

Package ‘event’

May 24, 2018

Version 1.1.1

Title Event History Procedures and Models

Depends R (>= 1.4), rmutil

Description Functions for setting up and analyzing event history data.

License GPL-2

URL <http://www.commanster.eu/rcode.html>

BugReports <https://github.com/swihart/event/issues>

Encoding UTF-8

LazyData true

LazyLoad true

NeedsCompilation yes

Author Bruce Swihart [cre, aut],
Jim Lindsey [aut] (Jim created this package, Bruce is maintaining the
CRAN version)

Maintainer Bruce Swihart <bruce.swihart@gmail.com>

Repository CRAN

Date/Publication 2018-05-24 18:04:28 UTC

R topics documented:

autointensity	2
bp	4
coxre	5
cprocess	6
ehr	7
event	9
hboxcox	11
hburr	12
hcauchy	12
hexp	13

hgamma	14
hgextval	15
hggamma	16
hglogis	17
hgweibull	18
hhjorth	19
hinvgauss	20
hlaplace	21
hlnorm	22
hlogis	22
hnorm	23
hpareto	24
hskewlaplace	25
hstudent	26
hweibull	27
ident	27
kalsurv	28
km	32
pbirth	33
plot.intensity	35
pp	36
tccov	37
tpast	38
ttime	38
tvcov	39

Index	41
--------------	-----------

autointensity	<i>Plot Autointensity Function of a Point Process</i>
---------------	---

Description

autointensity plots the autointensity function of a point process, including a solid horizontal line indicating the constant intensity of a Poisson process.

Usage

```
autointensity(times, window=NULL, maxlag=max(times),
ylab="Conditional probability", xlab="Lag",
main="Autointensity function", xlim=c(0,max(times)),
ylim=c(0,if(plotse)max(se1)else max(z$density)),
lty=1, plot=TRUE, plotse=TRUE, add=FALSE, ...)
```

Arguments

<code>times</code>	Vector of times between events.
<code>window</code>	Width of grouping interval.
<code>maxlag</code>	Maximum lag to be calculated, by default the maximum interevent time.
<code>plot</code>	If FALSE, values are returned but the function is not plotted.
<code>plotse</code>	If TRUE, plots pointwise two-standard error bands around the curve.
<code>add</code>	If TRUE, add curve to an existing plot.
<code>ylab</code>	Plotting control options.
<code>xlab</code>	Plotting control options.
<code>main</code>	Plotting control options.
<code>xlim</code>	Plotting control options.
<code>ylim</code>	Plotting control options.
<code>lty</code>	Plotting control options.
<code>...</code>	Plotting control options.

Value

A list containing the coordinates of the plotted function and the standard error bands.

Author(s)

J.K. Lindsey

References

Guttorp, P. (1995) Stochastic Modeling of Scientific Data. Chapman & Hall, pp. 229, 238-240.

See Also

[cprocess](#).

Examples

```
times <- rgamma(100,2,scale=4)
autointensity(times, window=3)
```

bp *Create a Vector of Cumulative Numbers of Previous Events for a Point Process (Birth Processes)*

Description

bp creates a vector of length $\text{sum}(y)$ of cumulative numbers of previous events for each individual for use in fitting birth processes with [ehr](#). Add one if the process starts at an event.

Usage

```
bp(y, id, censor=1)
```

Arguments

y	Vector of times.
id	Vector of corresponding individual identifiers for who had which sequence of times.
censor	Vector of censoring indicators.

Value

bp creates a vector of length $\text{sum}(y)$ of cumulative numbers of previous events for each individual for use in fitting birth processes with [ehr](#). Add one if the process starts at an event.

Author(s)

J.K. Lindsey

See Also

[ehr](#), [ident](#), [pp](#), [tccov](#), [tpast](#), [ttime](#), [tvcov](#).

Examples

```
y <- c(5,3,2,4)
i <- c(1,1,2,2)
birth <- bp(y, i)
birth
```

 coxre

Cox Proportional Hazards Model with Random Effect

Description

coxre fits a Cox proportional hazards model to event history data using a gamma distribution random effect. The parameter, gamma, is the variance of this mixing distribution.

If a matrix of response times is supplied, the model can be stratified by columns, i.e. a different intensity function is fitted for each column. To fit identical intensity functions to all response types, give the times as a vector.

Usage

```
coxre(response, censor, nest=NULL, cov=NULL, stratified=FALSE,
       cumul=FALSE, estimate=1, iter=10, print.level=0, ndigit=10,
       gradtol=0.00001, steptol=0.00001, iterlim=100, fscale=1,
       tysize=abs(estimate), stepmax=estimate)
```

Arguments

response	Vector or matrix of times to events, with one column per type of response (or subunit).
censor	Corresponding vector or matrix of censoring indicators. If NULL all values are set to one.
nest	Vector indicating to which unit each observation belongs.
cov	One covariate
stratified	If TRUE, a model stratified on type of response (the columns of response) is fitted instead of proportional intensities.
cumul	Set to TRUE if response times are from a common origin instead of times to (or between) events.
estimate	Initial estimate of the frailty parameter.
iter	Maximum number of iterations allowed for the inner EM loop.
print.level	nlm control options.
ndigit	nlm control options.
gradtol	nlm control options.
steptol	nlm control options.
iterlim	nlm control options.
fscale	nlm control options.
tysize	nlm control options.
stepmax	nlm control options.

Author(s)

D.G. Clayton and J.K. Lindsey

References

Clayton, D. (1987) The analysis of event history data: a review of progress and outstanding problems. *Statistics in Medicine* 7: 819-841

See Also

[kalsurv](#).

Examples

```
# 11 individuals, each with 5 responses
y <- matrix(c(51,36,50,35,42,
27,20,26,17,27,
37,22,41,37,30,
42,36,32,34,27,
27,18,33,14,29,
43,32,43,35,40,
41,22,36,25,38,
38,21,31,20,16,
36,23,27,25,28,
26,31,31,32,36,
29,20,25,26,25),ncol=5,byrow=TRUE)
# Different intensity functions
coxre(response=y, censor=matrix(rep(1,55),ncol=5), nest=1:11,
est=0.7, stratified=TRUE)
# Proportional intensity functions for the five responses
coxre(response=y, censor=matrix(rep(1,55),ncol=5), nest=1:11,
est=0.7, stratified=FALSE)
# Identical intensity functions
coxre(response=as.vector(t(y)), censor=rep(1,55),
nest=rep(1:11,rep(5,11)), est=0.7)
```

cprocess

Plot Counting Process Data

Description

cprocess plots the cumulative number of events (the counting process) over time for each vector in the list. One or both of times and events can be supplied.

Usage

```
cprocess(times=NULL, events=NULL, number=FALSE, lty=NULL, xlim=NULL,
ylim=NULL, xlab="Time", ylab="Counts", ...)
```

Arguments

times	Vector of times between events, a list of such vectors, or a repeated object.
events	Vector of counts of events, a list of such vectors, or a repeated object.
number	If TRUE, the counting processes are numbered consecutively on the graph.
ylab	Plotting control options.
xlab	Plotting control options.
xlim	Plotting control options.
ylim	Plotting control options.
lty	Plotting control options.
...	Plotting control options.

Author(s)

J.K. Lindsey

See Also

[read.list](#), [rmna](#).

Examples

```
times <- rgamma(20,2,scale=4)
cprocess(times)
```

 ehr

Regression Models for Event History Intensity Functions

Description

ehr fits an intensity function to event histories, where point is produced by `point <- pp(y)` and lambda is the user-defined log intensity function.

Nonlinear regression models for lambda can be supplied as formulae where parameters are unknowns. Factor variables cannot be used and parameters must be scalars. (See [finterp](#).)

Usage

```
ehr(point, lambda=NULL, linear=NULL, plambda=NULL, delta=1,
     envir=parent.frame(), print.level=0, tysize=rep(1,length(plambda)),
     ndigit=10, gradtol=0.00001, iterlim=100, fscale=1,
     stepmax=max(10*sqrt(plambda**plambda),10), steptol=0.0004)
```

Arguments

<code>point</code>	A point process vector produced by <code>pp</code> .
<code>lambda</code>	User-specified function of p , and possibly linear, giving the regression equation for the intensity or a formula beginning with <code>~</code> , specifying either a linear regression function in the Wilkinson and Rogers notation or a general function with named unknown parameters. The function may contain a linear part that must simply be given the name, <code>linear</code> , in the function. If no function is supplied, the intensity is taken to be constant (a homogeneous Poisson process).
<code>linear</code>	A formula beginning with <code>~</code> specifying the linear part of the regression function.
<code>plambda</code>	Vector of initial parameter estimates. If <code>lambda</code> is a formula with unknown parameters, their estimates must be supplied either in their order of appearance in the expression or in a named list.
<code>delta</code>	If any time intervals are different from unity, a vector of time intervals.
<code>envir</code>	Environment in which model formulae are to be interpreted or a data object of class, <code>repeated</code> , <code>tccov</code> , or <code>tvcov</code> . If <code>point</code> has class <code>repeated</code> , it is used as the environment.
<code>print.level</code>	<code>nlm</code> control options.
<code>ndigit</code>	<code>nlm</code> control options.
<code>gradtol</code>	<code>nlm</code> control options.
<code>steptol</code>	<code>nlm</code> control options.
<code>iterlim</code>	<code>nlm</code> control options.
<code>fscale</code>	<code>nlm</code> control options.
<code>tysize</code>	<code>nlm</code> control options.
<code>stepmax</code>	<code>nlm</code> control options.

Author(s)

J.K. Lindsey

References

Lindsey, J.K. (1995) Fitting parametric counting processes by using log linear models. *Journal of the Royal Statistical Society C44*, 201-212.

See Also

`bp`, `finterp`, `ident`, `pp`, `tccov`, `tpast`, `ttime`, `tvcov`.

Examples

```
y <- c(5,3,2,4)
# event indicator
py <- pp(y)
# time since previous event
ptime <- tpast(y)
```



```

# individual ID
i <- c(1,1,2,2)
id <- ident(y, i)
# times and corresponding covariate values
tx <- c(2,3,1,2,2,2,2)
x <- c(1,2,2,1,2,2,1)
zcov <- tvcov(y, x, tx)
# Poisson process
ehr(py, plambda=1)
# Weibull process
lambda1 <- function(p) p[1]+p[2]*log(ptime)
ehr(py, lambda=lambda1, plambda=c(1,1))
# or
ehr(py, lambda=~log(ptime), plambda=c(1,1))
# or
ehr(py, lambda=~b0+b1*log(ptime), plambda=list(b0=1,b1=1))
# Poisson process with time-varying covariate
lambda2 <- function(p) p[1]+p[2]*zcov
ehr(py, lambda=lambda2, plambda=c(1,1))
# or
ehr(py, lambda=~zcov, plambda=c(1,1))
# or
ehr(py, lambda=~c0+c1*zcov, plambda=list(c0=1,c1=1))
# Weibull process with time-varying covariate
lambda3 <- function(p) p[1]+p[2]*log(ptime)+p[3]*zcov
ehr(py, lambda=lambda3, plambda=c(1,1,1))
# or
ehr(py, lambda=~log(ptime)+zcov, plambda=c(1,1,1))
# or
ehr(py, lambda=~c0+b1*log(ptime)+c1*zcov, plambda=list(c0=1,c1=1,b1=1))
# gamma process with time-varying covariate
lambda4 <- function(p) hgamma(ptime, p[1], exp(p[2]+p[3]*zcov))
ehr(py, lambda=lambda4, plambda=c(1,1,1))
# or
ehr(py, lambda=~hgamma(ptime, b1, exp(c0+c1*zcov)),
plambda=list(c0=1,c1=1,b1=1))
# or
lambda5 <- function(p, linear) hgamma(ptime, p[1], exp(linear))
ehr(py, lambda=lambda5, linear=~zcov, plambda=c(1,1,1))

```

Description

[autointensity](#) Plot Autointensity Function of a Point Process

[bp](#) Create a Vector of Cumulative Numbers of Previous Events for a Point Process

[coxre](#) Cox Proportional Hazards Model with Random Effect

[cprocess](#) Counting Process Plot

[ehr](#) Fit an Intensity Function to Event Histories

[hboxcox](#) Log Hazard Function for a Box-Cox Process

[hburr](#) Log Hazard Function for a Burr Process

[hcauchy](#) Log Hazard Function for a Cauchy Process

[hexp](#) Log Hazard Function for an Exponential (Poisson) Process

[hgamma](#) Log Hazard Function for a Gamma Process

[hgextval](#) Log Hazard Function for an Extreme Value Process

[hggamma](#) Log Hazard Function for a Generalized Gamma Process

[hglogis](#) Log Hazard Function for a Generalized Logistic Process

[hgweibull](#) Log Hazard Function for a Generalized Weibull Process

[hhjorth](#) Log Hazard Function for a Hjorth Process

[hinvgauss](#) Log Hazard Function for a Inverse Gauss Process

[hlaplace](#) Log Hazard Function for a Laplace Process

[hlnorm](#) Log Hazard Function for a Log Normal Process

[hlogis](#) Log Hazard Function for a Logistic Process

[hnorm](#) Log Hazard Function for a Normal Process

[hpareto](#) Log Hazard Function for a Pareto Process

[hskewlaplace](#) Log Hazard Function for a Skew Laplace Process

[hstudent](#) Log Hazard Function for a Student Process

[hweibull](#) Log Hazard Function for a Weibull Process

[ident](#) Create an Individual Identification Vector for a Point Process

[kalsurv](#) Generalized Repeated Measurements Models for Event Histories

[km](#) Kaplan-Meier Survival Curves

[pbirth](#) Fit Overdispersed Count Data as a Birth Process

[pp](#) Create a Point Process Vector from Times between Events

[read.list](#) Read a List of Matrices of Unbalanced Repeated Measurements from a File

[read.surv](#) Read a List of Vectors of Event Histories from a File

[tccov](#) Create a Vector of Time-constant Covariates for a Point Process

[tpast](#) Create a Vector of Times Past since Previous Events for a Point Process

[ttime](#) Create a Vector of Total Time Elapsed for each Individual for a Point Process

[tvcov](#) Create a Vector of Time-varying Covariates for a Point Process

Description

This function provides information about the Box-Cox distribution with location parameter equal to m , dispersion equal to s , and power transformation equal to f : log hazard. (See 'rmutil' for the d/p/q/r boxcox functions density, cumulative distribution, quantiles, and random generation).

The Box-Cox distribution has density

$$f(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-((y^\nu/\nu - \mu)^2/(2\sigma^2)))/(1 - I(\nu < 0) - \text{sign}(\nu) * \text{pnorm}(0, \mu, \text{sqr}t(\sigma)))$$

where μ is the location parameter of the distribution, σ is the dispersion, ν is the family parameter, $I()$ is the indicator function, and $y > 0$.

$\nu = 1$ gives a truncated normal distribution.

Usage

```
hboxcox(y, m, s, f)
```

Arguments

y	vector of responses.
m	vector of location parameters.
s	vector of dispersion parameters.
f	vector of power parameters.

Author(s)

J.K. Lindsey

See Also

[dnorm](#) for the normal or Gaussian distribution.

Examples

```
hboxcox(2, 5, 5, 2)
```

 hburr

Log Hazard Function for a Burr Process

Description

These functions provide information about the Burr distribution with location parameter equal to m , dispersion equal to s , and family parameter equal to f : log hazard. (See 'rmutil' for the d/p/q/r boxcox functions density, cumulative distribution, quantiles, and random generation).

The Burr distribution has density

$$f(y) = \frac{\nu\sigma(y/\mu)^{\sigma-1}}{\mu(1+(y/\mu)^\sigma)^{\nu+1}}$$

where μ is the location parameter of the distribution, σ is the dispersion, and ν is the family parameter.

Usage

hburr(y, m, s, f)

Arguments

y	vector of responses.
m	vector of location parameters.
s	vector of dispersion parameters.
f	vector of family parameters.

Author(s)

J.K. Lindsey

Examples

hburr(2, 5, 1, 2)

 hcauchy

Log Hazard Function for a Cauchy Process

Description

Log Hazard Function for a Cauchy Process

Usage

hcauchy(y, m, s)

Arguments

y	Vector of times.
m	Location parameter.
s	Dispersion parameter.

Value

hcauchy returns the log hazard function for a Cauchy process with the given parameter values.

Author(s)

J.K. Lindsey

See Also

[hboxcox](#), [hburr](#), [hexp](#), [hgextval](#), [hgamma](#), [hggamma](#), [hhjorth](#), [hinvgauss](#), [hlaplace](#), [hlnorm](#), [hlogis](#), [hglogis](#), [hnorm](#), [hstudent](#), [hweibull](#), [hgweibull](#).

Examples

```
hcauchy(1:10, 3, 2)
```

hexp

Log Hazard Function for a Poisson Process

Description

Log Hazard Function for a Poisson Process

Usage

```
hexp(y, rate)
```

Arguments

y	Vector of times.
rate	Vector of rates.

Value

hexp returns the log hazard function for a Poisson process with the given parameter value.

Author(s)

J.K. Lindsey

See Also

[hboxcox](#), [h Burr](#), [hexp](#), [hgextval](#), [hgamma](#), [hggamma](#), [hhjorth](#), [hinvgauss](#), [hlaplace](#), [hlnorm](#), [hlogis](#), [hglogis](#), [hnorm](#), [hstudent](#), [hweibull](#), [hgweibull](#).

Examples

```
hexp(1:10, 3)
```

hgamma

Log Hazard Function for a Gamma Process

Description

Log Hazard Function for a Gamma Process

Usage

```
hgamma(y, shape, rate=1, scale=1/rate)
```

Arguments

y	Vector of times.
shape	Shape parameter.
rate	Rate parameter.
scale	Scale parameter.

Value

hgamma returns the log hazard function for a gamma process with the given parameter values.

Author(s)

J.K. Lindsey

See Also

[hboxcox](#), [h Burr](#), [hexp](#), [hgextval](#), [hcauchy](#), [hggamma](#), [hhjorth](#), [hinvgauss](#), [hlaplace](#), [hlnorm](#), [hlogis](#), [hglogis](#), [hnorm](#), [hstudent](#), [hweibull](#), [hgweibull](#).

Examples

```
hgamma(1:10, 3, 2)
```

Description

These functions provide information about the generalized extreme value distribution with location parameter equal to m , dispersion equal to s , and family parameter equal to f :log hazard. (See 'rmutil' for the d/p/q/r boxcox functions density, cumulative distribution, quantiles, and random generation).

The generalized extreme value distribution has density

$$f(y) = y^{\nu-1} \exp(y^\nu/\nu) \frac{\sigma}{\mu \mu^{\sigma-1} / (1 - I(\nu > 0) + \text{sign}(\nu) \exp(-\mu^{-\sigma}))} \exp(-(\exp(y^\nu \nu)/\mu)^\sigma)$$

where μ is the location parameter of the distribution, σ is the dispersion, ν is the family parameter, $I()$ is the indicator function, and $y > 0$.

$\nu = 1$ a truncated extreme value distribution.

Usage

```
hgextval(y, s, m, f)
```

Arguments

y	vector of responses.
m	vector of location parameters.
s	vector of dispersion parameters.
f	vector of family parameters.

Author(s)

J.K. Lindsey

See Also

[dweibull](#) for the Weibull distribution.

Examples

```
hgextval(1, 2, 1, 2)
```

Description

These functions provide information about the generalized gamma distribution with scale parameter equal to *m*, shape equal to *s*, and family parameter equal to *f*: log hazard. (See ‘*rmutil*’ for the *d/p/q/r* boxcox functions density, cumulative distribution, quantiles, and random generation).

The generalized gamma distribution has density

$$f(y) = \frac{\nu y^{\nu-1}}{(\mu/\sigma)^{\nu\sigma} \text{Gamma}(\sigma)} y^{\nu(\sigma-1)} \exp(-(y\sigma/\mu)^\nu)$$

where μ is the scale parameter of the distribution, σ is the shape, and ν is the family parameter.

$\nu = 1$ yields a gamma distribution, $\sigma = 1$ a Weibull distribution, and $\sigma = \infty$ a log normal distribution.

Usage

```
hggamma(y, s, m, f)
```

Arguments

<i>y</i>	vector of responses.
<i>m</i>	vector of location parameters.
<i>s</i>	vector of dispersion parameters.
<i>f</i>	vector of family parameters.

Author(s)

J.K. Lindsey

See Also

[dgamma](#) for the gamma distribution, [dweibull](#) for the Weibull distribution, [dlnorm](#) for the log normal distribution.

Examples

```
hggamma(2, 5, 4, 2)
```


Description

These functions provide information about the generalized logistic distribution with location parameter equal to m , dispersion equal to s , and family parameter equal to f : log hazard. (See 'rmutil' for the d/p/q/r boxcox functions density, cumulative distribution, quantiles, and random generation).

The generalized logistic distribution has density

$$f(y) = \frac{\nu\sqrt{3}\exp(-\sqrt{3}(y-\mu)/(\sigma\pi))}{\sigma\pi(1+\exp(-\sqrt{3}(y-\mu)/(\sigma\pi)))^{\nu+1}}$$

where μ is the location parameter of the distribution, σ is the dispersion, and ν is the family parameter.

$\nu = 1$ gives a logistic distribution.

Usage

```
hglogis(y, m, s, f)
```

Arguments

y	vector of responses.
m	vector of location parameters.
s	vector of dispersion parameters.
f	vector of family parameters.

Author(s)

J.K. Lindsey

See Also

[dlogis](#) for the logistic distribution.

Examples

```
hglogis(5, 5, 1, 2)
```

Description

These functions provide information about the generalized Weibull distribution, also called the exponentiated Weibull, with scale parameter equal to m , shape equal to s , and family parameter equal to f : log hazard. (See 'rmutil' for the d/p/q/r boxcox functions density, cumulative distribution, quantiles, and random generation).

The generalized Weibull distribution has density

$$f(y) = \frac{\sigma \nu y^{\sigma-1} (1 - \exp(-(y/\mu)^\sigma))^{\nu-1} \exp(-(y/\mu)^\sigma)}{\mu^\sigma}$$

where μ is the scale parameter of the distribution, σ is the shape, and ν is the family parameter.

$\nu = 1$ gives a Weibull distribution, for $\sigma = 1$, $\nu < 0$ a generalized F distribution, and for $\sigma > 0$, $\nu \leq 0$ a Burr type XII distribution.

Usage

```
hgweibull(y, s, m, f)
```

Arguments

y	vector of responses.
m	vector of location parameters.
s	vector of dispersion parameters.
f	vector of family parameters.

Author(s)

J.K. Lindsey

See Also

[dweibull](#) for the Weibull distribution, [df](#) for the F distribution, [dburr](#) for the Burr distribution.

Examples

```
hgweibull(5, 1, 3, 2)
```

`hhjorth`*Log Hazard Function for a Hjorth Process*

Description

These functions provide information about the Hjorth distribution with location parameter equal to m , dispersion equal to s , and family parameter equal to f : log hazard. (See 'rmutil' for the d/p/q/r boxcox functions density, cumulative distribution, quantiles, and random generation).

The Hjorth distribution has density

$$f(y) = (1 + \sigma y)^{-\nu/\sigma} \exp(-(y/\mu)^2/2) \left(\frac{y}{\mu^2} + \frac{\nu}{1 + \sigma y} \right)$$

where μ is the location parameter of the distribution, σ is the dispersion, and ν is the family parameter.

Usage

```
hhjorth(y, m, s, f)
```

Arguments

<code>y</code>	vector of responses.
<code>m</code>	vector of location parameters.
<code>s</code>	vector of dispersion parameters.
<code>f</code>	vector of family parameters.

Author(s)

J.K. Lindsey

Examples

```
hhjorth(5, 5, 5, 2)
```

`hinvgauss`*Log Hazard Function for a Inverse Gauss Process*

Description

These functions provide information about the inverse Gaussian distribution with mean equal to m and dispersion equal to s : log hazard. (See 'rmutil' for the d/p/q/r boxcox functions density, cumulative distribution, quantiles, and random generation).

The inverse Gaussian distribution has density

$$f(y) = \frac{1}{\sqrt{2\pi\sigma y^3}} e^{-(y-\mu)^2/(2y\sigma m^2)}$$

where μ is the mean of the distribution and σ is the dispersion.

Usage

```
hinvgauss(y, m, s)
```

Arguments

<code>y</code>	vector of responses.
<code>m</code>	vector of means.
<code>s</code>	vector of dispersion parameters.

Author(s)

J.K. Lindsey

See Also

[dnorm](#) for the normal distribution and [dlnorm](#) for the Lognormal distribution.

Examples

```
hinvgauss(5, 5, 1)
```

`hlaplace`*Log Hazard Function for a Laplace Process*

Description

These functions provide information about the Laplace distribution with location parameter equal to m and dispersion equal to s : log hazard. (See 'rmutil' for the d/p/q/r boxcox functions density, cumulative distribution, quantiles, and random generation).

The Laplace distribution has density

$$f(y) = \frac{\exp(-abs(y - \mu)/\sigma)}{(2\sigma)}$$

where μ is the location parameter of the distribution and σ is the dispersion.

Usage

```
hlaplace(y, m=0, s=1)
```

Arguments

<code>y</code>	vector of responses.
<code>m</code>	vector of location parameters.
<code>s</code>	vector of dispersion parameters.

Author(s)

J.K. Lindsey

See Also

[dexp](#) for the exponential distribution and [dcauchy](#) for the Cauchy distribution.

Examples

```
hlaplace(5, 2, 1)
```

`hlnorm`*Log Hazard Function for a Log Normal Process*

Description

Log Hazard Function for a Log Normal Process

Usage

```
hlnorm(y, m, s)
```

Arguments

<code>y</code>	Vector of times.
<code>m</code>	Mean parameter.
<code>s</code>	Variance parameter.

Value

`hlnorm` returns the log hazard function for a log normal process with the given parameter values.

Author(s)

J.K. Lindsey

See Also

[hboxcox](#), [hburr](#), [hexp](#), [hgextval](#), [hcauchy](#), [hgamma](#), [hggamma](#), [hhjorth](#), [hinvgauss](#), [hlaplace](#), [hlogis](#), [hglogis](#), [hnorm](#), [hstudent](#), [hweibull](#), [hgweibull](#).

Examples

```
hlnorm(1:10, 3, 2)
```

`hlogis`*Log Hazard Function for a Logistic Process*

Description

Log Hazard Function for a Logistic Process

Usage

```
hlogis(y, m, s)
```

Arguments

y	Vector of times.
m	Location parameter.
s	Scale parameter.

Value

hlogis returns the log hazard function for a logistic process with the given parameter values.

Author(s)

J.K. Lindsey

See Also

[hboxcox](#), [h Burr](#), [hexp](#), [hgextval](#), [hcauchy](#), [hgamma](#), [hggamma](#), [hhjorth](#), [hinvgauss](#), [hlaplace](#), [hlnorm](#), [hglogis](#), [hnorm](#), [hstudent](#), [hweibull](#), [hgweibull](#).

Examples

```
hlogis(1:10, 3, 2)
```

hnorm

Log Hazard Function for a Normal Process

Description

Log Hazard Function for a Normal Process

Usage

```
hnorm(y, m, s)
```

Arguments

y	Vector of times.
m	Mean parameter.
s	Variance parameter.

Value

hnorm returns the log hazard function for a normal process with the given parameter values.

Author(s)

J.K. Lindsey

See Also

[hboxcox](#), [hburr](#), [hexp](#), [hgextval](#), [hcauchy](#), [hgamma](#), [hggamma](#), [hhjorth](#), [hinvgauss](#), [hlaplace](#), [hlogis](#), [hglogis](#), [hlnorm](#), [hstudent](#), [hweibull](#), [hgweibull](#).

Examples

```
hnorm(1:10, 3, 2)
```

hpareto

Log Hazard Function for a Pareto Process

Description

These functions provide information about the Pareto distribution with location parameter equal to m and dispersion equal to s : log hazard. (See 'rmutil' for the d/p/q/r boxcox functions density, cumulative distribution, quantiles, and random generation).

The Pareto distribution has density

$$f(y) = \frac{\sigma}{\mu(\sigma - 1)(1 + y/(\mu(\sigma - 1)))^{\sigma+1}}$$

where μ is the mean parameter of the distribution and σ is the dispersion.

This distribution can be obtained as a mixture distribution from the exponential distribution using a gamma mixing distribution.

Usage

```
hpareto(y, m, s)
```

Arguments

y	vector of responses.
m	vector of location parameters.
s	vector of dispersion parameters.

Author(s)

J.K. Lindsey

See Also

[dexp](#) for the exponential distribution.

Examples

```
hpareto(5, 2, 2)
```


Description

These functions provide information about the skew Laplace distribution with location parameter equal to m , dispersion equal to s , and skew equal to f : log hazard. (See 'rmutil' for the d/p/q/r boxcox functions density, cumulative distribution, quantiles, and random generation). For $f=1$, this is an ordinary (symmetric) Laplace distribution.

The skew Laplace distribution has density

$$f(y) = \frac{\nu \exp(-\nu(y - \mu)/\sigma)}{(1 + \nu^2)\sigma}$$

if $y \geq \mu$ and else

$$f(y) = \frac{\nu \exp((y - \mu)/(\nu\sigma))}{(1 + \nu^2)\sigma}$$

where μ is the location parameter of the distribution, σ is the dispersion, and ν is the skew.

The mean is given by $\mu + \frac{\sigma(1-\nu^2)}{\sqrt{2}\nu}$ and the variance by $\frac{\sigma^2(1+\nu^4)}{2\nu^2}$.

Note that this parametrization of the skew (family) parameter is different than that used for the multivariate skew Laplace distribution in 'growth::elliptic'.

Usage

```
hskewlaplace(y, m=0, s=1, f=1)
```

Arguments

y	vector of responses.
m	vector of location parameters.
s	vector of dispersion parameters.
f	vector of skew parameters.

Author(s)

J.K. Lindsey

See Also

[dexp](#) for the exponential distribution, [dcauchy](#) for the Cauchy distribution, and [dlaplace](#) for the Laplace distribution.

Examples

```
hskewlaplace(5, 2, 1, 0.5)
```

hstudent	<i>Log Hazard Function for a Student t Process</i>
----------	--

Description

Log Hazard Function for a Student t Process

Usage

```
hstudent(y, m, s, f)
```

Arguments

y	Vector of times.
m	Location parameter.
s	Scale parameter.
f	Degrees of freedom.

Value

hstudent returns the log hazard function for a Student t process with the given parameter values.

Author(s)

J.K. Lindsey

See Also

[hboxcox](#), [hburr](#), [hexp](#), [hgextval](#), [hcauchy](#), [hgamma](#), [hggamma](#), [hhjorth](#), [hinvgauss](#), [hlaplace](#), [hlogis](#), [hglogis](#), [hnorm](#), [hnorm](#), [hweibull](#), [hgweibull](#).

Examples

```
hstudent(1:10, 3, 2, 5)
```

`hweibull`*Log Hazard Function for a Weibull Process*

Description

Log Hazard Function for a Weibull Process

Usage

```
hweibull(y, s, m)
```

Arguments

<code>y</code>	Vector of times.
<code>s</code>	Shape parameter.
<code>m</code>	Scale parameter.

Value

`hweibull` returns the log hazard function for a Weibull process with the given parameter values.

Author(s)

J.K. Lindsey

See Also

[hboxcox](#), [hburr](#), [hexp](#), [hgextval](#), [hcauchy](#), [hgamma](#), [hggamma](#), [hhjorth](#), [hinvgauss](#), [hlaplace](#), [hlogis](#), [hglogis](#), [hlnorm](#), [hnorm](#), [hstudent](#), [hgweibull](#).

Examples

```
hweibull(1:10, 1.5, 2)
```

`ident`*Create an Individual Identification Vector for a Point Process*

Description

`ident` creates a vector of length `sum(y)` by repeating the values of individual identifiers for the times for use with [ehr](#).

Usage

```
ident(y, id)
```

Arguments

<code>y</code>	Vector of times.
<code>id</code>	Vector of corresponding individual identifiers for who had which sequence of times.

Value

`ident` creates a vector of length `sum(y)` by repeating the values of individual identifiers for the times for use with [ehr](#).

Author(s)

J.K. Lindsey

See Also

[bp](#), [ehr](#), [pp](#), [tccov](#), [tpast](#), [ttime](#), [tvcov](#).

Examples

```
y <- c(5,3,2,4)
i <- c(1,1,2,2)
id <- ident(y, i)
id
```

kalsurv

Repeated Events Models with Frailty or Serial Dependence

Description

`kalsurv` is designed to handle event history models with time-varying covariates. The distributions have two extra parameters as compared to the functions specified by `intensity` and are generally longer tailed than those distributions. Dependence of inter-event times can be through gamma frailties (a type of random effect), with or without autoregression, or several kinds of serial dependence by updating, as in Kalman filtering.

By default, a gamma mixture of the distribution specified in `intensity` is used, as the conditional distribution in the serial dependence models, and as a symmetric multivariate (random effect) model for frailty dependence. For example, with a Weibull `intensity` and frailty dependence, this yields a multivariate Burr distribution and with Markov or serial dependence, univariate Burr conditional distributions.

If a value for `pfamily` is used, the gamma mixture is replaced by a power variance family mixture.

Nonlinear regression models can be supplied as formulae where parameters are unknowns in which case factor variables cannot be used and parameters must be scalars. (See [finterp](#).)

Marginal and individual profiles can be plotted using [mprofile](#) and [iprofile](#) and residuals with [plot.residuals](#).

Usage

```
kalsurv(response, intensity="exponential", distribution="Pareto",
depend="independence", update="Markov", mu=NULL, shape=NULL,
renewal=TRUE, density=FALSE, censor=NULL, delta=NULL, ccov=NULL,
tvcov=NULL, preg=NULL, ptvc=NULL, pbirth=NULL,
pintercept=NULL, pshape=NULL, pinitial=1, pdepend=NULL,
pfamily=NULL, envir=parent.frame(), print.level=0,
ndigit=10, gradtol=0.00001, steptol=0.00001, iterlim=100,
fscale=1, tysize=abs(p), stepmax=10*sqrt(p%*%p))
```

Arguments

response	A list of vectors with times between events for each individual, one matrix or dataframe of such times if all individuals have the same number of events, or an object of class, response (created by <code>restovec</code>) or repeated (created by <code>rmna</code> or <code>lvna</code>). If the repeated data object contains more than one response variable, give that object in <code>envir</code> and give the name of the response variable to be used here.
intensity	The form of intensity function to be put in the distribution given by <code>dist</code> . Choices are exponential, Weibull, gamma, log normal, log logistic, log Cauchy, log Student, and <code>gen(eralized)</code> logistic.
distribution	The outer distribution. Choices are Pareto, gamma, and Weibull.
depend	Type of dependence. Choices are independence, frailty, and serial.
update	Type of update for serial dependence. Choices are Markov, elapsed Markov, serial, event, cumulated, count, and kalman. With frailty dependence, weighting by length of observation time may be specified by setting <code>update</code> to time.
mu	A regression function for the location parameter or a formula beginning with <code>~</code> , specifying either a linear regression function in the Wilkinson and Rogers notation or a general function with named unknown parameters. Give the initial estimates in <code>preg</code> if there are no time-varying covariates and in <code>ptvc</code> if there are.
shape	A regression function for the shape parameter or a formula beginning with <code>~</code> , specifying either a linear regression function in the Wilkinson and Rogers notation or a general function with named unknown parameters. It must yield one value per observation.
renewal	If TRUE, a renewal process is modelled, with time reinitialized after each event. Otherwise, time is cumulated from the origin of observations.
density	If TRUE, the density of the function specified in <code>intensity</code> is used instead of the intensity.
censor	A vector of the same length as the number of individuals containing a binary indicator, with a one indicating that the last time period in the series terminated with an event and zero that it was censored. For independence and frailty models, where response is matrix, censor may also be a matrix of the same size. Ignored if response has class, response or repeated.

<code>delta</code>	Scalar or vector giving the unit of measurement for each response value, set to unity by default. For example, if a response is measured to two decimals, <code>delta=0.01</code> . If the response has been pretransformed, this must be multiplied by the Jacobian. This transformation cannot contain unknown parameters. For example, with a log transformation, <code>delta=1/y</code> . (The delta values for the censored response are ignored.) Ignored if response has class, response or repeated.
<code>ccov</code>	A vector or matrix containing time-constant baseline covariates with one entry per individual, a model formula using vectors of the same size, or an object of class, <code>tccov</code> (created by <code>tcctomat</code>). If response has class, repeated, the covariates must be supplied as a Wilkinson and Rogers formula unless none are to be used or <code>mu</code> is given.
<code>tvcov</code>	A list of matrices with time-varying covariate values, observed at the event times in response, for each individual (one column per variable), one matrix or dataframe of such covariate values, or an object of class, <code>tvcov</code> (created by <code>tvctomat</code>). If response has class, repeated, the covariates must be supplied as a Wilkinson and Rogers formula unless none are to be used or <code>mu</code> is given.
<code>preg</code>	Initial parameter estimates for the regression model: intercept plus one for each covariate in <code>ccov</code> . If <code>mu</code> is a formula or function, the parameter estimates must be given here only if there are no time-varying covariates. If <code>mu</code> is a formula with unknown parameters, their estimates must be supplied either in their order of appearance in the expression or in a named list.
<code>ptvc</code>	Initial parameter estimates for the coefficients of the time-varying covariates, as many as in <code>tvcov</code> . If <code>mu</code> is a formula or function, the parameter estimates must be given here if there are time-varying covariates present.
<code>pbirth</code>	If supplied, this is the initial estimate for the coefficient of the birth model.
<code>pintercept</code>	The initial estimate of the intercept for the generalized logistic intensity.
<code>pshape</code>	An initial estimate for the shape parameter of the intensity (except exponential intensity). If shape is a function or formula, the corresponding initial estimates. If shape is a formula with unknown parameters, their estimates must be supplied either in their order of appearance in the expression or in a named list.
<code>pinitial</code>	An initial estimate for the initial parameter. In frailty dependence, this is the frailty parameter.
<code>pdepend</code>	An initial estimate for the serial dependence parameter. For frailty dependence, if a value is given here, an autoregression is fitted as well as the frailty.
<code>pfamily</code>	An optional initial estimate for the second parameter of a two-parameter power variance family mixture instead of the default gamma mixture. This yields a gamma mixture as <code>family -> 0</code> , an inverse Gauss mixture for <code>family = 0.5</code> , and a compound distribution of a Poisson-distributed number of gamma distributions for $-1 < \text{family} < 0$.
<code>envir</code>	Environment in which model formulae are to be interpreted or a data object of class, repeated, <code>tccov</code> , or <code>tvcov</code> ; the name of the response variable should be given in response. If response has class repeated, it is used as the environment.
<code>print.level</code>	nlm control options.
<code>ndigit</code>	nlm control options.

gradtol	nlm control options.
steptol	nlm control options.
iterlim	nlm control options.
fscale	nlm control options.
tysize	nlm control options.
stepmax	nlm control options.

Value

A list of classes kalsurv and recursive is returned.

Author(s)

J.K. Lindsey

See Also

[coxre](#), [finterp](#), [gettvc](#),
[iprofile](#),
[mprofile](#),
[read.list](#), [restovec](#), [rmna](#), [tcctomat](#), [tvctomat](#).

Examples

```
treat <- c(0,0,1,1)
tr <- tcctomat(treat)
cens <- matrix(rbinom(20,1,0.9),ncol=5)
times <- # matrix(rweibull(20,2,1+3*rep(treat,5)),ncol=5)
matrix(c(1.36,0.18,0.84,0.65,1.44,1.79,1.04,0.43,1.35,1.63,2.15,1.15,
1.21,5.46,1.58,3.44,4.40,2.75,4.78,2.44),ncol=5,byrow=TRUE)
times <- restovec(times, censor=cens)
reps <- rmna(times, ccov=tr)
# exponential intensity model with independence
kalsurv(times, pinitial=0.5, preg=1, dep="independence",
intensity="exponential")
# Weibull intensity model with independence
kalsurv(times, pinitial=0.5, preg=1, pshape=1, dep="independence",
intensity="Weibull")
# same model with serial update
kalsurv(times, pinitial=0.5, pdep=0.1, preg=1, pshape=1, dep="serial",
intensity="Weibull")
# try power variance family instead of gamma distribution for mixture
kalsurv(times, pinitial=0.5, pdep=0.1, preg=1, pshape=1, dep="serial",
intensity="Weibull", pfamily=0.1)
# treatment effect with log link
kalsurv(times, pinitial=0.5, preg=c(1,0), pshape=1, intensity="Weibull",
ccov=treat)
# or equivalently
kalsurv(times, mu=~exp(a+b*treat), pinitial=0.1, preg=c(1,0), pshape=1,
```

```

intensity="Weibull", envir=reps)
# with identity link instead
kalsurv(times, mu=~treat, pinitial=0.5, preg=c(1,0), pshape=1,
intensity="Weibull")
# or equivalently
kalsurv(times, mu=~a+b*treat, pinitial=0.5, preg=c(1,0), pshape=1,
intensity="Weibull", envir=reps)
# add the birth model
kalsurv(times, pinitial=0.5, preg=c(1,0), pshape=1,
intensity="Weibull", ccov=treat, pbirth=0)
# try frailty dependence
kalsurv(times, pinitial=0.5, preg=c(1,0), pshape=1, dep="frailty",
intensity="Weibull", ccov=treat)
# add autoregression
kalsurv(times, pinitial=0.5, preg=c(1,0), pshape=1, dep="frailty",
pdep=0.1, intensity="Weibull", ccov=treat)
# switch to gamma intensity model
kalsurv(times, pinitial=0.5, preg=c(1,0), pshape=1, intensity="gamma",
ccov=treat)

```

km

Kaplan-Meier Survivor Curves

Description

km calculates the Kaplan-Meier estimates for survival.

To plot the survivor curve, use `plot()`; for the empirical intensity curve, use `plot.intensity()`; for diagnostic curves to choose a distribution to which the data might belong, use `plot.dist()`.

Usage

```

km(times, censor, group=1, freq=1, cdf=FALSE)
## S3 method for class 'km'
plot(x, add=FALSE, xlim=NULL, ylim=c(0,1),
main=NULL, xlab="Time", ylab=NULL, lty=NULL, ...)
## S3 method for class 'km'
plot.intensity(x, add=FALSE, xlab="Time", ylab="Hazard", type="l", lty=NULL, ...)
## S3 method for class 'km'
plot.dist(x, ...)

```

Arguments

times	Vector of times to events or a list of vectors of such times for different individuals.
censor	Vector of censoring indicators corresponding to the vector of times or to the last time in each vector of a list.
group	Vector indicating to which group each individual belongs.
freq	Vector of frequencies for grouped data.

cdf	If TRUE, calculate the cdf instead of the survivor curve.
x	An object produced by km.
add	Plotting control options.
main	Plotting control options.
type	Plotting control options.
ylab	Plotting control options.
xlab	Plotting control options.
xlim	Plotting control options.
ylim	Plotting control options.
lty	Plotting control options.
...	Plotting control options.

Value

A matrix with class, km, containing the Kaplan-Meier estimates is returned.

Author(s)

J.K. Lindsey

See Also

[plot.intensity](#), [plot.surv](#)

Examples

```
surv <- rgamma(40,2,scale=5)
cens <- rbinom(40,1,0.9)
treat <- gl(2,20)
plot(km(surv, cens, group=treat), main="",xlab="Months",
ylab="Probability of deterioration")
plot.dist(km(surv, cens, group=treat))
plot.intensity(km(surv, cens, group=treat),ylab="Risk of deterioration")
```

pbirth

Fit Overdispersed Count Data as a Birth Process

Description

pbirth fits binomial, binomial exponential, binomial logistic, binomial total, Poisson, Poisson exponential, negative binomial, gen(eralized) negative binomial, and generalized negative binomial processes as a birth process.

Usage

```
pbirth(frequencies, p, intensity="negative binomial",
       type="spectral decomposition", print.level=0, ndigit=10,
       gradtol=0.00001, steptol=0.00001, fscale=1, iterlim=100,
       tysize=abs(p), stepmax=10*sqrt(p*%p))
```

Arguments

<code>frequencies</code>	Vector of frequencies or a matrix with each row a different series of frequencies.
<code>p</code>	Vector of initial estimates.
<code>intensity</code>	The intensity function of the process: binomial, binomial exponential, binomial logistic, binomial total, Poisson, Poisson exponential, negative binomial, or generalized negative binomial.
<code>type</code>	Algorithm used for matrix exponentiation: spectral decomposition or series approximation.
<code>print.level</code>	nlm control options.
<code>ndigit</code>	nlm control options.
<code>gradtol</code>	nlm control options.
<code>steptol</code>	nlm control options.
<code>iterlim</code>	nlm control options.
<code>fscale</code>	nlm control options.
<code>tysize</code>	nlm control options.
<code>stepmax</code>	nlm control options.

Author(s)

J.K. Lindsey

References

Faddy, M.J. and Fenlon, J.S. (1999) Stochastic modelling of the invasion process of nematodes in fly larvae. *Applied Statistics* 48: 31-37.

Examples

```
y <- rnbinom(100,2,0.6)
fr <- tabulate(y)
pbirth(fr, p=log(-log(0.7)), intensity="Poisson", type="series")
pbirth(fr, p=c(log(-log(0.7)),log(5)),
       intensity="negative binomial", type="series")
pbirth(fr, p=c(log(-log(0.7)),log(5),-1),
       intensity="gen negative binomial", type="series")
```

plot.intensity *Plot Intensity Functions*

Description

Plot the empirical intensity curve for given times between events.

Usage

```
## S3 method for class 'intensity'  
plot(x, ...)  
## Default S3 method:  
plot.intensity(x, censor=1, group=1, colour=TRUE, mix=1,  
ylim=c(0,1), ylab="p", xlab="Time",  
main="Empirical Hazard Function(s)", ...)
```

Arguments

censor	Vector of censoring indicators corresponding to the vector of times or to the last time in each vector of a list.
group	Vector indicating to which group each individual belongs.
colour	Use a different colour for each curve.
x	An object produced by <code>km</code> for <code>plot.intensity</code> ; for <code>plot.intensity.default</code> it is <code>times</code> (Vector of times to events or a list of vectors of such times for different individuals.) These changes were made for S3 methods compatibility.
mix	...
main	Plotting control options.
ylab	Plotting control options.
xlab	Plotting control options.
ylim	Plotting control options.
...	Plotting control options.

Author(s)

J.K. Lindsey

See Also

[km](#), [plot.surv](#)

Examples

```
surv <- rgamma(40,2,scale=5)
cens <- rbinom(40,1,0.9)
treat <- gl(2,20)
plot(km(surv, cens, group=treat), main="", xlab="Months",
     ylab="Probability of deterioration")
plot.dist(km(surv, cens, group=treat))
plot.intensity(km(surv, cens, group=treat), ylab="Risk of deterioration")
```

pp

Create a Point Process Vector from Times between Events

Description

pp creates a vector of length $\text{sum}(y)$ of zeroes with a one at the end of each uncensored time interval for use with [ehr](#).

Usage

```
pp(y, censor=1)
```

Arguments

y	Vector of times.
censor	Vector of censoring indicators.

Value

pp creates a vector of length $\text{sum}(y)$ of zeroes with a one at the end of each uncensored time interval for use with [ehr](#).

Author(s)

J.K. Lindsey

See Also

[bp](#), [ehr](#), [ident](#), [tccov](#), [tpast](#), [ttime](#), [tvcov](#).

Examples

```
y <- c(5,3,2,4)
py <- pp(y)
py
```

`tccov`*Create a Vector of Time-constant Covariates for a Point Process*

Description

`tccov` creates a vector of length `sum(y)` of time-constant covariates for use with [ehr](#). `id` must be numbered consecutively. `x` must have one value for each distinct `id`,

Usage

```
tccov(y, x, id)
```

Arguments

<code>y</code>	Vector of times.
<code>x</code>	Vector covariate.
<code>id</code>	Vector of corresponding individual identifiers for who had which sequence of times.

Value

`tccov` creates a vector of length `sum(y)` of time-constant covariates for use with [ehr](#). `id` must be numbered consecutively. `x` must have one value for each distinct `id`,

Author(s)

J.K. Lindsey

See Also

[bp](#), [ehr](#), [ident](#), [pp](#), [tpast](#), [ttime](#), [tvcov](#).

Examples

```
y <- c(5,3,2,4)
id <- c(1,1,2,2)
x <- c(5.2,3.1)
xcov <- tccov(y, x, id)
xcov
```

tpast	<i>Create a Vector of Times Past since Previous Events for a Point Process</i>
-------	--

Description

tpast creates a vector of length $\text{sum}(y)$ of times since the immediately preceding event occurred for use with [ehr](#).

Usage

```
tpast(y)
```

Arguments

y Vector of times.

Value

tpast creates a vector of length $\text{sum}(y)$ of times since the immediately preceding event occurred for use with [ehr](#).

Author(s)

J.K. Lindsey

See Also

[bp](#), [ehr](#), [ident](#), [pp](#), [tccov](#), [ttime](#), [tvcov](#).

Examples

```
y <- c(5,3,2,4)
ptime <- tpast(y)
ptime
```

ttime	<i>Create a Vector of Total Time Elapsed for each Individual for a Point Process</i>
-------	--

Description

ttime creates a vector of length $\text{sum}(y)$ of times since each individual began for use with [ehr](#).

Usage

```
ttime(y, id)
```

Arguments

<code>y</code>	Vector of times.
<code>id</code>	Vector of corresponding individual identifiers for who had which sequence of times.

Value

`ttime` creates a vector of length $\text{sum}(y)$ of times since each individual began for use with [ehr](#).

Author(s)

J.K. Lindsey

See Also

[bp](#), [ehr](#), [ident](#), [pp](#), [tccov](#), [tpast](#), [tvcov](#).

Examples

```
y <- c(5,3,2,4)
id <- c(1,1,2,2)
itime <- ttime(y, id)
itime
```

tvcov

Create a Vector of Time-varying Covariates for a Point Process

Description

`tvcov` creates a vector of length $\text{sum}(y)$ of time-varying covariates for use with [ehr](#). It may also be used to create weight vectors.

Usage

```
tvcov(y, x, tx)
```

Arguments

<code>y</code>	Vector of times.
<code>x</code>	Vector covariate.
<code>tx</code>	Vector of times at which <code>x</code> changes.

Value

`tvcov` creates a vector of length $\text{sum}(y)$ of time-varying covariates for use with [ehr](#). It may also be used to create weight vectors.

Author(s)

J.K. Lindsey

See Also

[bp](#), [ehr](#), [ident](#), [pp](#), [tccov](#), [tpast](#), [ttime](#).

Examples

```
y <- c(5,3,2,4)
x <- c(1,2,2,1,2,2,1)
tx <- c(2,3,1,2,2,2,2)
zcov <- tvcov(y, x, tx)
zcov
```


Index

*Topic **distribution**

hboxcox, 11
hburr, 12
hcauchy, 12
hexp, 13
hgamma, 14
hgextval, 15
hggamma, 16
hglogis, 17
hgweibull, 18
hhjorth, 19
hinvgauss, 20
hlaplace, 21
hlnorm, 22
hlogis, 22
hnorm, 23
hpareto, 24
hskewlaplace, 25
hstudent, 26
hweibull, 27

*Topic **documentation**

event, 9

*Topic **hplot**

autointensity, 2
cprocess, 6
km, 32
plot.intensity, 35

*Topic **manip**

bp, 4
ident, 27
pp, 36
tccov, 37
tpast, 38
ttime, 38
tvcov, 39

*Topic **models**

coxre, 5
ehr, 7
kalsurv, 28

pbirth, 33

autointensity, 2, 9

bp, 4, 8, 9, 28, 36–40

coxre, 5, 9, 31

cprocess, 3, 6, 9

dburr, 18

dcauchy, 21, 25

deviance.intensity (ehr), 7

deviance.kalsurv (kalsurv), 28

deviance.pbirth (pbirth), 33

dexp, 21, 24, 25

df, 18

dgamma, 16

dlaplace, 25

dlnorm, 16, 20

dlogis, 17

dnorm, 11, 20

dweibull, 15, 16, 18

ehr, 4, 7, 10, 27, 28, 36–40

event, 9

finterp, 7, 8, 28, 31

fitted.kalsurv (kalsurv), 28

gettvc, 31

hboxcox, 10, 11, 13, 14, 22–24, 26, 27

hburr, 10, 12, 13, 14, 22–24, 26, 27

hcauchy, 10, 12, 14, 22–24, 26, 27

hexp, 10, 13, 13, 14, 22–24, 26, 27

hgamma, 10, 13, 14, 14, 22–24, 26, 27

hgextval, 10, 13, 14, 15, 22–24, 26, 27

hggamma, 10, 13, 14, 16, 22–24, 26, 27

hglogis, 10, 13, 14, 17, 22–24, 26, 27

hgweibull, 10, 13, 14, 18, 22–24, 26, 27

hhjorth, 10, 13, 14, 19, 22–24, 26, 27

hinvgauss, *10, 13, 14, 20, 22–24, 26, 27*
hlaplace, *10, 13, 14, 21, 22–24, 26, 27*
hlnorm, *10, 13, 14, 22, 23, 24, 27*
hlogis, *10, 13, 14, 22, 22, 24, 26, 27*
hnorm, *10, 13, 14, 22, 23, 23, 26, 27*
hpareto, *10, 24*
hskewlaplace, *10, 25*
hstudent, *10, 13, 14, 22–24, 26, 27*
hweibull, *10, 13, 14, 22–24, 26, 27*

ident, *4, 8, 10, 27, 36–40*
iprofile, *28, 31*

kalsurv, *6, 10, 28*
km, *10, 32, 35*

lvna, *29*

mprofile, *28, 31*

pbirth, *10, 33*
plot.dist (km), *32*
plot.intensity, *33, 35*
plot.intensity.km (km), *32*
plot.km (km), *32*
plot.residuals, *28*
plot.surv, *33, 35*
plot.surv (km), *32*
pp, *4, 8, 10, 28, 36, 37–40*
print.intensity (ehr), *7*
print.kalsurv (kalsurv), *28*
print.km (km), *32*
print.llrf (coxre), *5*
print.pbirth (pbirth), *33*

read.list, *7, 10, 31*
read.surv, *10*
residuals.kalsurv (kalsurv), *28*
restovec, *29, 31*
rmna, *7, 29, 31*

tccov, *4, 8, 10, 28, 36, 37, 38–40*
tcctomat, *30, 31*
tpast, *4, 8, 10, 28, 36, 37, 38, 39, 40*
ttime, *4, 8, 10, 28, 36–38, 38, 40*
tvcov, *4, 8, 10, 28, 36–39, 39*
tvctomat, *30, 31*

vdm (ehr), *7*