Package ‘eha’

September 15, 2019

Encoding UTF-8
Version 2.7.6
Date 2019-09-14
Title Event History Analysis
Description Sampling of risk sets in Cox regression, selections in the Lexis diagram, bootstrapping. Parametric proportional hazards fitting with left truncation and right censoring for common families of distributions, piecewise constant hazards, and discrete models. Parametric accelerated failure time models for left truncated and right censored data.

BugReports https://github.com/goranbrostrom/eha/issues
License GPL (>= 2)
LazyData yes
ByteCompile yes
Depends R (>= 3.0.0), survival (>= 2.42-5)
Imports stats, graphics
NeedsCompilation yes
Maintainer Göran Broström <goran.brostrom@umu.se>
RoxygenNote 6.1.0
Suggests knitr, rmarkdown
VignetteBuilder knitr
Author Göran Broström [aut, cre], Jianming Jin [ctb]
Repository CRAN
Date/Publication 2019-09-15 00:10:25 UTC
### R topics documented:

<table>
<thead>
<tr>
<th>R package/Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>eha-package</td>
<td>3</td>
</tr>
<tr>
<td>aftreg</td>
<td>4</td>
</tr>
<tr>
<td>aftreg.fit</td>
<td>6</td>
</tr>
<tr>
<td>age.window</td>
<td>7</td>
</tr>
<tr>
<td>cal.window</td>
<td>8</td>
</tr>
<tr>
<td>check.dist</td>
<td>9</td>
</tr>
<tr>
<td>check.surv</td>
<td>10</td>
</tr>
<tr>
<td>coxreg</td>
<td>11</td>
</tr>
<tr>
<td>coxreg.fit</td>
<td>14</td>
</tr>
<tr>
<td>cro</td>
<td>16</td>
</tr>
<tr>
<td>eha-defunct</td>
<td>17</td>
</tr>
<tr>
<td>EV</td>
<td>18</td>
</tr>
<tr>
<td>fert</td>
<td>19</td>
</tr>
<tr>
<td>frail.fit</td>
<td>20</td>
</tr>
<tr>
<td>geome.fit</td>
<td>20</td>
</tr>
<tr>
<td>Gompertz</td>
<td>21</td>
</tr>
<tr>
<td>infants</td>
<td>23</td>
</tr>
<tr>
<td>join.spells</td>
<td>24</td>
</tr>
<tr>
<td>Loglogistic</td>
<td>25</td>
</tr>
<tr>
<td>Lognormal</td>
<td>26</td>
</tr>
<tr>
<td>logrye</td>
<td>27</td>
</tr>
<tr>
<td>ltx</td>
<td>28</td>
</tr>
<tr>
<td>make.communal</td>
<td>29</td>
</tr>
<tr>
<td>makeham</td>
<td>30</td>
</tr>
<tr>
<td>male.mortality</td>
<td>31</td>
</tr>
<tr>
<td>mlreg</td>
<td>32</td>
</tr>
<tr>
<td>mort</td>
<td>35</td>
</tr>
<tr>
<td>oldmort</td>
<td>36</td>
</tr>
<tr>
<td>Pch</td>
<td>37</td>
</tr>
<tr>
<td>perstat</td>
<td>38</td>
</tr>
<tr>
<td>phfunc</td>
<td>39</td>
</tr>
<tr>
<td>phreg</td>
<td>40</td>
</tr>
<tr>
<td>phreg.fit</td>
<td>43</td>
</tr>
<tr>
<td>piecewise</td>
<td>44</td>
</tr>
<tr>
<td>plot.aftreg</td>
<td>45</td>
</tr>
<tr>
<td>plot.coxreg</td>
<td>46</td>
</tr>
<tr>
<td>plot.hazdata</td>
<td>47</td>
</tr>
<tr>
<td>plot.phreg</td>
<td>48</td>
</tr>
<tr>
<td>plot.weibreg</td>
<td>50</td>
</tr>
<tr>
<td>print.aftreg</td>
<td>51</td>
</tr>
<tr>
<td>print.coxreg</td>
<td>52</td>
</tr>
<tr>
<td>print.phreg</td>
<td>52</td>
</tr>
<tr>
<td>print.risksets</td>
<td>53</td>
</tr>
<tr>
<td>print.weibreg</td>
<td>54</td>
</tr>
<tr>
<td>risksets</td>
<td>55</td>
</tr>
<tr>
<td>scania</td>
<td>56</td>
</tr>
</tbody>
</table>
**eha-package**

**Summary**

Sampling of risk sets in Cox regression, selections in the Lexis diagram, bootstrapping. Parametric proportional hazards fitting with left truncation and right censoring for common families of distributions, piecewise constant hazards, and discrete models. Parametric accelerated failure time models for left truncated and right censored data.

**Details**

Eha enhances the recommended `survival` package in several ways, see the description. The main applications in mind are demography and epidemiology. For standard Cox regression analysis the function `coxph` in `survival` is still recommended. The function `coxreg` in `eha` in fact calls coxph for the standard kind of analyses.

**Author(s)**

*Maintainer:* Göran Broström <goran.brostrom@umu.se>

**References**


**See Also**

Useful links:

- Report bugs at [https://github.com/goranbrostrom/eha/issues](https://github.com/goranbrostrom/eha/issues)
Accelerated Failure Time Regression

Description

The accelerated failure time model with parametric baseline hazard(s). Allows for stratification with different scale and shape in each stratum, and left truncated and right censored data.

Usage

aftreg(formula = formula(data), data = parent.frame(),
        na.action = getOption("na.action"), dist = "weibull", init,
        shape = 0, id, param = c("lifeAcc", "lifeExp"), control = list(eps
        = 1e-08, maxiter = 20, trace = FALSE), singular.ok = TRUE,
        model = FALSE, x = FALSE, y = TRUE)

Arguments

formula a formula object, with the response on the left of a ~ operator, and the terms on the right. The response must be a survival object as returned by the Surv function.
data a data.frame in which to interpret the variables named in the formula.
na.action a missing-data filter function, applied to the model.frame, after any subset argument has been used. Default is options()$na.action.
dist Which distribution? Default is "weibull", with the alternatives "gompertz", "ev", "loglogistic" and "lognormal". A special case like the exponential can be obtained by choosing "weibull" in combination with shape = 1.
init vector of initial values of the iteration. Default initial value is zero for all variables.
shape If positive, the shape parameter is fixed at that value. If zero or negative, the shape parameter is estimated. Stratification is now regarded as a meaningful option even if shape is fixed.
id If there are more than one spell per individual, it is essential to keep spells together by the id argument. This allows for time-varying covariates.
param Which parametrization should be used? The lifeAcc uses the parametrization given in the vignette, while the lifeExp uses the same as in the survreg function.
control a list with components eps (convergence criterion), maxiter (maximum number of iterations), and trace (logical, debug output if TRUE). You can change any component without mention the other(s).
singular.ok Not used.
model Not used.
x Return the design matrix in the model object?
y Return the response in the model object?
**Details**

The parameterization is different from the one used by `survreg`, when `param = "lifeAcc"`. The result is then true acceleration of time. Then the model is

\[
S(t; a, b, \beta, z) = S_0((t/\exp(b - z\beta))^{\exp(a)})
\]

S(t; a, b, beta, z) = S_0((t/exp(b - z beta))^{exp(a)})

where \(S_0\) is some standardized survivor function. The baseline parameters \(a\) and \(b\) are log shape and log scale, respectively. This is for the default parametrization. With the `lifeExp` parametrization, some signs are changed:

\[ b - zbeta \]

is changed to

\[ b + zbeta \]

. For the Gompertz distribution, the base parametrization is canonical, a necessity for consistency with the shape/scale paradigm (this is new in 2.3).

**Value**

A list of class `c("aftreg","coxreg")` with components

- `coefficients` Fitted parameter estimates.
- `var` Covariance matrix of the estimates.
- `loglik` Vector of length two; first component is the value at the initial parameter values, the second component is the maximized value.
- `score` The score test statistic (at the initial value).
- `linear.predictors` The estimated linear predictors.
- `means` Means of the columns of the design matrix.
- `w.means` Weighted (against exposure time) means of covariates; weighted relative frequencies of levels of factors.
- `n` Number of spells in indata (possibly after removal of cases with NA's).
- `events` Number of events in data.
- `terms` Used by extractor functions.
- `assign` Used by extractor functions.
- `wald.test` The Wald test statistic (at the initial value).
- `y` The Surv vector.
- `isF` Logical vector indicating the covariates that are factors.
- `covars` The covariates.
- `ttr` Total Time at Risk.
- `levels` List of levels of factors.
- `formula` The calling formula.
**aftreg.fit**

Parametric proportional hazards regression

This function is called by `aftreg`, but it can also be directly called by a user.

### Description

This function is called by `aftreg`, but it can also be directly called by a user.

### Usage

```r
aftreg.fit(X, Y, dist, param, strata, offset, init, shape, id, control, pfixed)
```

### Arguments

- **X**: The design (covariate) matrix.
- **Y**: A survival object, the response.
- **dist**: Which baseline distribution?
- **param**: Which parametrization?
- **strata**: A stratum variable.
- **offset**: Offset.
- **init**: Initial regression parameter values.
- **shape**: If positive, a fixed value of the shape parameter in the distribution. Otherwise, the shape is estimated.

---

**call**  The call.

**method**  The method.

**convergence**  Did the optimization converge?

**fail**  Did the optimization fail? (Is NULL if not).

**pfixed**  TRUE if shape was fixed in the estimation.

**param**  The parametrization.

**Author(s)**

Göran Broström

**See Also**

`coxreg`, `phreg`, `survreg`

**Examples**

```r
data(mort)
aftreg(Surv(enter, exit, event) ~ ses, param = "lifeExp", data = mort)
```
age.window

id See corresponding argument to aftreg.
control Controls convergence and output.
ptfixed A logical indicating fixed shape parameter(s).

Details
See aftreg for more detail.

Value
coefficients Estimated regression coefficients plus estimated scale and shape coefficients, sorted by strata, if present.
df Degrees of freedom; No. of regression parameters.
var Variance-covariance matrix
loglik Vector of length 2. The first component is the maximized loglikelihood with only scale and shape in the model, the second the final maximum.
conver TRUE if convergence
fail TRUE if failure
iter Number of Newton-Raphson iterates.
n.strata The number of strata in the data.

Author(s)
Göran Broström

See Also
aftreg

---

age.window Age cut of survival data

Description
For a given age interval, each spell is cut to fit into the given age interval.

Usage
age.window(dat, window, surv = c("enter", "exit", "event"))

Arguments
dat Input data frame. Must contain survival data.
window Vector of length two; the age interval.
surv Vector of length three giving the names of the central variables in 'dat'.
cal.window

Calendar time cut of survival data

Description
For a given time interval, each spell is cut so that it fully lies in the given time interval

Usage

```r
cal.window(dat, window, surv = c("enter", "exit", "event", "birthdate"))
```

Arguments

- `dat` Input data frame. Must contain survival data and a birth date.
- `window` Vector of length two; the time interval
- `surv` Vector of length four giving the names of the central variables in `dat`

Details
The window must be in the order (begin, end)

Value
A data frame of the same form as the input data frame, but 'cut' as desired. Intervals exceeding `window[2]` will be given `event = 0`

Author(s)
Göran Broström

See Also

- `cal.window`
- `coxreg`
- `aftreg`

Examples

```r
dat <- data.frame(enter = 0, exit = 5.731, event = 1, x = 2)
window <- c(2, 5.3)
dat.trim <- age.window(dat, window)
```
check.dist

Author(s)
Göran Broström

See Also
age.window, coxreg, aftreg

Examples

```r
dat <- data.frame(enter = 0, exit = 5.731, event = 1,
birthdate = 1962.505, x = 2)
window <- c(1963, 1965)
dat.trim <- cal.window(dat, window)
```

Description
Comparison of the cumulative hazards functions for a semi-parametric and a parametric model.

Usage
check.dist(sp, pp, main = NULL, col = NULL, printLegend = TRUE)

Arguments
- `sp`: An object of type "coxreg", typically output from coxreg
- `pp`: An object of type "phreg", typically output from phreg
- `main`: Header for the plot. Default is distribution and "cumulative hazard function"
- `col`: Line colors. should be NULL (black lines) or of length 2
- `printLegend`: Should a legend be printed? Default is TRUE.

Details
For the moment only a graphical comparison. The arguments sp and pp may be swapped.

Value
No return value.

Author(s)
Göran Broström
check.surv

**Check the integrity of survival data.**

**Description**

Check that exit occurs after enter, that spells from an individual do not overlap, and that each individual experiences at most one event.

**Usage**

```r
check.surv(enter, exit, event, id = NULL, eps = 1e-08)
```

**Arguments**

- `enter`  
  Left truncation time.
- `exit`  
  Time of exit.
- `event`  
  Indicator of event. Zero means 'no event'.
- `id`  
  Identification of individuals.
- `eps`  
  The smallest allowed spell length or overlap.

**Details**

Interval lengths must be strictly positive.
**Value**

A vector of id’s for the insane individuals. Of zero length if no errors.

**Author(s)**

Göran Broström

**See Also**

join.spells, coxreg, aftreg

**Examples**

```r
xx <- data.frame(enter = c(0, 1), exit = c(1.5, 3), event = c(0, 1), id = c(1,1))
check.surv(xx$enter, xx$exit, xx$event, xx$id)
```

---

**coxreg**

*Cox regression*

**Description**

Performs Cox regression with some special attractions, especially sampling of risksets and the weird bootstrap.

**Usage**

```r
coxreg(formula = formula(data), data = parent.frame(), weights, subset, t.offset, na.action = getOption("na.action"), init = NULL, method = c("efron", "breslow", "mpl", "ml"), control = list(eps = 1e-08, maxiter = 25, trace = FALSE), singular.ok = TRUE, model = FALSE, center = TRUE, x = FALSE, y = TRUE, hazards = TRUE, boot = FALSE, efrac = 0, geometric = FALSE, rs = NULL, frailty = NULL, max.survs = NULL)
```

**Arguments**

- `formula` a formula object, with the response on the left of a ~ operator, and the terms on the right. The response must be a survival object as returned by the Surv function.
- `data` a data.frame in which to interpret the variables named in the formula.
- `weights` Case weights; time-fixed or time-varying.
- `subset` An optional vector specifying a subset of observations to be used in the fitting process.
- `t.offset` Case offsets; time-varying.
na.action a missing-data filter function, applied to the model.frame, after any subset argument has been used. Default is options()$na.action.

init vector of initial values of the iteration. Default initial value is zero for all variables.

method Method of treating ties, "efron" (default), "breslow", "mppl" (maximum partial partial likelihood), or "ml" (maximum likelihood).

control a list with components eps (convergence criterion), maxiter (maximum number of iterations), and silent (logical, controlling amount of output). You can change any component without mention the other(s).

singular.ok Not used

model Not used

center Logical. If center = TRUE (default), the baseline hazards are calculated at the means of the covariates and for the reference category for factors, otherwise at the value zero. See Details.

x Return the design matrix in the model object?

y return the response in the model object?

hazards Calculate baseline hazards? Default is TRUE.

boot Number of boot replicates. Defaults to FALSE, no boot samples.

efrac Upper limit of fraction failures in 'mppl'.

geometric If TRUE, forces an 'ml' model with constant riskset probability. Default is FALSE.

rs Risk set?

frailty Grouping variable for frailty analysis. Not in use yet.

max.survs Sampling of risk sets? If given, it should be (the upper limit of) the number of survivors in each risk set.

Details

The default method, efron, and the alternative, breslow, are both the same as in coxph in package survival. The methods mppl and ml are maximum likelihood, discrete-model, based.

If center = TRUE (default), graphs show the "baseline" distribution at the means of (continuous) covariates, and for the reference category in case of factors (avoiding representing "flying pigs"). If center = FALSE the baseline distribution is at the value zero of all covariates. It is usually a good idea to use center = FALSE in combination with "precentering" of covariates, that is, subtracting a reference value, ideally close to the center of the covariate distribution. In that way, the "reference" will be the same for all subsets of the data.

Value

A list of class c("coxreg", "coxph") with components

coefficients Fitted parameter estimates.

cov Covariance matrix of the estimates.
**loglik**  Vector of length two; first component is the value at the initial parameter values, the second component is the maximized value.

**score**  The score test statistic (at the initial value).

**linear.predictors**  The estimated linear predictors.

**residuals**  The martingale residuals.

**hazard**  The estimated baseline hazard, calculated at the means of the covariates (rather, columns of the design matrix). Is a list, with one component per stratum. Each component is a matrix with two columns, the first contains risktimes, the second the corresponding hazard atom.

**means**  Means of the columns of the design matrix corresponding to covariates, if center = TRUE. Columns corresponding to factor levels give a zero in the corresponding position in means. If center = FALSE, means are all zero.

**w.means**  Weighted (against exposure time) means of covariates; weighted relative frequencies of levels of factors.

**n**  Number of spells in indata (possibly after removal of cases with NA's).

**events**  Number of events in data.

**terms**  Used by extractor functions.

**assign**  Used by extractor functions.

**y**  The Surv vector.

**isF**  Logical vector indicating the covariates that are factors.

**covars**  The covariates.

**ttr**  Total Time at Risk.

**levels**  List of levels of factors.

**formula**  The calling formula.

**bootstrap**  The (matrix of) bootstrap replicates, if requested on input. It is up to the user to do whatever desirable with this sample.

**boot.sd**  The estimated standard errors of the bootstrap replicates.

**call**  The call.

**method**  The method.

**convergence**  Did the optimization converge?

**fail**  Did the optimization fail? (Is NULL if not).

---

**Warning**

The use of rs is dangerous, see note. It can however speed up computing time considerably for huge data sets.

**Note**

This function starts by creating risksets, if no riskset is supplied via rs, with the aid of risksets. Supplying output from risksets via rs fails if there are any NA's in the data! Note also that it depends on stratification, so rs contains information about stratification. Giving another strata variable in the formula is an error. The same is ok, for instance to supply stratum interactions.
Author(s)
Göran Broström

References

See Also
coxph, risksets

Examples

```r
dat <- data.frame(time = c(4, 3, 1, 1, 2, 2, 3),
    status = c(1, 1, 1, 0, 1, 1, 0),
    x = c(0, 2, 1, 1, 0, 0),
    sex = c(0, 0, 0, 0, 1, 1, 1))
coxreg(Surv(time, status) ~ x + strata(sex), data = dat) # stratified model
# Same as:
rs <- risksets(Surv(dat$time, dat$status), strata = dat$sex)
coxreg(Surv(time, status) ~ x, data = dat, rs = rs) # stratified model
```

---

coxreg.fit  
Cox regression

Description
Called by coxreg, but a user can call it directly.

Usage
coxreg.fit(X, Y, rs, weights, t.offset = NULL, strats, offset, init,
max.survs, method = "breslow", center = TRUE, boot = FALSE,
efrac = 0, calc.hazards = TRUE, calc.martres = TRUE, control,
verbose = TRUE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>The design matrix.</td>
</tr>
<tr>
<td>Y</td>
<td>The survival object.</td>
</tr>
<tr>
<td>rs</td>
<td>The risk set composition. If absent, calculated.</td>
</tr>
<tr>
<td>weights</td>
<td>Case weights; time-fixed or time-varying.</td>
</tr>
<tr>
<td>t.offset</td>
<td>Case offset; time-varying.</td>
</tr>
<tr>
<td>strats</td>
<td>The stratum variable. Can be absent.</td>
</tr>
</tbody>
</table>
**offset**  Offset. Can be absent.

**init**  Start values. If absent, equal to zero.

**max.survs**  Sampling of risk sets? If so, gives the maximum number of survivors in each risk set.

**method**  Either "efron" (default) or "breslow".

**center**  See *coxreg*.

**boot**  Number of bootstrap replicates. Defaults to FALSE, no bootstrapping.

**efrac**  Upper limit of fraction failures in 'mppl'.

**calc.hazards**  Should estimates of baseline hazards be calculated?

**calc.martres**  Should martingale residuals be calculated?

**control**  See *coxreg*

**verbose**  Should Warnings about convergence be printed?

**Details**

rs is dangerous to use when NA’s are present.

**Value**

A list with components

- **coefficients**  Estimated regression parameters.
- **var**  Covariance matrix of estimated coefficients.
- **loglik**  First component is value at init, second at maximum.
- **score**  Score test statistic, at initial value.
- **linear.predictors**  Linear predictors.
- **residuals**  Martingale residuals.
- **hazard**  Estimated baseline hazard. At value zero of 'design' variables.
- **means**  Means of the columns of the design matrix.
- **bootstrap**  The bootstrap replicates, if requested on input.
- **conver**  TRUE if convergence.
- **f.conver**  TRUE if variables converged.
- **fail**  TRUE if failure.
- **iter**  Number of performed iterations.

**Note**

It is the user's responsibility to check that indata is sane.

**Author(s)**

Göran Broström
See Also

croreg.risksets

Examples

```r
X <- as.matrix(data.frame(
    x = c(0, 2,1,4,1,0,3),
    sex = c(1, 0,0,0,1,1,1)))
time <- c(1,2,3,4,5,6,7)
status <- c(1,1,1,0,1,1,0)
stratum <- rep(1, length(time))
coxreg.fit(X, Surv(time, status), strats = stratum, max.survs = 6,
            control = list(eps=1.e-4, maxiter = 10, trace = FALSE))
```

**cro**

*Creates a minimal representation of a data frame.*

Description

Given a data frame with a defined response variable, this function creates a unique representation of the covariates in the data frame, vector (matrix) of responses, and a pointer vector, connecting the responses with the corresponding covariates.

Usage

cro(dat, response = 1)

Arguments

dat
  A data frame
response
  The column(s) where the response resides.

Details

The rows in the data frame are converted to text strings with `paste` and compared with `match`.

Value

A list with components

`y`
  The response.
`covar`
  A data frame with unique rows of covariates.
`keys`
  Pointers from `y` to `covar`, connecting each response with its covariate vector.
Note

This function is based on suggestions by Anne York and Brian Ripley.

Author(s)

Göran Broström

See Also

match, paste

Examples

```r
dat <- data.frame(y = c(1.1, 2.3, 0.7), x1 = c(1, 0, 1), x2 = c(0, 1, 0))
cro(dat)
```

Defunct functions

These functions were duplicates of functions in the package glmmML.

Usage

ghq(...)

ghmboot(...)

glmmbootFit(...)

glmmML(...)

glmmML.fit(...)

Arguments

... input parameters

Details

Instead of using these functions, use the corresponding functions in glmmML with the same name.
**The EV Distribution**

**Description**
Density, distribution function, quantile function, hazard function, cumulative hazard function, and random generation for the EV distribution with parameters shape and scale.

**Usage**
- `dEV(x, shape = 1, scale = 1, log = FALSE)`
- `pEV(q, shape = 1, scale = 1, lower.tail = TRUE, log.p = FALSE)`
- `qEV(p, shape = 1, scale = 1, lower.tail = TRUE, log.p = FALSE)`
- `hEV(x, shape = 1, scale = 1, log = FALSE)`
- `HEV(x, shape = 1, scale = 1, log.p = FALSE)`
- `rEV(n, shape = 1, scale = 1)`

**Arguments**
- `shape, scale` shape and scale parameters, both defaulting to 1.
- `lower.tail` logical; if TRUE (default), probabilities are \( P(X \leq x) \), otherwise, \( P(X > x) \).
- `x, q` vector of quantiles.
- `p` vector of probabilities.
- `n` number of observations. If \( \text{length}(n) > 1 \), the length is taken to be the number required.
- `log, log.p` logical; if TRUE, probabilities \( p \) are given as \( \log(p) \).

**Details**
The EV distribution with scale parameter \( a \) and shape parameter \( \sigma \) has hazard function given by
\[
h(x) = \frac{b}{\sigma} \left( x/\sigma \right)^{\sigma-1} \exp \left( (x/\sigma)^b \right)
\]
for \( x \geq 0 \).

**Value**
- `dEV` gives the density, `pEV` gives the distribution function, `qEV` gives the quantile function, `hEV` gives the hazard function, `HEV` gives the cumulative hazard function, and `rEV` generates random deviates.

Invalid arguments will result in return value NaN, with a warning.
Description
Birth intervals for married women with at least one birth, 19th northern Sweden

Usage
data(fert)

Format
A data frame with 12169 observations the lengths (in years) of birth intervals for 1859 married women with at least one birth. The first interval \( \text{parity} = 0 \) is the interval from marriage to first birth.

- **id**  Personal identification number for mother.
- **parity**  Time order of birth interval for the present mother. The interval with \( \text{parity} = 0 \) is the first, from marriage to first birth.
- **age**  The age of mother at start of interval.
- **year**  The calendar year at start of interval.
- **next.ivl**  The length of the coming time interval.
- **event**  An indicator for whether the \( \text{next.ivl} \) ends in a new birth \( \text{event} = 1 \) or is right censored \( \text{event} = 0 \). Censoring occurs when the woman ends her fertility period within her first marriage (marriage dissolution or reaching the age of 48).
- **prev.ivl**  The length of the previous time interval. May be used as explanatory variable in a Cox regression of birth intervals.
- **ses**  Socio-economic status, a factor with levels lower, upper, farmer, and unknown.
- **parish**  The Skelleftea region consists of three parishes, Jorn, Norsjo, and Skelleftea.

Details
The data set contain clusters of dependent observations defined by mother’s id.

Source
Data is coming from The Demographic Data Base, Umea University, Umea, Sweden.

References
http://www.ddb.umu.se
Examples

```r
data(fert)
fit <- coxreg(Surv(next.ivl, event) ~ ses + prev.ivl, data = fert, subset =
(parity == 1))
drop1(fit, test = "Chisq")
```

---

**frail.fit**

*Frailty experiment*

**Description**

Utilizing GLMM models: Experimental, not exported (yet).

**Usage**

```r
frail.fit(X, Y, rs, strats, offset, init, max.survs, frailty, control)
```

**Arguments**

- `X`: design matrix
- `Y`: survival object
- `rs`: output from `risksets`
- `strats`: strata
- `offset`: offset
- `init`: start values
- `max.survs`: for sampling of riskset survivors
- `frailty`: grouping variable
- `control`: control of optimization

---

**geome.fit**

*Constant intensity discrete time proportional hazards*

**Description**

This function is called from `coxreg`. A user may call it directly.

**Usage**

```r
geome.fit(X, Y, rs, strats, offset, init, max.survs, method = "ml",
control)
```
Arguments

- **X**: The design matrix
- **Y**: Survival object
- **rs**: risk set produced by `risksets`
- **strats**: Stratum indicator
- **offset**: Offset
- **init**: Initial values
- **max.survs**: Maximal survivors
- **method**: "ml", always, i.e., this argument is ignored.
- **control**: See `coxreg`.

Value

See the code.

Note

Nothing special

`coxreg` is a defunct function

Author(s)

Göran Broström

References

See `coxreg`.

See Also

`coxreg`

---

### Description

The Gompertz Distribution

Density, distribution function, quantile function, hazard function, cumulative hazard function, and random generation for the Gompertz distribution with parameters shape and scale.
Gompertz

Usage

dgompertz(x, shape = 1, scale = 1, log = FALSE,  
  param = c("default", "canonical"))  
pgompertz(q, shape = 1, scale = 1, lower.tail = TRUE, log.p = FALSE,  
  param = c("default", "canonical"))  
qgompertz(p, shape = 1, scale = 1, lower.tail = TRUE, log.p = FALSE,  
  param = c("default", "canonical"))  
hgompertz(x, shape = 1, scale = 1, log = FALSE,  
  param = c("default", "canonical"))  
Hgompertz(x, shape = 1, scale = 1, log.p = FALSE,  
  param = c("default", "canonical"))  
rgompertz(n, shape = 1, scale = 1, param = c("default", "canonical"))

Arguments

shape, scale shape and scale parameters, both defaulting to 1.  
lower.tail logical; if TRUE (default), probabilities are $P(X \leq x)$, otherwise, $P(X > x)$.  
param default or canonical.  
x, q vector of quantiles.  
p vector of probabilities.  
n number of observations. If length(n) > 1, the length is taken to be the number required.  
log, log.p logical; if TRUE, probabilities p are given as log(p).

Details

The Gompertz distribution with scale parameter $a$ and shape parameter $\sigma$ has hazard function given by  

$$h(x) = a \exp(x/\sigma)$$  

for $x \geq 0$. If param = "canonical", then then $a \rightarrow a/b$, so that $b$ is a true scale parameter (for any fixed $a$), and $b$ is an 'AFT parameter'.

Value

dgompertz gives the density, pgompertz gives the distribution function, qgompertz gives the quantile function, hgompertz gives the hazard function, Hgompertz gives the cumulative hazard function, and rgompertz generates random deviates.

Invalid arguments will result in return value NaN, with a warning.
Description

Matched data on infant mortality, from seven parishes in Sweden, 1821–1894.

Usage
data( infants )

Format

A data frame with 80 rows and five variables.

stratum Triplet No. Each triplet consist of one infant whose mother died (a case), and two controls, i.e., infants whose mother did not die. Matched on covariates below.
enter Age (in days) of case when its mother died.
exit Age (in days) at death or right censoring (at age 365 days).
event Follow-up ends with death (1) or right censoring (0).
mother dead for cases, alive for controls.
age Mother's age at infant's birth.
sex The infant's sex.
parish Birth parish, either Nedertorneå or not Nedertorneå.
civst Civil status of mother, married or unmarried.
sest Socio-economic status of mother, either farmer or not farmer.
year Year of birth of the infant.

Details

From 5641 first-born in seven Swedish parishes 1820-1895, from Fleninge in the very south to Nedertorneå in the very north, those whose mother died during their first year of life were selected, in all 35 infants. To each of them, two controls were selected by matching on the given covariates.

Source

Data originate from The Demographic Data Base, Umeå University, Umeå, Sweden, http://www.ddb.umu.se.

References

Examples

data(infants)
fit <- coxreg(Surv(enter, exit, event) ~ strata(stratum) + mother, data = infants)
fit
fit.w <- phreg(Surv(enter, exit, event) ~ mother + parish + ses, data = infants)
fit.w ## Weibull proportional hazards model.

join.spells  

join.spells

Straighten up a survival data frame

Description

Unnecessary cut spells are glued together, overlapping spells are "polished", etc.

Usage

join.spells(dat, strict = FALSE, eps = 1e-08)

Arguments

dat A data frame with names enter, exit, event, id.
strict If TRUE, nothing is changed if errors in spells (non-positive length, overlapping intervals, etc.) are detected. Otherwise (the default), bad spells are removed, with "earlier life" having higher priority.
eps Tolerance for equality of two event times. Should be kept small.

Details

In case of overlapping intervals (i.e., a data error), the appropriate id's are returned if strict is TRUE.

Value

A data frame with the same variables as the input, but individual spells are joined, if possible (identical covariate values, and adjacent time intervals).

Author(s)

Göran Broström

References

Loglogistic

The Loglogistic Distribution

Description
Density, distribution function, quantile function, hazard function, cumulative hazard function, and random generation for the Loglogistic distribution with parameters shape and scale.

Usage

dllogis(x, shape = 1, scale = 1, log = FALSE)
pllogis(q, shape = 1, scale = 1, lower.tail = TRUE, log.p = FALSE)
qllogis(p, shape = 1, scale = 1, lower.tail = TRUE, log.p = FALSE)
hllogis(x, shape = 1, scale = 1, prop = 1, log = FALSE)
Hllogis(x, shape = 1, scale = 1, prop = 1, log.p = FALSE)
rllogis(n, shape = 1, scale = 1)

Arguments

shape, scale shape and scale parameters, both defaulting to 1.
lower.tail logical; if TRUE (default), probabilities are \( P(X \leq x) \), otherwise, \( P(X > x) \).
x, q vector of quantiles.
p vector of probabilities.
n number of observations. If length(n) > 1, the length is taken to be the number required.
log, log.p logical; if TRUE, probabilities p are given as log(p).
prop proportionality constant in the extended Loglogistic distribution.

Details
The Loglogistic distribution with scale parameter \( a \) and shape parameter \( \sigma \) has hazard function given by

\[
h(x) = \left(\frac{b}{\sigma}\right)\left(\frac{x}{\sigma}\right)^{(b-1)} \exp\left(\left(\frac{x}{\sigma}\right)^b\right)
\]

for \( x \geq 0 \).

Value
dllogis gives the density, pllogis gives the distribution function, qllogis gives the quantile function, hllogis gives the hazard function, Hllogis gives the cumulative hazard function, and rllogis generates random deviates.
Invalid arguments will result in return value NaN, with a warning.
**Lognormal**

*The Lognormal Distribution*

**Description**
Density, distribution function, quantile function, hazard function, cumulative hazard function, and random generation for the Lognormal distribution with parameters shape and scale.

**Usage**

```r
hlnorm(x, meanlog = 0, sdlog = 1, shape = 1 / sdlog, scale = exp(meanlog),
       prop = 1, log = FALSE)
```

```r
Hlnorm(x, meanlog = 0, sdlog = 1, shape = 1 / sdlog, scale = exp(meanlog),
       prop = 1, log.p = FALSE)
```

**Arguments**

- **x**: vector of quantiles.
- **meanlog**: mean in the Normal distribution.
- **sdlog, shape**: sdlog is standard deviation in the Normal distribution, shape = 1/sdlog.
- **scale**: is exp(meanlog).
- **prop**: proportionality constant in the extended Lognormal distribution.
- **log, log.p**: logical; if TRUE, probabilities p are given as log(p).

**Details**

The Lognormal distribution with scale parameter $a$ and shape parameter $\sigma$ has hazard function given by

$$h(x) = \left(\frac{b}{\sigma}\right)\left(\frac{x}{\sigma}\right)^{b-1} \exp\left(\frac{x}{\sigma}\right)^b$$

for $x \geq 0$.

**Value**

dlnorm gives the density, plnorm gives the distribution function, qlnorm gives the quantile function, hlnorm gives the hazard function, Hlnorm gives the cumulative hazard function, and rlnorm generates random deviates.

Invalid arguments will result in return value NaN, with a warning.
Description

The data consists of yearly rye prices from 1801 to 1894. Logged and detrended, so the time series is supposed to measure short term fluctuations in rye prices.

Usage

data(scania)

Format

A data frame with 94 observations in two columns on the following 2 variables.

year  The year the price is recorded.
foodprices  Detrended log rye prices.

Details

The Scanian area in southern Sweden was during the 19th century a mainly rural area.

Source

The Scanian Economic Demographic Database.

References


Examples

data(logrye)
summary(logrye)
Description

This (generic) function prints the LaTeX code of the results of a fit from \texttt{coxreg}, \texttt{phreg}, or \texttt{aftreg}, similar to what \texttt{xtable} does for fits from other functions.

Usage

\begin{verbatim}
ltx(x, caption = NULL, label = NULL, dr = NULL, digits = max(options()$digits - 4, 3), ...)
\end{verbatim}

Arguments

- \texttt{x} \hspace{1cm} The output from a call to \texttt{coxreg}, \texttt{phreg}, or \texttt{aftreg}
- \texttt{caption} \hspace{1cm} A suitable caption for the table.
- \texttt{label} \hspace{1cm} A label used in the LaTeX code.
- \texttt{dr} \hspace{1cm} Output from a \texttt{drop1} call.
- \texttt{digits} \hspace{1cm} Number of digits to be printed.
- \texttt{...} \hspace{1cm} Not used.

Details

The result is a printout which is (much) nicer than the standard printed output from \texttt{glm} and friends.

Value

LaTeX code version of the results from a run with \texttt{coxreg}, \texttt{phreg}, or \texttt{aftreg}.

Note

There is no method in \texttt{xtable} for \texttt{coxreg}.

Author(s)

Göran Broström.

See Also

\texttt{xtable}, \texttt{coxreg}
Examples

```r
data(oldmort)
fit <- coxreg(Surv(enter, exit, event) ~ civ + sex, data = oldmort)
dr <- drop1(fit, test = "Chisq")
ltx(fit, dr = dr, caption = "A test example.", label = "tab:test1")
```

Description

Given an ordinary data frame suitable for survival analysis, and a data frame with "communal" time series, this function includes the communal covariates as fixed, by the "cutting spells" method.

Usage

```r
make.communal(dat, com.dat, communal = TRUE, start, period = 1,
lag = 0, surv = c("enter", "exit", "event", "birthdate"),
tol = 1e-04, fortran = TRUE)
```

Arguments

dat A data frame containing interval specified survival data and covariates, of which one must give a "birth date", the connection between duration and calendat time
com.dat Data frame with communal covariates. They must have the same start year and periodicity, given by com.ins
communal Boolean: if TRUE, then it is a true communal (default), otherwise a fixed. The first component is the first year (start date in decimal form), and the second component is the period length. The third is lag and the fourth is scale.
start Start date in decimal form.
period Period length. Defaults to one.
lag The lag of the effect. Defaults to zero.
surv Character vector of length 4 giving the names of interval start, interval end, event indicator, birth date, in that order. These names must correspond to names in dat
tol Largest length of an interval considered to be of zero length. The cutting sometimes produces zero length intervals, which we want to discard.
fortran If TRUE, then a Fortran implementation of the function is used. This is the default. This possibility is only for debugging purposes. You should of course get identical results with the two methods.

Details

The main purpose of this function is to prepare a data file for use with `coxreg`, `aftreg`, and `coxph`. 
Value

The return value is a data frame with the same variables as in the combination of `dat` and `com.dat`. Therefore it is an error to have common name(s) in the two data frames.

Note

Not very vigorously tested.

Author(s)

Göran Broström

See Also

`coxreg`, `aftreg`, `coxph`, `cal.window`

Examples

```r
dat <- data.frame(enter = 0, exit = 5.731, event = 1,
  birthdate = 1962.505, x = 2)
## Birth date: July 2, 1962 (approximately).
com.dat <- data.frame(price = c(12, 3, -5, 6, -8, -9, 1, 7))
dat.com <- make.communal(dat, com.dat, start = 1962.000)
```

Description

Density, distribution function, quantile function, hazard function, cumulative hazard function, and random generation for the Gompertz-Makeham distribution with parameters `shape` and `scale`.

Usage

```r
dmakeham(x, shape = c(1, 1), scale = 1, log = FALSE)
pmakeham(q, shape = c(1, 1), scale = 1, lower.tail = TRUE, log.p = FALSE)
qmakeham(p, shape = c(1, 1), scale = 1, lower.tail = TRUE, log.p = FALSE)
hmakeham(x, shape = c(1, 1), scale = 1, log = FALSE)
Hmakeham(x, shape = c(1, 1), scale = 1, log.p = FALSE)
rmakeham(n, shape = c(1, 1), scale = 1)
```
Arguments

- **shape**: A vector, default value c(1, 1).
- **scale**: defaulting to 1.
- **lower.tail**: logical; if TRUE (default), probabilities are \( P(X \leq x) \), otherwise, \( P(X > x) \).
- **x, q**: vector of quantiles.
- **p**: vector of probabilities.
- **n**: number of observations. If `length(n) > 1`, the length is taken to be the number required.
- **log, log.p**: logical; if TRUE, probabilities p are given as log(p).

Details

The Gompertz-Makeham distribution with scale parameter \( a \) and shape parameter \( \sigma \) has hazard function given by

\[
h(x) = a[1] + a[2] \exp(x/\sigma)
\]

for \( x \geq 0 \).

Value

- `dmakeham` gives the density,
- `pmakeham` gives the distribution function,
- `qmakeham` gives the quantile function,
- `hmakeham` gives the hazard function,
- `Hmakeham` gives the cumulative hazard function, and
- `rmakeham` generates random deviates.

Invalid arguments will result in return value NaN, with a warning.

---

male.mortality  

Male mortality in ages 40-60, nineteenth century

Description

Males born in the years 1800-1820 and surviving at least 40 years in the parish Skellefteå in northern Sweden are followed from their fortieth birthday until death or the sixtieth birthday, whichever comes first.

Usage

data(male.mortality)

Format

A data frame with 2058 observations on the following 6 variables.

- **id**: Personal identification number.
- **enter**: Start of duration. Measured in years since the fortieth birthday.
- **exit**: End of duration. Measured in years since the fortieth birthday.
- **event**: a logical vector indicating death at end of interval.
- **birthdate**: The birthdate in decimal form.
- **ses**: Socio-economic status, a factor with levels lower, upper
Details

The interesting explanatory covariate is \textit{ses} (socioeconomic status), which is a time-varying covariate. This explains why several individuals are represented by more than one record each. Left truncation and right censoring are introduced this way.

Note

This data set is also known, and accessible, as \textit{mort}.

Source

Data is coming from The Demographic Data Base, Umeå University, Umeå, Sweden.

References

http://www.ddb.umu.se

Examples

\begin{verbatim}
  data(male.mortality)
  coxreg(Surv(enter, exit, event) ~ ses, data = male.mortality)
\end{verbatim}

\textbf{mlreg} \hspace{1cm} \textit{ML proportional hazards regression}

Description

Maximum Likelihood estimation of proportional hazards models. Is deprecated, use \textit{coxreg} instead.

Usage

\begin{verbatim}
  mlreg(formula = formula(data), data = parent.frame(),
  na.action = getOption("na.action"), init = NULL, method = c("ML",
  "MPPL"), control = list(eps = 1e-08, maxiter = 10, n.points = 12, trace
  = FALSE), singular.ok = TRUE, model = FALSE, center = TRUE,
  x = FALSE, y = TRUE, boot = FALSE, geometric = FALSE,
  rs = NULL, frailty = NULL, max.survs = NULL)
\end{verbatim}

Arguments

- \textbf{formula} a formula object, with the response on the left of a \sim operator, and the terms on the right. The response must be a survival object as returned by the \textit{Surv} function.
- \textbf{data} a data.frame in which to interpret the variables named in the formula.
- \textbf{na.action} a missing-data filter function, applied to the model.frame, after any subset argument has been used. Default is \texttt{options()}$\texttt{na.action}.
mlreg

init vector of initial values of the iteration. Default initial value is zero for all variables.

method Method of treating ties, "ML", the default, means pure maximum likelihood, i.e., data are treated as discrete. The choice "MPPL" implies that risk sets with no tied events are treated as in ordinary Cox regression. This is a cameleont that adapts to data, part discrete and part continuous.

control a list with components eps (convergence criterion), maxiter (maximum number of iterations), and silent (logical, controlling amount of output). You can change any component without mention the other(s).

singular.ok Not used.

model Not used.

center Should covariates be centered? Default is TRUE

x Return the design matrix in the model object?

y return the response in the model object?

boot No. of bootstrap replicates. Defaults to FALSE, i.e., no bootstrapping.

geometric If TRUE, the intensity is assumed constant within strata.

rs Risk set? If present, speeds up calculations considerably.

frailty A grouping variable for frailty analysis. Full name is needed.

max.survs Sampling of risk sets?

Details

Method ML performs a true discrete analysis, i.e., one parameter per observed event time. Method MPPL is a compromise between the discrete and continuous time approaches; one parameter per observed event time with multiple events. With no ties in data, an ordinary Cox regression (as with coxreg) is performed.

Value

A list of class c(“mlreg”, “coxreg”, “coxph”) with components

coefficients Fitted parameter estimates.

var Covariance matrix of the estimates.

loglik Vector of length two; first component is the value at the initial parameter values, the second component is the maximized value.

score The score test statistic (at the initial value).

linear.predictors The estimated linear predictors.

residuals The martingale residuals.

hazard The estimated baseline hazard.

means Means of the columns of the design matrix.

w.means Weighted (against exposure time) means of covariates; weighted relative frequencies of levels of factors.
mlreg

\begin{itemize}
\item n \hspace{2em} Number of spells in indata (possibly after removal of cases with NA's).
\item events \hspace{2em} Number of events in data.
\item terms \hspace{2em} Used by extractor functions.
\item assign \hspace{2em} Used by extractor functions.
\item wald.test \hspace{2em} The Wart test statistic (at the initial value).
\item y \hspace{2em} The Surv vector.
\item isF \hspace{2em} Logical vector indicating the covariates that are factors.
\item covars \hspace{2em} The covariates.
\item ttr \hspace{2em} Total Time at Risk.
\item levels \hspace{2em} List of levels of factors.
\item formula \hspace{2em} The calling formula.
\item call \hspace{2em} The call.
\item bootstrap \hspace{2em} The bootstrap sample, if requested on input.
\item sigma \hspace{2em} Present if a frailty model is fitted. Equals the estimated frailty standard deviation.
\item sigma.sd \hspace{2em} The standard error of the estimated frailty standard deviation.
\item method \hspace{2em} The method.
\item convergence \hspace{2em} Did the optimization converge?
\item fail \hspace{2em} Did the optimization fail? (Is NULL if not).
\end{itemize}

**Warning**

The use of rs is dangerous, see note above. It can however speed up computing time.

**Note**

This function starts by creating risksets, if no riskset is supplied via rs, with the aid of risksets. This latter mechanism fails if there are any NA's in the data! Note also that it depends on stratification, so rs contains information about stratification. Giving another strata variable in the formula is an error. The same is ok, for instance to supply stratum interactions.

Note further that mlreg is deprecated. coxreg should be used instead.

**Author(s)**

Göran Broström

**References**


**See Also**

coxreg, risksets
Examples

```r
dat <- data.frame(time = c(4, 3, 1, 1, 2, 2, 3),
                  status = c(1, 1, 1, 0, 1, 1, 0),
                  x = c(0, 2, 1, 1, 1, 0, 0),
                  sex = c(0, 0, 0, 0, 1, 1, 1))
mlreg(Surv(time, status) ~ x + strata(sex), data = dat) # stratified model
# Same as:
rs <- risksets(Surv(dat$time, dat$status), strata = dat$sex)
mlreg(Surv(time, status) ~ x, data = dat, rs = rs) # stratified model
```

---

**mort**

*Male mortality in ages 40-60, nineteenth century*

Description

Males born in the years 1800-1820 and surviving at least 40 years in the parish Skellefteå in northern Sweden are followed from their fortieth birthday until death or the sixtieth birthday, whichever comes first.

Usage

```r
data(mort)
```

Format

A data frame with 2058 observations on the following 6 variables.

- `id` Personal identification number.
- `enter` Start of duration. Measured in years since the fortieth birthday.
- `exit` End of duration. Measured in years since the fortieth birthday.
- `event` A logical vector indicating death at end of interval.
- `birthdate` The birthdate in decimal form.
- `ses` Socio-economic status, a factor with levels `lower`, `upper`

Details

The interesting explanatory covariate is `ses` (socioeconomic status), which is a time-varying covariate. This explains why several individuals are represented by more than one record each. Left truncation and right censoring are introduced this way.

Note

This data set is also known, and accessible, as `male.mortality`
Source

Data is coming from The Demographic Data Base, Umea University, Umeå, Sweden.

References

http://www.ddb.umu.se

Examples

```r
data(mort)
coxreg(Surv(enter, exit, event) ~ ses, data = mort)
```

oldmort

Old age mortality, Sundsvall, Sweden, 1860-1880.

Description

The data consists of old age life histories from 1 January 1860 to 31 December 1880, 21 years. Only (parts of) life histories above age 60 is considered.

Usage

```r
data(oldmort)
```

Format

A data frame with 6508 observations from 4603 persons on the following 13 variables.

- **id**: Identification number.
- **enter**: Start age for the interval.
- **exit**: Stop age for the interval.
- **event**: Indicator of death; equals TRUE if the person died at the end of the interval, FALSE otherwise.
- **birthdate**: Birthdate as a real number (i.e., "1765-06-27" is 1765.490).
- **m.id**: Mother's identification number.
- **f.id**: Father's identification number.
- **sex**: Gender, a factor with levels male female
- **civ**: Civil status, a factor with levels unmarried married widow
- **ses.50**: Socio-economic status at age 50, a factor with levels middle unknown upper farmer lower
- **birthplace**: A factor with levels parish region remote
- **imr.birth**: Infant mortality rate at birth in the region of birth
- **region**: Subregion of Sundsvall, a factor with levels town industry rural
The Sundsvall area in mid-Sweden was during the 19th century a fast growing forest industry. At the end of the century, it was one of the largest sawmill area in Europe. The town Sundsvall is fast growing part of the region and center for the commerce.

Source

The Demographic Data Base, Umeå University, Sweden.

References


Examples

data(oldmort)
summary(oldmort)
### maybe str(oldmort); plot(oldmort) ...

---

Pch  

The Piecewise Constant Hazards distribution.

Description

Density, distribution function, quantile function, hazard function, cumulative hazard function, mean, and random generation for the Piecewise Constant Hazards (pch) distribution.

Usage

ppch(q, cuts, levels, lower.tail = TRUE, log.p = FALSE)
dpch(x, cuts, levels, log = FALSE)
hpch(x, cuts, levels, log = FALSE)
Hpch(x, cuts, levels, log.p = FALSE)
qpch(p, cuts, levels, lower.tail = TRUE, log.p = FALSE)
mpch(cuts, levels)
rpch(n, cuts, levels)

Arguments

cuts  Vector of cut points defining the intervals where the hazard function is constant.
levels Vector of levels (values of the hazard function).
lower.tail logical; if TRUE (default), probabilities are \( P(X \leq x) \), otherwise, \( P(X > x) \).
x, q vector of quantiles.
p vector of probabilities.
log, log.p  logical; if TRUE, probabilities p are given as log(p).
n  number of observations. If length(n) > 1, the length is taken to be the number
required.

Details
The pch distribution has a hazard function that is piecewise constant on intervals defined by cut-
points

\[ 0 < c_1 < \cdots < c_n < \infty, n \geq 0 \]

If n = 0, this reduces to an exponential distribution.

Value
dpch gives the density, ppch gives the distribution function, qpch gives the quantile function, hpch
gives the hazard function, Hpch gives the cumulative hazard function, mpch gives the mean, and
rpch generates random deviates.

Note
the parameter levels must have length at least 1, and the number of cut points must be one less
than the number of levels.

perstat

DESCRIPTION
Calculates occurrence / exposure rates for time periods given by period and for ages given by age.

USAGE
perstat(surv, period, age = c(0, 200))

ARGUMENTS
Surv  An (extended) surv object (4 columns with enter, exit, event, birthdate)
period A vector of dates (in decimal form)
age    A vector of length 2; lowest and highest age

VALUE
A list with components
events  No. of events in each time period.
exposure Exposure times in each period.
intensity events / exposure
phfunc

Author(s)
Göran Broström

See Also
piecewise

phfunc Loglikelihood function of a proportional hazards regression

Description
Calculates minus the log likelihood function and its first and second order derivatives for data from a Weibull regression model.

Usage
phfunc(beta = NULL, lambda, p, X = NULL, Y, offset = rep(0, length(Y)), ord = 2, pfixed = FALSE, dist = "weibull")

Arguments
beta Regression parameters
lambda The scale parameter
p The shape parameter
X The design (covariate) matrix.
Y The response, a survival object.
offset Offset.
ord ord = 0 means only loglihood, 1 means score vector as well, 2 loglihood, score and hessian.
pfixed Logical, if TRUE the shape parameter is regarded as a known constant in the calculations, meaning that it is not cosidered in the partial derivatives.
dist Which distribtion? The default is "weibull", with the alternatives "loglogistic" and "lognormal".

Details
Note that the function returns log likelihood, score vector and minus hessian, i.e. the observed information. The model is

\[ S(t; p, \lambda, \beta, z) = S_0((t/\lambda)^p)e^{z\beta} \]
Value

A list with components

- \( f \) The log likelihood. Present if \( \text{ord} \geq 0 \)
- \( fp \) The score vector. Present if \( \text{ord} \geq 1 \)
- \( fpp \) The negative of the hessian. Present if \( \text{ord} \geq 2 \)

Author(s)

Göran Broström

See Also

- \( \text{phreg} \)

Description

Proportional hazards model with parametric baseline hazard(s). Allows for stratification with different scale and shape in each stratum, and left truncated and right censored data.

Usage

\[
\text{phreg}(\text{formula} = \text{formula(data)}, \text{data} = \text{parent.frame()},
\text{na.action} = \text{getOption("na.action")}, \text{dist} = \text{"weibull"}, \text{cuts} = \text{NULL},
\text{init}, \text{shape} = 0, \text{param} = \text{c("canonical", \"rate\")}, \text{control} = \text{list(eps}
\text{ = 1e-08, maxiter = 20, trace = FALSE), singular.ok = TRUE},
\text{model} = \text{FALSE}, x = \text{FALSE}, y = \text{TRUE}, \text{center} = \text{TRUE})
\]

Arguments

- \( \text{formula} \) a formula object, with the response on the left of a \( \sim \) operator, and the terms on the right. The response must be a survival object as returned by the \text{Surv} function.
- \( \text{data} \) a data.frame in which to interpret the variables named in the formula.
- \( \text{na.action} \) a missing-data filter function, applied to the model.frame, after any subset argument has been used. Default is \text{options()$na.action}.
- \( \text{dist} \) Which distribution? Default is "weibull", with the alternatives "ev" (Extreme value), "gompertz", "pch" (piecewise constant hazards function), "loglogistic" and "lognormal". A special case like the exponential can be obtained by choosing "weibull" in combination with shape = 1, or "pch" without cuts.
- \( \text{cuts} \) Only used with \( \text{dist} = \text{"pch"} \). Specifies the points in time where the hazard function jumps. If omitted, an exponential model is fitted.
**init**
vector of initial values of the iteration. Default initial value is zero for all variables.

**shape**
If positive, the shape parameter is fixed at that value (in each stratum). If zero or negative, the shape parameter is estimated. If more than one stratum is present in data, each stratum gets its own estimate.

**param**
Applies only to the Gompertz distribution: "canonical" is defined in the description of the Gompertz distribution; "rate" transforms scale to 1/log(scale), giving the same parametrization as in Stata and SAS. The latter thus allows for a negative rate, or a "cure" (Gompertz) model. The default is "canonical"; if this results in extremely large scale and/or shape estimates, consider trying "rate".

**control**
a list with components eps (convergence criterion), maxiter (maximum number of iterations), and silent (logical, controlling amount of output). You can change any component without mention the other(s).

**singular.ok**
Not used.

**model**
Not used.

**x**
Return the design matrix in the model object?

**y**
Return the response in the model object?

**center**
Logical, only affects plotting. Results are reported as is, without centering. See Details.

### Details

The parameterization is the same as in `coxreg` and `coxph`, but different from the one used by `survreg` (which is not a proportional hazards modelling function). The model is

\[
S(t; a, b, \beta, z) = S_0\left(\frac{t}{b}\right)^a \exp((z - \text{mean}(z))\beta)
\]

where \(S_0\) is some standardized survivor function.

If center = TRUE (default), graphs show the "baseline" distribution at the means of (continuous) covariates, and for the reference category in case of factors (avoiding representing "flying pigs"). If center = FALSE the baseline distribution is at the value zero of all covariates. It is usually a good idea to use center = FALSE in combination with "precentering" of covariates, that is, subtracting a reference value, ideally close to the center of the covariate distribution. In that way, the "reference" will be the same for all subsets of the data.

### Value

A list of class c("phreg","coxreg") with components

**coefficients**
Fitted parameter estimates.

**cuts**
Cut points for the "pch" distribution. NULL otherwise.

**hazards**
The estimated constant levels in the case of the "pch" distribution. NULL otherwise.

**var**
Covariance matrix of the estimates.

**loglik**
Vector of length two; first component is the value at the initial parameter values, the second component is the maximized value.
score
  The score test statistic (at the initial value).

linear.predictors
  The estimated linear predictors.

means
  Means of the columns of the design matrix, except those columns corresponding to a factor level, if center = TRUE. Otherwise all zero.

w.means
  Weighted (against exposure time) means of covariates; weighted relative frequencies of levels of factors.

n
  Number of spells in indata (possibly after removal of cases with NA's).

events
  Number of events in data.

terms
  Used by extractor functions.

assign
  Used by extractor functions.

wald.test
  The Wald test statistic (at the initial value).

y
  The Surv vector.

isF
  Logical vector indicating the covariates that are factors.

covars
  The covariates.

ttr
  Total Time at Risk.

levels
  List of levels of factors.

formula
  The calling formula.

call
  The call.

method
  The method.

convergence
  Did the optimization converge?

fail
  Did the optimization fail? (Is NULL if not).

pfixed
  TRUE if shape was fixed in the estimation.

Warning
  The lognormal and loglogistic distributions are included on an experimental basis for the moment. Use with care, results may be unreliable!

  The gompertz distribution has an exponentially increasing hazard function under the canonical parametrization. This may cause instability in the convergence of the fitting algorithm in the case of near-exponential data. It may be resolved by using param = "rate".

Note
  The lognormal and loglogistic baseline distributions are extended to a three-parameter family by adding a "proportionality" parameter (multiplying the baseline hazard function). The log of the estimated parameter turns up as 'Intercept' in the printed output. The reason for this extension is that the standard lognormal and loglogistic distributions are not closed under proportional hazards.

Author(s)
  Göran Broström
See Also

coxreg, check.dist, link{aftreg}.

Examples

data(mort)
fit <- phreg(Surv(enter, exit, event) ~ ses, data = mort)
fit
plot(fit)
fit.cr <- coxreg(Surv(enter, exit, event) ~ ses, data = mort)
check.dist(fit.cr, fit)

phreg.fit  Parametric proportional hazards regression

Description

This function is called by phreg, but it can also be directly called by a user.

Usage

phreg.fit(X, Y, dist, strata, offset, init, shape, control, 
          center = NULL)

Arguments

X        The design (covariate) matrix.
Y        A survival object, the response.
dist     Which baseline distribution?
strata   A stratum variable.
offset   Offset.
init     Initial regression parameter values.
shape    If positive, a fixed value of the shape parameter in the distribution. Otherwise, 
          the shape is estimated.
control  Controls convergence and output.
center   Deprecated (not used). Kept for backward capability. Results are reported as is, 
          no centering.

Details

See phreg for more detail.
Value

coefficients Estimated regression coefficients plus estimated scale and shape coefficients, sorted by strata, if present.
var Variance-covariance matrix
loglik Vector of length 2. The first component is the maximized loglihood with only scale and shape in the model, the second the final maximum.
score Score test statistic at initial values
linear.predictors Linear predictors for each interval.
means Means of the covariates
conver TRUE if convergence
fail TRUE if failure
iter Number of Newton-Raphson iterates.
n.strata The number of strata in the data.

Author(s)
Göran Broström

See Also
phreg

Description
Calculate piecewise hazards, no. of events, and exposure times in each interval indicated by cutpoints.

Usage
piecewise(enter, exit, event, cutpoints)

Arguments
enter Left interval endpoint
exit Right interval endpoint
event Indicator of event
cutpoints Vector of cutpoints

Details
Exact calculation.
Value

A list with components

- **events**: Vector of number of events
- **exposure**: Vector of total exposure time
- **intensity**: Vector of hazards, intensity == events / exposure

Author(s)

Göran Broström

See Also

`perstat`

---

**plot.aftreg**  
*Plots output from an AFT regression*

---

Description

Just a simple plot of the hazard (cumulative hazard, density, survival) functions for each stratum.

Usage

```r
## S3 method for class 'aftreg'
plot(x, fn = c("haz", "cum", "den", "sur"),
     main = NULL, xlim = NULL, ylim = NULL, xlab = "Duration",
     ylab = "", col, lty, printLegend = TRUE, new.data = x$means, ...)
```

Arguments

- **x**: A `aftreg` object
- **fn**: Which functions should be plotted! Default is all. They will scroll by, so you have to take care of explicitly what you want to be produced. See, eg, `par(mfrow = ...)`
- **main**: Header for the plot
- **xlim**: x limits
- **ylim**: y limits
- **xlab**: x label
- **ylab**: y label
- **col**: Colors?
- **lty**: Line types?
- **printLegend**: Should legend be printed? Default is yes.
- **new.data**: At which covariate values?
- **...**: Extra parameters passed to `plot`
Details

The plot is drawn at the mean values of the covariates, by default.

Value

No return value.

Author(s)

Göran Broström

See Also

aftreg

Examples

```r
y <- rllogis(40, shape = 1, scale = 1)
x <- rep(c(1,1,2,2), 10)
fit <- aftreg(Surv(y, rep(1, 40)) ~ x, dist = "loglogistic")
plot(fit)
```

Description

A plot of a baseline function of a coxreg fit is produced, one curve for each stratum.

Usage

```r
## S3 method for class 'coxreg'
plot(x, fn = c("cum", "surv", "log", "loglog"),
     fig = TRUE, xlim = NULL, ylim = NULL, main = NULL,
     xlab = "Duration", ylab = "", col, lty, printLegend = TRUE,
     newdata = NULL, ...)
```

Arguments

- `x` A coxreg object
- `fn` What should be plotted? Default is "cumhaz", and the other choices are "surv", "log", and "loglog".
- `fig` logical. If TRUE the plot is actually drawn, otherwise only the coordinates of the curve(s) are returned.
- `xlim` Start and end of the x axis.
plot.hazdata

\begin{verbatim}
ylim Start and end of the y axis.
main A headline for the plot
xlab Label on the x axis.
ylab Label on the y axis.
col Color of the curves. Defaults to 'black'.
lty Line type(s).
printLegend Either a logical or a text string: if TRUE, a legend is printed at a default place, if FALSE, no legend is printed. Otherwise, if a text string, it should be one of "bottomleft", "bottomright", "topleft", etc., see legend for all possible choices.
newdata Not used
\end{verbatim}

\textbf{Value}

An object of class \texttt{hazdata} containing the coordinates of the curve(s).

\textbf{Description}

Baseline hazards estimates.

\textbf{Usage}

\begin{verbatim}
## S3 method for class 'hazdata'
plot(x, strata = NULL, fn = c("cum", "surv", "log", "loglog"), fig = TRUE, xlim = NULL, ylim = NULL, main = NULL, xlab = NULL, ylab = NULL, col = "black", lty = 1, printLegend = TRUE, where = NULL, ...)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
\item \texttt{x} A hazdata object, typically the 'hazards' element in the output from \texttt{link{coxreg}} with \texttt{hazards = TRUE}.
\item \texttt{strata} Stratum names if there are strata present.
\item \texttt{fn} Which type of plot?
\item \texttt{fig} Should a plot actually be produced? Default is TRUE.
\item \texttt{xlim} Horizontal plot limits. If NULL, calculated by the function.
\item \texttt{ylim} Vertical plot limits. If NULL, set to \texttt{c(0,1)}
\item \texttt{main} A heading for the plot.
\item \texttt{xlab} Label on the x axis.
\end{itemize}
Details

It is also possible to have as first argument an object of type "coxreg", given that it contains a component of type "hazdata".

Value

A list where the elements are two-column matrices, one for each stratum in the model. The first column contains risktimes, and the second the y coordinates for the requested curve(s).

Author(s)

Göran Broström

Examples

```r
time0 <- numeric(50)
group <- c(rep(0, 25), rep(1, 25))
time1 <- rexp( 50, exp(group) )
event <- rep(1, 50)
fit <- coxreg(Surv(time0, time1, event) ~ strata(group), hazards = TRUE)
plot(fit$hazards)
```

Description

Plot(s) of the hazard, density, cumulative hazards, and/or the survivor function(s) for each stratum.

Usage

```r
## S3 method for class 'phreg'
plot(x, fn = c("haz", "cum", "den", "sur"),
     main = NULL, xlim = NULL, ylim = NULL, xlab = "Duration",
     ylab = "", col, lty, printLegend = TRUE, new.data = NULL, ...)
```
Arguments

- **x**: A `phreg` object
- **fn**: Which functions should be plotted! Default is all. They will scroll by, so you have to take care explicitly what you want to be produced. See, eg, `par(mfrow = ...)`
- **main**: Header for the plot
- **xlim**: x limits
- **ylim**: y limits
- **xlab**: x label
- **ylab**: y label
- **col**: Color(s) for the curves. Defaults to black.
- **lty**: Line type for the curve(s). Defaults to `1:(No. of strata)`.
- **printLegend**: Logical, or character ("topleft", "bottomleft", "topright" or "bottomright"); if `TRUE` or character, a legend is added to the plot if the number of strata is two or more.
- **new.data**: Now deprecated; reference hazard is given by the fit; either zero or the means all covariates, and (always) the reference category for factors.
- ...: Extra parameters passed to `plot` and `lines`.

Value

No return value.

Author(s)

Göran Broström

See Also

- `phreg`

Examples

```r
y <- rlogis(40, shape = 1, scale = 1)
x <- rep(c(1,1,2,2), 10)
fit <- phreg(Surv(y, rep(1, 40)) ~ x, dist = "loglogistic")
plot(fit)
```
plot.weibreg

Plots output from a Weibull regression

Description

Plot(s) of the hazard, density, cumulative hazards, and/or the survivor function(s) for each stratum.

Usage

## S3 method for class 'weibreg'
plot(x, fn = c("haz", "cum", "den", "sur"),
     main = NULL, xlim = NULL, ylim = NULL, xlab = NULL,
     ylab = NULL, new.data = x$means, ...)

Arguments

x
A weibreg object

fn
Which functions should be plotted! Default is all. They will scroll by, so you have to take care explicitly what you want to be produced. See, eg, par(mfrow = ...)

main
Header for the plot

xlim
x limits

ylim
y limits

xlab
x label

ylab
y label

new.data
At which covariate values?

...
Extra parameters passed to 'plot'

Details

The plot is drawn at the mean values of the covariates.

Value

No return value

Author(s)

Göran Broström

See Also

phreg, weibreg
print.aftreg

Examples

```r
y <- rweibull(4, shape = 1, scale = 1)
x <- c(1,1,2,2)
fit <- weibreg(Surv(y, c(1,1,1,1)) ~ x)
plot(fit)
```

Description

The hazard, the cumulative hazard, the density, and the survivor baseline functions are plotted.

Usage

```r
## S3 method for class 'aftreg'
print(x, digits = max(options()$digits - 4, 3), ...)
```

Arguments

- `x` A `aftreg` object
- `digits` Precision in printing
- `...` Not used.

Value

No value is returned.

Note

Doesn’t work for threeway or higher order interactions. Use `print.coxph` in that case.

Author(s)

Göran Broström

See Also

`phreg`, `print.coxph`
print.coxreg

**Prints coxreg objects**

**Description**

More "pretty-printing" than `print.coxph`, which is a fall-back for 'difficult' objects.

**Usage**

```r
## S3 method for class 'coxreg'
print(x, digits = max(options()$digits - 4, 3), ...)
```

**Arguments**

- `x` A `coxreg` object, typically the result of running `coxreg`
- `digits` Output format.
- `...` Other arguments.

**Details**

Doesn't work with three-way and higher interactions, in which case `print.coxph` is used. Prints also output from `mlreg`.

**Value**

No value is returned.

**Author(s)**

Göran Broström

**See Also**

`coxreg, print.coxph`

---

print.phreg

**Prints phreg objects**

**Description**

The hazard, the cumulative hazard, the density, and the survivor baseline functions are plotted.

**Usage**

```r
## S3 method for class 'phreg'
print(x, digits = max(options()$digits - 4, 3), ...)
```
print.risksets

Arguments

x  A phreg object
digits  Precision in printing
...  Not used.

Value

No value is returned.

Note

Doesn’t work for three-way or higher order interactions. Use print.coxph in that case.

Author(s)

Göran Broström

See Also

phreg, print.coxph

print.risksets  Prints a summary of the content of a set of risk sets.

Description

Given the output from risksets, summary statistics are given for it.

Usage

## S3 method for class 'risksets'
print(x, ...)

Arguments

x  An object of class 'risksets'.
...  Not used for the moment.

Value

No value is returned; the function prints summary statistics of risk sets.

Note

There is no summary.risksets yet. On the TODO list.
Author(s)
  Göran Broström

See Also
  risksets

Examples

rs <- with(mort, risksets(Surv(enter, exit, event)))
print(rs)

print.weibreg  Prints weibreg objects

Description
  The hazard, the cumulative hazard, the density, and the survivor baseline functions are plotted.

Usage

## S3 method for class 'weibreg'
print(x, digits = max(options()$digits - 4, 3), ...)

Arguments
  x  A weibreg object
  digits  Precision in printing
  ...  Not used.

Value
  No value is returned.

Note
  Doesn’t work for threeway or higher order interactions. Use print.coxph in that case.

Author(s)
  Göran Broström

See Also
  weibreg.print.coxph
### risksets

**Finds the compositions and sizes of risk sets**

#### Description

Focus is on the risk set composition just prior to a failure.

#### Usage

```r
risksets(x, strata = NULL, max.survs = NULL, members = TRUE)
```

#### Arguments

- **x**: A `Surv` object.
- **strata**: Stratum indicator.
- **max.survs**: Maximum number of survivors in each risk set. If smaller than the 'natural number', survivors are sampled from the present ones. No sampling if missing.
- **members**: If TRUE, all members of all risk sets are listed in the resulting list, see below.

#### Details

If the input argument max.survs is left alone, all survivors are accounted for in all risk sets.

#### Value

A list with components

- **antrs**: No. of risk sets in each stratum. The number of strata is given by `length(antrs)`.
- **risktimes**: Ordered distinct failure time points.
- **eventset**: If 'members' is TRUE, a vector of pointers to events in each risk set, else NULL.
- **riskset**: If 'members' is TRUE, a vector of pointers to the members of the risk sets, in order. The 'n.events' first are the events. If 'members' is FALSE, 'riskset' is NULL.
- **size**: The sizes of the risk sets.
- **n.events**: The number of events in each risk set.
- **sample_fraction**: The sampling fraction of survivors in each risk set.

#### Note

can be used to "sample the risk sets".

#### Author(s)

Göran Broström
See Also
table.events, coxreg.

Examples

```r
enter <- c(0, 1, 0, 0)
exit <- c(1, 2, 3, 4)
event <- c(1, 1, 1, 0)
risksets(Surv(enter, exit, event))
```

scania

Old age mortality, Scania, southern Sweden, 1813-1894.

Description
The data consists of old age life histories from 1 January 1813 to 31 December 1894. Only (parts of) life histories above age 50 is considered.

Usage
data(scania)

Format
A data frame with 1931 observations from 1931 persons on the following 9 variables.

- `id` Identification number (enumeration).
- `enter` Start age for the interval.
- `exit` Stop age for the interval.
- `event` Indicator of death; equals TRUE if the person died at the end of the interval, FALSE otherwise.
- `birthdate` Birthdate as a real number (i.e., "1765-06-27" is 1765.490).
- `sex` Gender, a factor with levels male female.
- `parish` One of five parishes in Scania, coded 1, 2, 3, 4, 5. Factor.
- `ses` Socio-economic status at age 50, a factor with levels upper and lower.
- `immigrant` a factor with levels no region and yes.

Details
The Scanian area in southern Sweden was during the 19th century a mainly rural area.

Source
The Scanian Economic Demographic Database, Lund University, Sweden.
summary.aftreg

References

http://www.ed.lu.se/databases

Examples

data(scania)
summary(scania)

summary.aftreg      Prints aftreg objects

Description

This is the same as print.aftreg

Usage

## S3 method for class 'aftreg'
summary(object, ...)

Arguments

  object    A aftreg object
  ...       Additional ...

Author(s)

Göran Broström

See Also

print.coxreg

Examples

## The function is currently defined as
function (object, ...)
print(object)
**summary.coxreg**  
*Prints coxreg objects*

**Description**

This is the same as `print.coxreg`

**Usage**

```r
## S3 method for class 'coxreg'
summary(object, ...)
```

**Arguments**

- `object`: A coxreg object
- `...`: Additional ...

**Author(s)**

Göran Broström

**See Also**

`print.coxreg`

**Examples**

```r
## The function is currently defined as
def print_function(object, ...)
print(object)
```

---

**summary.phreg**  
*Prints phreg objects*

**Description**

This is the same as `print.phreg`

**Usage**

```r
## S3 method for class 'phreg'
summary(object, ...)
```
summary.weibreg

Arguments

  object  A phreg object
  ...     Additional ...

Author(s)

  Göran Broström

See Also

  print.coxreg

Examples

  ## The function is currently defined as
  function (object, ...)
  print(object)

summary.weibreg                Prints a weibreg object

Description

  This is the same as print.weibreg

Usage

  ## S3 method for class 'weibreg'
  summary(object, ...)

Arguments

  object  A weibreg object
  ...    Additional ...

Author(s)

  Göran Broström

See Also

  print.weibreg
Examples

```r
## The function is currently defined as
function (object, ...) 
print(object)
```

Description

Given a survival object, (a matrix with two or three columns) and a set of specified cut times, split
each record into multiple subrecords at each cut time. The new survival object will be in 'counting
process' format, with an enter time, exit time, and event status for each record.

Usage

SurvSplit(Y, cuts)

Arguments

Y

A survival object, a matrix with two or three columns.

cuts

The cut points, must be strictly positive and distinct.

Value

A list with components

Y

The new survival object with three columns, i.e., in 'counting process' form.

ivl

Interval No., starting from leftmost, (0, cuts[1]) or similar.

idx

Row number for original Y row.

Note

This function is used in phreg for the piecewise constant hazards model. It uses age.window for
each interval.

Author(s)

Göran Broström

See Also

survSplit, age.window.
Examples

```r
## Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##-- or do help(data=index) for the standard data sets.

## The function is currently defined as
function(Y, cuts){
  if (NCOL(Y) == 2) Y <- cbind(rep(0, NROW(Y)), Y)
  indat <- cbind(Y, 1:NROW(Y), rep(-1, NROW(Y)))
  colnames(indat) <- c("enter", "exit", "event", "idx", "ivl")
  n <- length(cuts)
  cuts <- sort(cuts)
  if ((cuts[1] <= 0) || (cuts[n] == Inf))
    stop("cuts must be positive and finite.")
  cuts <- c(0, cuts, Inf)
  n <- n + 1
  out <- list()
  indat <- as.data.frame(indat)
  for (i in 1:n){
    out[[i]] <- age.window(indat, cuts[i:(i+1)])
    out[[i]]$ivl <- i
    out[[i]] <- t(out[[i]])
  }
  Y <- matrix(unlist(out), ncol = 5, byrow = TRUE)
  colnames(Y) <- colnames(indat)
  list(Y = Y[, 1:3],
       ivl = Y[, 5],
       idx = Y[, 4])
}
```

---

**swe07**

*Swedish population and deaths in ages 61–80, 2007.*

**Description**

The Swedish population and No. of deaths by age and sex in the ages 61–80. Data from the year 2007.

**Usage**

data(swe07)

**Format**

A data frame with 80 rows and five variables.

- **pop** Average population size during the year 2007 by age and sex.
deaths  Number of deaths by age and sex during the year 2007.
sex   Sex.
age   Age.
log.pop The logarithm of the first variable, pop. Included for convenience, may be used as an offset in a Poisson regression.

Details

The average population is calculated as the mean of the population 1 January 2007 and 1 January 2008.

Source

Data is taken from Statistics Sweden.

References

http://www.scb.se

Examples

data(swe07)
fit <- glm(deaths ~ offset(log.pop) + sex * as.factor(age), family = poisson, data = swe07)
drop1(fit, test = "Chisq")  ## Proportional hazards?

---

**table.events**

*Calculating failure times, risk set sizes and No. of events in each risk set*

**Description**

From input data of the 'interval' type, with an event indicator, summary statistics for each risk set (at an event time point) are calculated.

**Usage**

table.events(enter = rep(0, length(exit)), exit, event, strict = TRUE)

**Arguments**

- **enter**: Left truncation time point.
- **exit**: End time point, an event or a right censoring.
- **event**: Event indicator.
- **strict**: If TRUE, then tabulating is not done after a time point where all individuals in a riskset failed.
toBinary

Value
A list with components

- times: Ordered distinct event time points.
- events: Number of events at each event time point.
- riskset.sizes: Number at risk at each event time point.

Author(s)
Göran Broström

See Also
risksets

Examples

```r
exit = c(1,2,3,4,5)
event = c(1,1,0,1,1)
table.events(exit = exit, event = event)
```

Description
The result of the transformation can be used to do survival analysis via logistic regression. If the cloglog link is used, this corresponds to a discrete time analogue to Cox’s proportional hazards model.

Usage

toBinary(dat, surv = c("enter", "exit", "event"), strats,
          max.survs = NROW(dat))

Arguments

- `dat`: A data frame with three variables representing the survival response. The default is that they are named enter, exit, and event.
- `surv`: A character vector with the names of the three variables representing survival.
- `strats`: An eventual stratification variable.
- `max.survs`: Maximal number of survivors per risk set. If set to a (small) number, survivors are sampled from the risk sets.
Details

toBinary calls risksets in the eha package.

Value

Returns a data frame expanded risk set by risk set. The three "survival variables" are replaced by a variable named event (which overwrites an eventual variable by that name in the input). Two more variables are created, riskset and orig.row.

- **event**: Indicates an event in the corresponding risk set.
- **riskset**: Factor (with levels 1, 2, ... ) indicating risk set.
- **risktime**: The 'risktime' (age) in the corresponding riskset.
- **orig.row**: The row number for this item in the original data frame.

Note

The survival variables must be three. If you only have `exit` and `event`, create a third containing all zeros.

Author(s)

Göran Broström

See Also

coxreg, glm.

Examples

```r
enter <- rep(0, 4)
exit <- 1:4
event <- rep(1, 4)
z <- rep(c(-1, 1), 2)
dat <- data.frame(enter, exit, event, z)
binDat <- toBinary(dat)
dat
binDat
coxreg(Surv(enter, exit, event) ~ z, method = "ml", data = dat)
## Same as:
summary(glm(event ~ z + riskset, data = binDat, family = binomial(link = cloglog)))
```
toDate

Convert time in years since "0000-01-01" to a date.

Description

This function uses \texttt{as.Date} and a simple linear transformation.

Usage

toDate(times)

Arguments

times a vector of durations

Value

A vector of dates as character strings of the type "1897-05-21".

Author(s)

Göran Broström

See Also

toTime

Examples

##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random, 
##--or do help(data=index) for the standard data sets. 
toDate(1897.357)


toTime

Calculate duration in years from "0000-01-01" to a given date

Description

Given a vector of dates, the output is a vector of durations in years since "0000-01-01".

Usage

toTime(dates)
toTpch

Description
Transform a "survival" data frame Surv(enter, exit, event) to tabular form with 'event' and 'exposure' and aggregating

Usage

toTpch(dat, cuts)

Arguments

dat The survival data frame.
cuts Vector defining the age periods of constant hazard.
Description

Proportional hazards regression with piecewise constant hazards and tabular data.

Usage

tpchreg(formula, cuts, data)

Arguments

formula a formula
cuts intervals for constant hazard.
data a data frame with event, exposure, age plus covariates

Description

Proportional hazards model with baseline hazard(s) from the Weibull family of distributions. Allows for stratification with different scale and shape in each stratum, and left truncated and right censored data.

Usage

weibreg(formula = formula(data), data = parent.frame(),
na.action = getOption("na.action"), init, shape = 0,
control = list(eps = 1e-04, maxiter = 10, trace = FALSE),
singular.ok = TRUE, model = FALSE, x = FALSE, y = TRUE,
center = TRUE)

Arguments

formula a formula object, with the response on the left of a ~ operator, and the terms on the right. The response must be a survival object as returned by the Surv function.
data a data.frame in which to interpret the variables named in the formula.
na.action a missing-data filter function, applied to the model.frame, after any subset argument has been used. Default is options()$na.action.
**Start**

`init` vector of initial values of the iteration. Default initial value is zero for all variables.

`shape` If positive, the shape parameter is fixed at that value (in each stratum). If zero or negative, the shape parameter is estimated. If more than one stratum is present in data, each stratum gets its own estimate.

`control` a list with components `eps` (convergence criterion), `maxiter` (maximum number of iterations), and `silent` (logical, controlling amount of output). You can change any component without mention the other(s).

`singular.ok` Not used.

`model` Not used.

`x` Return the design matrix in the model object?

`y` Return the response in the model object?

`center` Deprecated, and not used. Will be removed in the future.

**Details**

The parameterization is the same as in `coxreg` and `coxph`, but different from the one used by `survreg`. The model is

\[ h(t; a, b, \beta, z) = \left(\frac{b}{a/b}\right)^{a-1} \exp(z\beta) \]

This is in correspondence with Weibull. To compare regression coefficients with those from `survreg` you need to divide by estimated shape (\(\hat{a}\)) and change sign. The p-values and test statistics are however the same, with one exception; the score test is done at maximized scale and shape in `weibreg`.

This model is a Weibull distribution with shape parameter \(a\) and scale parameter \(b \exp(-z\beta/a)\)

**Value**

A list of class \(c(\text{"weibreg"}, \text{"coxreg"})\) with components

- `coefficients` Fitted parameter estimates.
- `var` Covariance matrix of the estimates.
- `loglik` Vector of length two; first component is the value at the initial parameter values, the second component is the maximized value.
- `score` The score test statistic (at the initial value).
- `linear.predictors` The estimated linear predictors.
- `means` Means of the columns of the design matrix.
- `w.means` Weighted (against exposure time) means of covariates; weighted relative frequencies of levels of factors.
- `n` Number of spells in indata (possibly after removal of cases with NA's).
- `events` Number of events in data.
- `terms` Used by extractor functions.
- `assign` Used by extractor functions.
wald.test  The Wald test statistic (at the initial value).
y  The Surv vector.
isF  Logical vector indicating the covariates that are factors.
covars  The covariates.
ttr  Total Time at Risk.
levels  List of levels of factors.
formula  The calling formula.
call  The call.
method  The method.
convergence  Did the optimization converge?
fail  Did the optimization fail? (Is NULL if not).
pfixed  TRUE if shape was fixed in the estimation.

Warning

The print method print.weibreg doesn’t work if threeway or higher order interactions are present.

Note further that covariates are internally centered, if center = TRUE, by this function, and this is not corrected for in the output. This affects the estimate of \( \log(\text{scale}) \), but nothing else. If you don’t like this, set center = FALSE.

Note

This function is not maintained, and may behave in unpredictable ways. Use phreg with dist = "weibull" (the default) instead! Will soon be declared deprecated.

Author(s)

Göran Broström

See Also

phreg, coxreg, print.weibreg

Examples

dat <- data.frame(time = c(4, 3, 1, 1, 2, 2, 3),
  status = c(1, 1, 1, 0, 1, 1, 0),
  x = c(0, 2, 1, 1, 0, 0),
  sex = c(0, 0, 0, 0, 1, 1, 1))
weibreg( Surv(time, status) ~ x + strata(sex), data = dat) #stratified model
Description

This function is called by \texttt{weibreg}, but it can also be directly called by a user.

Usage

\begin{verbatim}
weibreg.fit(X, Y, strata, offset, init, shape, control, center = TRUE)
\end{verbatim}

Arguments

\begin{itemize}
\item \textbf{X} \hspace{1cm} The design (covariate) matrix.
\item \textbf{Y} \hspace{1cm} A survival object, the response.
\item \textbf{strata} \hspace{1cm} A stratum variable.
\item \textbf{offset} \hspace{1cm} Offset.
\item \textbf{init} \hspace{1cm} Initial regression parameter values.
\item \textbf{shape} \hspace{1cm} If positive, a fixed value of the shape parameter in the Weibull distribution. Otherwise, the shape is estimated.
\item \textbf{control} \hspace{1cm} Controls convergence and output.
\item \textbf{center} \hspace{1cm} Should covariates be centered?
\end{itemize}

Details

See \texttt{weibreg} for more detail.

Value

\begin{itemize}
\item \textbf{coefficients} \hspace{1cm} Estimated regression coefficients plus estimated scale and shape coefficients, sorted by strata, if present.
\item \textbf{var} \hspace{1cm} Vector of length 2. The first component is the maximized loglihood with only scale and shape in the model, the second the final maximum.
\item \textbf{loglik} \hspace{1cm} Score test statistic at initial values
\item \textbf{linear.predictors} \hspace{1cm} Linear predictors for each interval.
\item \textbf{means} \hspace{1cm} Means of the covariates
\item \textbf{conver} \hspace{1cm} TRUE if convergence
\item \textbf{fail} \hspace{1cm} TRUE if failure
\item \textbf{iter} \hspace{1cm} Number of Newton-Raphson iterates.
\item \textbf{n.strata} \hspace{1cm} The number of strata in the data.
\end{itemize}
Weibull

Author(s)

Göran Broström

See Also

weibreg

Description

hweibull calculates the hazard function of a Weibull distribution, and Hweibull calculates the corresponding cumulative hazard function.

Usage

hweibull(x, shape, scale = 1, log = FALSE)

Arguments

x  Vector of quantiles.
shape  The shape parameter.
scale  The scale parameter, defaults to 1.
log  logical; if TRUE, the log of the hazard function is given.

Details

See dweibull.

Value

The (cumulative) hazard function, evaluated at x.

Author(s)

Göran Broström

See Also

pweibull
Examples

\begin{verbatim}
hweibull(3, 2, 1) dweibull(3, 2, 1) / pweibull(3, 2, 1, lower.tail = FALSE) Hweibull(3, 2, 1) -pweibull(3, 2, 1, log.p = TRUE, lower.tail = FALSE)
\end{verbatim}

Description

Calculates minus the log likelihood function and its first and second order derivatives for data from a Weibull regression model. Is called by \texttt{weibreg}.

Usage

```
wfunk(beta = NULL, lambda, p, X = NULL, Y, offset = rep(0, length(Y)), ord = 2, pfixed = FALSE)
```

Arguments

- `beta`: Regression parameters
- `lambda`: The scale parameter
- `p`: The shape parameter
- `X`: The design (covariate) matrix.
- `Y`: The response, a survival object.
- `offset`: Offset.
- `ord`: `ord = 0` means only loglikelihood, `1` means score vector as well, `2` loglikelihood, score and hessian.
- `pfixed`: Logical, if TRUE the shape parameter is regarded as a known constant in the calculations, meaning that it is not considered in the partial derivatives.

Details

Note that the function returns log likelihood, score vector and minus hessian, i.e. the observed information. The model is

\[ h(t; p, \lambda, \beta, z) = p/\lambda(t/\lambda)^{(p-1)} \exp(-(t/\lambda)^p) \exp(z\beta) \]

This is in correspondence with \texttt{dweibull}. 

---

**wfunk**

Loglihood function of a Weibull regression
Value

A list with components

- **f**: The log likelihood. Present if `ord >= 0`
- **fp**: The score vector. Present if `ord >= 1`
- **fpp**: The negative of the hessian. Present if `ord >= 2`

Author(s)

Göran Broström

See Also

- `weibreg`
Index

*Topic **cluster**
  toBinary, 63

*Topic **datasets**
  fert, 19
  infants, 23
  logrye, 27
  male.mortality, 31
  mort, 35
  oldmort, 36
  scania, 56
  swe07, 61

*Topic **distribution**
  check.dist, 9
  EV, 18
  Gompertz, 21
  Loglogistic, 25
  Lognormal, 26
  makeham, 30
  Pch, 37
  phfunc, 39
  wfunk, 72

*Topic **dplot**
  plot.aftreg, 45
  plot.phreg, 48
  plot.weibreg, 50

*Topic **manip**
  check.surv, 10
  cro, 16
  join.spells, 24
  SurvSplit, 60

*Topic **nonparametric**
  perstat, 38
  piecewise, 44

*Topic **printing**
  ltx, 28

*Topic **print**
  summary.aftreg, 57
  summary.coxreg, 58
  summary.phreg, 58

  summary.weibreg, 59

*Topic **regression**
  aftreg, 4
  aftreg.fit, 6
  coxreg, 11
  coxreg.fit, 14
  mlreg, 32
  phreg, 40
  phreg.fit, 43
  print.aftreg, 51
  print.phreg, 52
  print.weibreg, 54
  weibreg, 67
  weibreg.fit, 70

*Topic **summary**
  print.risksets, 53

*Topic **survival**
  aftreg, 4
  aftreg.fit, 6
  age.window, 7
  cal.window, 8
  check.surv, 10
  coxreg, 11
  coxreg.fit, 14
  geome.fit, 20
  join.spells, 24
  make.communal, 29
  mlreg, 32
  perstat, 38
  phfunc, 39
  phreg, 40
  phreg.fit, 43
  piecewise, 44
  plot.aftreg, 45
  plot.hazdata, 47
  plot.phreg, 48
  plot.weibreg, 50
  print.aftreg, 51
  print.coxreg, 52
INDEX

print.phreg, 52
print.weibreg, 54
risksets, 55
summary.aftreg, 57
summary.coxreg, 58
summary.phreg, 58
summary.weibreg, 59
summary.aftreg, 57
summary.coxreg, 58
summary.phreg, 58
summary.weibreg, 59
table.events, 62
toBinary, 63
toDate, 65
toTime, 65
weibreg, 67
weibreg.fit, 70
Weibull, 71
wfunk, 72

aftreg, 4, 6–9, 11, 25, 28–30, 46
aftreg.fit, 6
age.window, 7, 9, 60
cal.window, 8, 8, 30
check.dist, 9, 43
check.surv, 10, 25
coxph, 3, 12, 14, 29, 30, 41, 68
coxreg, 3, 6, 8–11, 11, 14–16, 21, 25, 28–30, 33, 34, 41, 43, 52, 56, 64, 68, 69
coxreg.fit, 14
cro, 16
dEV (EV), 18
dgompertz (Gompertz), 21
dllogis (Loglogistic), 25
dlnorm (Lognormal), 26
dmakeham (makeham), 30
dpch (Pch), 37
dweibull, 71, 72
eha (eha-package), 3
eha-defunct, 17
eha-package, 3
EV, 18
fert, 19
frail.fit, 20
geome.fit, 20
ghq (eha-defunct), 17
glm, 64
glmmboot (eha-defunct), 17
glmmbootFit (eha-defunct), 17
glmmML (eha-defunct), 17
Gompertz, 21, 41
gompertz (Gompertz), 21
HEV (EV), 18
hEV (EV), 18
Hgompertz (Gompertz), 21
hgomertz (Gompertz), 21
Hllogis (Loglogistic), 25
hllogis (Loglogistic), 25
Hlnorm (Lognormal), 26
hlnorm (Lognormal), 26
Hmakeham (makeham), 30
hmakeham (makeham), 30
Hpch (Pch), 37
hpch (Pch), 37
Hweibull (Weibull), 71
hweibull (Weibull), 71
infants, 23
join.spells, 11, 24
legend, 47
Llogis (Loglogistic), 25
Lnorm (Lognormal), 26
Loglogistic, 25
Lognormal, 26
logrye, 27
ltx, 28
make.communal, 29
makeham, 30
male.mortality, 31
match, 17
mlreg, 32, 52
mort, 35
mpch (Pch), 37
oldmort, 36
paste, 17
Pch, 37
pch (Pch), 37
perstat, 38, 45
pEV (EV), 18
pgompertz (Gompertz), 21
phfunc, 39
phreg, 6, 9, 10, 28, 40, 40, 43, 44, 49–51, 53, 60, 69
INDEX

phreg.fit, 43
toTpch, 66
piecewise, 39, 44
tpchreg, 67
pllogis (Loglogistic), 25
weibreg, 50, 54, 67, 70–73
plnorm (Lognormal), 26
Weibull, 68, 71
plot.aftreg, 45
weibreg, 50
plot.coxreg, 46
wpach, 72
plot.default, 48
plot.hazdata, 47
plot.phreg, 48
plot.weibreg, 50
plot.phreg, 48
print.aftreg, 51, 57
ppch (Pch), 37
print.aftreg, 51, 57
print.coxph, 51–54
print.coxreg, 52, 57–59
print.phreg, 52, 58
print.risksets, 53
print.risksets, 13, 14, 16, 20, 21, 34, 55, 63
print.weibreg, 54, 59, 69
print.coxreg, 52, 57–59
print.risksets, 53
print.weibreg, 54, 59, 69
print.phreg, 52, 58
pweibull, 71
qllogis (Loglogistic), 25
qEV (EV), 18
qlnorm (Lognormal), 26
qgompertz (Gompertz), 21
qmakeham (makeham), 30
qpch (Pch), 37
rEV (EV), 18
rgompertz (Gompertz), 21
risksets, 13, 14, 16, 20, 21, 34, 55, 63
rlogis (Loglogistic), 25
rnorm (Lognormal), 26
rlogis (Loglogistic), 25
rmakeham (makeham), 30
rpch (Pch), 37
scania, 56
summary.aftreg, 57
sumsummary.coxreg, 58
summary.phreg, 58
summary.coxreg, 58
summary.weibreg, 59
summary.phreg, 58
summary.weibreg, 59
Survreg, 4–6, 41, 68
SurvSplit, 60
survSplit, 60
swe07, 61

table.events, 56, 62
toBinary, 63
toDate, 65, 66
toTime, 63, 65