, Universitat Autònoma de Barcelona

References

Kelly PJ, Lim LL. Survival analysis for recurrent event data: an application to childhood infectious diseases. *Stat Med* 2000 Jan 15; **19**(1):13-33.

Bender R, Augustin T, Blettner M. Generating survival times to simulate Cox proportional hazards models. *Stat Med* 2005 Jun 15; **24**(11):1713-1723.

Moríña D, Navarro A. The R package `survsim` for the simulation of simple and complex survival data. *Journal of Statistical Software* 2014 Jul; **59**(2):1-20.

See Also

[survsim-package](#), [accum](#), [rec.ev.sim](#), [mult.ev.sim](#), [crisk.sim](#)

Examples

```
### A cohort with 1000 subjects, with a maximum follow-up time of 3600 days and two
### covariates, following a Bernoulli and uniform distribution respectively, and a
### corresponding beta of -0.4 for the first covariate and a corresponding beta of 0
### for the second covariate. Notice that the time to censorship is assumed to
### follow a Weibull distribution, as no other distribution is stated.
```

```
sim.data <- simple.surv.sim(n=1000, foltime=3600, dist.ev=c('llogistic'),
anc.ev=c(0.69978200185280),beta0.ev=c(5.84298525742252),,anc.cens=1.17783687569519,
beta0.cens=7.39773677281100,z=list(c("unif", 0.8, 1.2)), beta=list(c(-0.4),
c(0)), x=list(c("bern", 0.5), c("unif", 0.7, 1.3)))
```

```
summary(sim.data)
```


Index

- * **aggregated**
 - accum, 4
 - * **competing risks**
 - crisk.sim, 6
 - * **individual**
 - crisk.sim, 6
 - mult.ev.sim, 8
 - rec.ev.sim, 11
 - simple.surv.sim, 14
 - * **multiple**
 - mult.ev.sim, 8
 - * **recurrent**
 - rec.ev.sim, 11
 - * **simulation**
 - crisk.sim, 6
 - mult.ev.sim, 8
 - rec.ev.sim, 11
 - simple.surv.sim, 14
 - * **single**
 - simple.surv.sim, 14
 - * **survival**
 - crisk.sim, 6
 - mult.ev.sim, 8
 - rec.ev.sim, 11
 - simple.surv.sim, 14
 - * **survsim**
 - accum, 4
 - crisk.sim, 6
 - mult.ev.sim, 8
 - rec.ev.sim, 11
 - simple.surv.sim, 14
 - survsim-package, 2
- accum, 3, 4, 8, 10, 14, 16
- crisk.sim, 3, 5, 6, 10, 14, 16
- mult.ev.sim, 3, 5, 8, 8, 11, 14, 16
- rec.ev.sim, 3, 5, 8, 10, 11, 16
- simple.surv.sim, 3, 5, 8, 10, 14, 14
- survsim, 5
- survsim(survsim-package), 2
- survsim-package, 2