

Package ‘rsurv’

October 25, 2024

Title Random Generation of Survival Data

Version 0.0.2

Date 2024-10-24

Description Random generation of survival data from a wide range of regression models, including accelerated failure time (AFT), proportional hazards (PH), proportional odds (PO), accelerated hazard (AH), Yang and Prentice (YP), and extended hazard (EH) models. The package 'rsurv' also stands out by its ability to generate survival data from an unlimited number of baseline distributions provided that an implementation of the quantile function of the chosen baseline distribution is available in R. Another nice feature of the package 'rsurv' lies in the fact that linear predictors are specified via a formula-based approach, facilitating the inclusion of categorical variables and interaction terms. The functions implemented in the package 'rsurv' can also be employed to simulate survival data with more complex structures, such as survival data with different types of censoring mechanisms, survival data with cure fraction, survival data with random effects (frailties), multivariate survival data, and competing risks survival data. Details about the R package 'rsurv' can be found in Demarqui (2024) <[doi:10.48550/arXiv.2406.01750](https://doi.org/10.48550/arXiv.2406.01750)>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.4.0)

Imports bellreg (>= 0.0.2.2), dplyr, MASS, Rdpack, stabledist

RdMacros Rdpack

Suggests copula, flexsurv, frailtyEM, GGally, knitr, LambertW, rmarkdown, survival, survstan, testthat (>= 3.0.0)

VignetteBuilder knitr

URL <https://github.com/fndemarqui/rsurv>,
<https://fndemarqui.github.io/rsurv/>

BugReports <https://github.com/fndemarqui/rsurv/issues>

Config/testthat/edition 3

NeedsCompilation no

Author Fabio Demarqui [aut, cre, cph]
 (<<https://orcid.org/0000-0001-9236-1986>>)
Maintainer Fabio Demarqui <fndemarqui@est.ufmg.br>
Repository CRAN
Date/Publication 2024-10-25 04:10:02 UTC

Contents

rsurv-package	2
bernoulli	3
inv_pgf	3
lp	4
negbin	5
qsurv	5
raftreg	6
rahreg	7
rehreg	8
rfrailty	9
rinterval	10
rphreg	10
rporeg	11
rypreg	12
Index	14

rsurv-package	<i>The 'rsurv' package</i>
---------------	----------------------------

Description

Random generation of survival data based on different survival regression models available in the literature, including Accelerated Failure Time (AFT) model, Proportional Hazard (PH) model, Proportional Odds (PO) model and the Yang & Prentice (YP) model.

`_PACKAGE`

References

- Demarqui FN, Mayrink VD (2021). “Yang and Prentice model with piecewise exponential baseline distribution for modeling lifetime data with crossing survival curves.” *Brazilian Journal of Probability and Statistics*, **35**(1), 172 – 186. doi:10.1214/20BJPS471.
- Yang S, Prentice RL (2005). “Semiparametric analysis of short-term and long-term hazard ratios with two-sample survival data.” *Biometrika*, **92**(1), 1-17.

 bernoulli

Implemented link functions for the mixture cure rate model

Description

This function is used to specify different link functions for the count component of the mixture cure rate model.

Usage

```
bernoulli(link = "logit")
```

Arguments

link desired link function; currently implemented links are: logit, probit, cloglog and cauchy.

Value

A list containing the codes associated with the count distribution assumed for the latent variable N and the chosen link.

inv_pgf

Inverse of the probability generating function

Description

This function is used to specify different link functions for the count component of the promotion time cure rate model

Usage

```
inv_pgf(formula, incidence = "bernoulli", kappa = NULL, zeta = NULL, data, ...)
```

Arguments

formula formula specifying the linear predictor for the incidence sub-model.
 incidence the desired incidence model.
 kappa vector of regression coefficients associated with the incidence sub-model.
 zeta extra negative-binomial parameter.
 data a data.frame containing the explanatory covariates passed to the formula.
 ... further arguments passed to other methods.

Value

A vector with the values of the inverse of the desired probability generating function.

lp	<i>Linear predictors</i>
----	--------------------------

Description

Function to construct linear predictors.

Usage

```
lp(formula, coefs, data, ...)
```

Arguments

formula	formula specifying the linear predictors.
coefs	vector of regression coefficients.
data	data frame containing the covariates used to construct the linear predictors.
...	further arguments passed to other methods.

Value

a vector containing the linear predictors.

Examples

```
library(rsurv)
library(dplyr)

n <- 100
coefs <- c(1, 0.7, 2.3)

simdata <- data.frame(
  age = rnorm(n),
  sex = sample(c("male", "female"), size = n, replace = TRUE)
) |>
  mutate(
    lp = lp(~age+sex, coefs)
  )
glimpse(simdata)
```

negbin	<i>Implemented link functions for the promotion time cure rate model with negative binomial distribution</i>
--------	--

Description

This function is used to specify different link functions for the count component of the promotion time cure rate model.

Usage

```
negbin(zeta = stop("'theta' must be specified"), link = "log")
```

Arguments

zeta	The known value of the additional parameter.
link	desired link function; currently implemented links are: log, identity and sqrt.

Value

A list containing the codes associated with the count distribution assumed for the latent variable N and the chosen link.

qsurv	<i>Generic quantile function</i>
-------	----------------------------------

Description

Generic quantile function used internally to simulating from an arbitrary baseline survival distribution.

Usage

```
qsurv(p, baseline, package = NULL, ...)
```

Arguments

p	vector of quantiles associated with the right tail area of the baseline survival distribution.
baseline	the name of the baseline distribution.
package	the name of the package where the baseline distribution is implemented. It ensures that the right quantile function from the right package is found, regardless of the current R search path.
...	further arguments passed to other methods.

Value

a vector of quantiles.

Examples

```
library(rsurv)
set.seed(1234567890)

u <- sort(runif(5))
x1 <- qexp(u, rate = 1, lower.tail = FALSE)
x2 <- qsurv(u, baseline = "exp", rate = 1)
x3 <- qsurv(u, baseline = "exp", rate = 1, package = "stats")
x4 <- qsurv(u, baseline = "gengamma.orig", shape=1, scale=1, k=1, package = "flexsurv")

cbind(x1, x2, x3, x4)
```

raftreg

Random generation from accelerated failure time models

Description

Function to generate a random sample of survival data from accelerated failure time models.

Usage

```
raftreg(u, formula, baseline, beta, dist = NULL, package = NULL, data, ...)
```

Arguments

u	a numeric vector of quantiles.
formula	formula specifying the linear predictors.
baseline	the name of the baseline survival distribution.
beta	vector of regression coefficients.
dist	an alternative way to specify the baseline survival distribution.
package	the name of the package where the assumed quantile function is implemented.
data	data frame containing the covariates used to generate the survival times.
...	further arguments passed to other methods.

Value

a numeric vector containing the generated random sample.

Examples

```

library(rsurv)
library(dplyr)
n <- 1000
simdata <- data.frame(
  age = rnorm(n),
  sex = sample(c("f", "m"), size = n, replace = TRUE)
) %>%
  mutate(
    t = raftreg(runif(n), ~ age+sex, beta = c(1, 2),
               dist = "weibull", shape = 1.5, scale = 1),
    c = runif(n, 0, 10)
  ) %>%
  rowwise() %>%
  mutate(
    time = min(t, c),
    status = as.numeric(time == t)
  )
glimpse(simdata)

```

rahreg

Random generation from accelerated hazard models

Description

Function to generate a random sample of survival data from accelerated hazard models.

Usage

```
rahreg(u, formula, baseline, beta, dist = NULL, package = NULL, data, ...)
```

Arguments

<code>u</code>	a numeric vector of quantiles.
<code>formula</code>	formula specifying the linear predictors.
<code>baseline</code>	the name of the baseline survival distribution.
<code>beta</code>	vector of regression coefficients.
<code>dist</code>	an alternative way to specify the baseline survival distribution.
<code>package</code>	the name of the package where the assumed quantile function is implemented.
<code>data</code>	data frame containing the covariates used to generate the survival times.
<code>...</code>	further arguments passed to other methods.

Value

a numeric vector containing the generated random sample.

Examples

```

library(rsurv)
library(dplyr)
n <- 1000
simdata <- data.frame(
  age = rnorm(n),
  sex = sample(c("f", "m"), size = n, replace = TRUE)
) %>%
  mutate(
    t = rahreg(runif(n), ~ age+sex, beta = c(1, 2),
              dist = "weibull", shape = 1.5, scale = 1),
    c = runif(n, 0, 10)
  ) %>%
  rowwise() %>%
  mutate(
    time = min(t, c),
    status = as.numeric(time == t)
  )
glimpse(simdata)

```

rehreg

Random generation from extended hazard models

Description

Function to generate a random sample of survival data from extended hazard models.

Usage

```
rehreg(u, formula, baseline, beta, phi, dist = NULL, package = NULL, data, ...)
```

Arguments

u	a numeric vector of quantiles.
formula	formula specifying the linear predictors.
baseline	the name of the baseline survival distribution.
beta	vector of regression coefficients.
phi	vector of regression coefficients.
dist	an alternative way to specify the baseline survival distribution.
package	the name of the package where the assumed quantile function is implemented.
data	data frame containing the covariates used to generate the survival times.
...	further arguments passed to other methods.

Value

a numeric vector containing the generated random sample.

Examples

```

library(rsurv)
library(dplyr)
n <- 1000
simdata <- data.frame(
  age = rnorm(n),
  sex = sample(c("f", "m"), size = n, replace = TRUE)
) %>%
  mutate(
    t = rehref(runif(n), ~ age+sex, beta = c(1, 2), phi = c(-1, 2),
              dist = "weibull", shape = 1.5, scale = 1),
    c = runif(n, 0, 10)
  ) %>%
  rowwise() %>%
  mutate(
    time = min(t, c),
    status = as.numeric(time == t)
  )
glimpse(simdata)

```

rfrailty

*Frailties random generation***Description**

The frailty function for adding a simple random effects term to the linear predictor of a given survival regression model.

Usage

```

rfrailty(
  cluster,
  frailty = c("gamma", "gaussian", "ps"),
  sigma = 1,
  alpha = NULL,
  ...
)

```

Arguments

cluster	a vector determining the grouping of subjects (always converted to a factor object internally).
frailty	the frailty distribution; current implementation includes the gamma (default), lognormal and positive stable (ps) distributions.
sigma	standard deviation assumed for the frailty distribution; sigma = 1 by default; this value is ignored for positive stable (ps) distribution.

alpha stability parameter of the positive stable distribution; alpha must lie in (0,1) interval and an NA is return otherwise.

... further arguments passed to other methods.

Value

a vector with the generated frailties.

rinterval	<i>Random generation of type I and type II interval censored survival data</i>
-----------	--

Description

Function to generate a random sample of type I and type II interval censored survival data.

Usage

```
rinterval(time, tau, type = c("I", "II"), prob)
```

Arguments

time a numeric vector of survival times.

tau either a vector of censoring times (for type I interval-censored survival data) or time grid of scheduled visits (for type II interval censored survival data).

type type of interval-censored survival data (I or II).

prob = 0.5 attendance probability of scheduled visit; ignored when type = I.

Value

a data.frame containing the generated random sample.

rphreg	<i>Random generation from proportional hazards models</i>
--------	---

Description

Function to generate a random sample of survival data from proportional hazards models.

Usage

```
rphreg(u, formula, baseline, beta, dist = NULL, package = NULL, data, ...)
```

Arguments

<code>u</code>	a numeric vector of quantiles.
<code>formula</code>	formula specifying the linear predictors.
<code>baseline</code>	the name of the baseline survival distribution.
<code>beta</code>	vector of regression coefficients.
<code>dist</code>	an alternative way to specify the baseline survival distribution.
<code>package</code>	the name of the package where the assumed quantile function is implemented.
<code>data</code>	data frame containing the covariates used to generate the survival times.
<code>...</code>	further arguments passed to other methods.

Value

a numeric vector containing the generated random sample.

Examples

```
library(rsurv)
library(dplyr)
n <- 1000
simdata <- data.frame(
  age = rnorm(n),
  sex = sample(c("f", "m"), size = n, replace = TRUE)
) %>%
  mutate(
    t = rphreg(runif(n), ~ age+sex, beta = c(1, 2),
              dist = "weibull", shape = 1.5, scale = 1),
    c = runif(n, 0, 10)
  ) %>%
  rowwise() %>%
  mutate(
    time = min(t, c),
    status = as.numeric(time == t)
  )
glimpse(simdata)
```

 rporeg

Random generation from proportional odds models

Description

Function to generate a random sample of survival data from proportional odds models.

Usage

```
rporeg(u, formula, baseline, beta, dist = NULL, package = NULL, data, ...)
```

Arguments

<code>u</code>	a numeric vector of quantiles.
<code>formula</code>	formula specifying the linear predictors.
<code>baseline</code>	the name of the baseline survival distribution.
<code>beta</code>	vector of regression coefficients.
<code>dist</code>	an alternative way to specify the baseline survival distribution.
<code>package</code>	the name of the package where the assumed quantile function is implemented.
<code>data</code>	data frame containing the covariates used to generate the survival times.
<code>...</code>	further arguments passed to other methods.

Value

a numeric vector containing the generated random sample.

Examples

```
library(rsurv)
library(dplyr)
n <- 1000
simdata <- data.frame(
  age = rnorm(n),
  sex = sample(c("f", "m"), size = n, replace = TRUE)
) %>%
  mutate(
    t = rporeg(runif(n), ~ age+sex, beta = c(1, 2),
              dist = "weibull", shape = 1.5, scale = 1),
    c = runif(n, 0, 10)
  ) %>%
  rowwise() %>%
  mutate(
    time = min(t, c),
    status = as.numeric(time == t)
  )
glimpse(simdata)
```

 rypreg

Random generation from Yang and Prentice models

Description

Function to generate a random sample of survival data from Yang and Prentice models.

Usage

```
rypreg(u, formula, baseline, beta, phi, dist = NULL, package = NULL, data, ...)
```

Arguments

<code>u</code>	a numeric vector of quantiles.
<code>formula</code>	formula specifying the linear predictors.
<code>baseline</code>	the name of the baseline survival distribution.
<code>beta</code>	vector of short-term regression coefficients.
<code>phi</code>	vector of long-term regression coefficients.
<code>dist</code>	an alternative way to specify the baseline survival distribution.
<code>package</code>	the name of the package where the assumed quantile function is implemented.
<code>data</code>	data frame containing the covariates used to generate the survival times.
<code>...</code>	further arguments passed to other methods.

Value

a numeric vector containing the generated random sample.

Examples

```
library(rsurv)
library(dplyr)
n <- 1000
simdata <- data.frame(
  age = rnorm(n),
  sex = sample(c("f", "m"), size = n, replace = TRUE)
) %>%
  mutate(
    t = rypreg(runif(n), ~ age+sex, beta = c(1, 2), phi = c(-1, 2),
              dist = "weibull", shape = 1.5, scale = 1),
    c = runif(n, 0, 10)
  ) %>%
  rowwise() %>%
  mutate(
    time = min(t, c),
    status = as.numeric(time == t)
  )
glimpse(simdata)
```

Index

[bernoulli](#), [3](#)

[inv_pgf](#), [3](#)

[lp](#), [4](#)

[negbin](#), [5](#)

[qsurv](#), [5](#)

[raftreg](#), [6](#)

[rahreg](#), [7](#)

[rehreg](#), [8](#)

[rfrailty](#), [9](#)

[rinterval](#), [10](#)

[rphreg](#), [10](#)

[rporeg](#), [11](#)

[rsurv](#) ([rsurv-package](#)), [2](#)

[rsurv-package](#), [2](#)

[rypreg](#), [12](#)