Package ‘eha’

January 12, 2021

Encoding UTF-8
Version 2.8.5
Date 2021-01-11
Title Event History Analysis


BugReports https://github.com/goranbrostrom/eha/issues
License GPL (>= 2)
LazyData yes
ByteCompile yes

URL http://ehar.se/r/eha/
Depends R (>= 3.5.0)
Imports stats, graphics, survival (>= 3.0)
NeedsCompilation yes
Maintainer Göran Broström <goran.brostrom@umu.se>
RoxygenNote 7.1.1
Suggests knitr, rmarkdown, bookdown
VignetteBuilder knitr, utils
Author Göran Broström [aut, cre], Jianming Jin [ctb]
Repository CRAN
Date/Publication 2021-01-12 14:30:13 UTC
### R topics documented:

<table>
<thead>
<tr>
<th>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>7</td>
</tr>
<tr>
<td>age.window</td>
<td>8</td>
</tr>
<tr>
<td>cal.window</td>
<td>9</td>
</tr>
<tr>
<td>check.dist</td>
<td>10</td>
</tr>
<tr>
<td>check.surv</td>
<td>11</td>
</tr>
<tr>
<td>child</td>
<td>12</td>
</tr>
<tr>
<td>compHaz</td>
<td>13</td>
</tr>
<tr>
<td>coxfunk</td>
<td>14</td>
</tr>
<tr>
<td>coxreg</td>
<td>15</td>
</tr>
<tr>
<td>coxreg.fit</td>
<td>18</td>
</tr>
<tr>
<td>cro</td>
<td>20</td>
</tr>
<tr>
<td>EV</td>
<td>21</td>
</tr>
<tr>
<td>fert</td>
<td>22</td>
</tr>
<tr>
<td>frail.fit</td>
<td>23</td>
</tr>
<tr>
<td>geome.fit</td>
<td>24</td>
</tr>
<tr>
<td>Gompertz</td>
<td>25</td>
</tr>
<tr>
<td>hazards</td>
<td>26</td>
</tr>
<tr>
<td>import_strata</td>
<td>26</td>
</tr>
<tr>
<td>import_Surv</td>
<td>26</td>
</tr>
<tr>
<td>infants</td>
<td>27</td>
</tr>
<tr>
<td>join.spells</td>
<td>28</td>
</tr>
<tr>
<td>Loglogistic</td>
<td>29</td>
</tr>
<tr>
<td>Lognormal</td>
<td>30</td>
</tr>
<tr>
<td>logrank</td>
<td>31</td>
</tr>
<tr>
<td>logrye</td>
<td>32</td>
</tr>
<tr>
<td>ltx</td>
<td>33</td>
</tr>
<tr>
<td>make.communal</td>
<td>34</td>
</tr>
<tr>
<td>makeham</td>
<td>35</td>
</tr>
<tr>
<td>male.mortality</td>
<td>36</td>
</tr>
<tr>
<td>mlreg</td>
<td>37</td>
</tr>
<tr>
<td>mort</td>
<td>41</td>
</tr>
<tr>
<td>oe</td>
<td>42</td>
</tr>
<tr>
<td>oldmort</td>
<td>42</td>
</tr>
<tr>
<td>Pch</td>
<td>43</td>
</tr>
<tr>
<td>pchreg</td>
<td>44</td>
</tr>
<tr>
<td>perstat</td>
<td>46</td>
</tr>
<tr>
<td>phfunc</td>
<td>47</td>
</tr>
<tr>
<td>phreg</td>
<td>49</td>
</tr>
<tr>
<td>phreg.fit</td>
<td>52</td>
</tr>
<tr>
<td>piecewise</td>
<td>53</td>
</tr>
<tr>
<td>plot.aftreg</td>
<td>54</td>
</tr>
<tr>
<td>plot.coxreg</td>
<td>55</td>
</tr>
<tr>
<td>plot.hazdata</td>
<td>56</td>
</tr>
<tr>
<td>plot.logrank</td>
<td>58</td>
</tr>
</tbody>
</table>
Description


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>

Other contributors:

- Jianming Jin [contributor]

References


See Also

Useful links:

- http://ehar.se/r/eha/
- Report bugs at https://github.com/goranbrostrom/eha/issues

---

aftreg  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.
aftreg

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?
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

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

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:

\[ \text{is changed to} \quad b - z\beta \]

\[ \text{is changed to} \quad b + z\beta \]

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).
- `n.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

<table>
<thead>
<tr>
<th>call</th>
<th>The call.</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>The method.</td>
</tr>
<tr>
<td>convergence</td>
<td>Did the optimization converge?</td>
</tr>
<tr>
<td>fail</td>
<td>Did the optimization fail? (Is NULL if not).</td>
</tr>
<tr>
<td>pfixed</td>
<td>TRUE if shape was fixed in the estimation.</td>
</tr>
<tr>
<td>param</td>
<td>The parametrization.</td>
</tr>
</tbody>
</table>

Author(s)
Göran Broström

See Also
coxreg, phreg, survreg

Examples

data(mort)
aftreg(Surv(enter, exit, event) ~ ses, param = "lifeExp", data = mort)

aftreg.fit
Parametric proportional hazards regression

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

Usage
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.
id See corresponding argument to aftreg.
control Controls convergence and output.
pfixed 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 loglihood 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

```r
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'.

Details

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

Description

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

Usage

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.
check.dist

Author(s)
  Göran Broström

See Also
  age.window, coxreg, aftreg

Examples

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)

check.dist

Graphical goodness-of-fit test

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

Usage
  check.dist(sp, pp, main = NULL, col = 1:2, lty = 1:2, 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
  lty       line types.
  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

See Also

coxreg and phreg.

Examples

data(mort)
oldpar <- par(mfrow = c(2, 2))
fit.cr <- coxreg(Surv(enter, exit, event) ~ ses, data = mort)
fit.w <- phreg(Surv(enter, exit, event) ~ ses, data = mort)
fit.g <- phreg(Surv(enter, exit, event) ~ ses, data = mort,
dist = "gompertz")
fit.ev <- phreg(Surv(enter, exit, event) ~ ses, data = mort,
dist = "ev")
check.dist(fit.cr, fit.w)
check.dist(fit.cr, fit.g)
check.dist(fit.cr, fit.ev)
par(oldpar)

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

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)
```

---

**child**

*Child mortality, Skellefteå, Sweden 1850–1900.*

**Description**

Children born in Skellefteå, Sweden, 1850-1884, are followed fifteen years or until death or out-migration.

**Usage**

```r
data(child)
```

**Format**

A data frame with 26855 children born 1850-1884.

- **id** An identification number.
- **m.id** Mother's id.
- **sex** Sex.
- **socBranch** Working branch of family (father).
- **birthdate** Birthdate.
- **enter** Start age of follow-up, always zero.
- **exit** Age of departure, either by death or emigration.
- **event** Type of departure, death = 1, right censoring = 0.
- **illeg** Born out of marriage ("illegitimate")?
- **m.age** Mother's age.

**Details**

The Skellefteå region is a large region in the northern part of Sweden.
**compHaz**

**Source**

Data originate from the Centre for Demographic and Ageing Research, Umeå University, Umeå, Sweden, [https://www.umu.se/en/centre-for-demographic-and-ageing-research/](https://www.umu.se/en/centre-for-demographic-and-ageing-research/).

**Examples**

```r
fit <- coxreg(Surv(enter, exit, event) ~ sex + socBranch, data = child, coxph = TRUE)
summary(fit)
```

---

**compHaz**  
*Graphical comparison of cumulative hazards*

**Description**

Comparison of the estimated baseline cumulative hazards functions for two survival models.

**Usage**

```r
compHaz(
  fit1,
  fit2,
  main = NULL,
  lty = 1:2,
  col = c("red", "blue"),
  printLegend = TRUE
)
```

**Arguments**

- `fit1`: An object of type "coxreg", "phreg", or other output from from survival fitters.
- `fit2`: An object of type "coxreg", "phreg", or other output from survival fitters.
- `main`: Header for the plot. Default is NULL.
- `lty`: line types.
- `col`: Line colors. should be NULL (black lines) or of length 2.
- `printLegend`: Should a legend be printed? Default is TRUE.

**Value**

No return value.

**Author(s)**

Göran Broström

**See Also**

`hazards`, `coxreg`, and `phreg`.
Examples

```r
fit.cr <- coxreg(Surv(enter, exit, event) ~ sex, data = oldmort)
fit.w <- phreg(Surv(enter, exit, event) ~ sex, data = oldmort)
compHaz(fit.cr, fit.w)
```

Description

Calculates minus the log likelihood function and its first and second order derivatives for data from a Cox regression model. It is used by `coxreg` if the argument `coxph = FALSE`.

Usage

```r
coxfunk(beta, X, offset, rs, what = 2)
```

Arguments

- `beta`: Regression parameters
- `X`: The design (covariate) matrix.
- `offset`: Offset.
- `rs`: Risk set created by `risksets(..., collate_sets = TRUE)`
- `what`: what = 0 means only loglihood, 1 means score vector as well, 2 loglihood, score and hessian.

Details

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

Value

A list with components

- `loglik`: The log likelihood.
- `dloglik`: The score vector. Nonzero if what >= 1
- `d2loglik`: The hessian. Nonzero if ord >= 2

Author(s)

Göran Broström

See Also

`coxreg`
**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", "mppl", "ml"), control = list(eps = 1e-08, maxiter = 25, trace = FALSE), singular.ok = TRUE, model = FALSE, center = NULL, x = FALSE, y = TRUE, hazards = NULL, boot = FALSE, efrac = 0, geometric = FALSE, rs = NULL, frailty = NULL, max.survs = NULL, coxph = 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.
- `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 `getOption("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`: deprecated. See Details.
- `x`: Return the design matrix in the model object?
- `y`: return the response in the model object?
- `hazards`: deprecated. Was: Calculate baseline hazards? Default is TRUE. Calculating hazards is better done separately, after fitting. In most cases.
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.
coxph Logical, defaults to TRUE. Determines if standard work should be passed to coxph via entry points.

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.

Value

A list of class c("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.
hazards The estimated baseline hazards, calculated at the value zero 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 risk times, 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).
n.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.
The covariates.

Total Time at Risk.

List of levels of factors.

The calling formula.

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

The estimated standard errors of the bootstrap replicates.

The call.

The method.

Number of strata.

Did the optimization converge?

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

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

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.

Göran Broström


See Also

coxph, risksets

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))
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 = "efron",  
          boot = FALSE,  
          efrac = 0,  
          calc.martres = TRUE,  
          control,  
          verbose = TRUE,  
          calc.hazards = NULL,  
          center = NULL)

Arguments

 X The design matrix.
 Y The survival object.
 rs The risk set composition. If absent, calculated.
 weights Case weights; time-fixed or time-varying.
 t.offset Case offset; time-varying.
 strats The stratum variable. Can be absent.
 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".

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

efrac
  Upper limit of fraction failures in 'mppl'.

calc.martres
  Should martingale residuals be calculated?

control
  See coxreg

verbose
  Should Warnings about convergence be printed?

calc.hazards
  Deprecated. See coxreg.

center
  Deprecated. See coxreg.

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

coxreg, 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

```r
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.
The EV Distribution

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 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

$$h(x) = (b/\sigma)(x/\sigma)^{(b - 1)} \exp((x/\sigma)^b)$$

for $x \geq 0$.
fert

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 (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 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 next.ivl ends in a new birth (event = 1) or is right censored (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

https://www.umu.se/enheten-for-demografi-och-aldrandeforskning/

Examples

data(fert)
fit <- coxreg(Surv(next.ivl, event) ~ ses + prev.ivl, data = fert, subset =
(parity == 1))
summary(fit)

Description

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

Usage

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`
Gompertz

The Gompertz Distribution

Description

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

Usage

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

Arguments

shape, scale  shape and scale parameters, both defaulting to 1.
rate the rate parameter for that parametrization, replaces scale.
lower.tail logical; if TRUE (default), probabilities are \( P(X \leq x) \), otherwise, \( P(X > x) \).
param default or canonical or rate.
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 \( a \to a/b \), so that \( b \) is a true scale parameter (for any fixed \( a \)), and \( b \) is an 'AFT parameter'. If param = "rate", then \( b \to 1/b \).
import \texttt{Surv} function imported from \texttt{survival}

\textbf{Description}

This function is imported from the \texttt{survival} package. See \texttt{Surv}.

\begin{verbatim}
import_Surv

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 \texttt{NaN}, with a warning.

\begin{verbatim}
hazards
\end{verbatim}

\textbf{Description}

Get baseline hazards atoms from fits from

\textbf{Usage}

\texttt{hazards(x, cum = TRUE, ...)}

\textbf{Arguments}

\begin{itemize}
\item \texttt{x} \quad A reg object.
\item \texttt{cum} \quad Logical: Should the cumulative hazards be returned?
\item \texttt{...} \quad Additional arguments for various methods.
\end{itemize}

\textbf{Value}

A list where each component is a two-column matrix representing hazard atoms from one stratum. The first column is event time, and the second column is the corresponding hazard atom.

\begin{verbatim}
import_strata
\end{verbatim}

\textbf{Description}

This function is imported from the \texttt{survival} package. See \texttt{strata}.

\begin{verbatim}
import_Surv
\end{verbatim}

\textbf{Description}

This function is imported from the \texttt{survival} package. See \texttt{Surv}.
infants

Infant mortality and maternal death, Sweden 1821–1894.

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.
ses  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, https://www.
umu.se/enheten-for-demografi-och-aldrandeforskning/.

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)
summary(fit.w) ## Weibull proportional hazards model.

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

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

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.
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

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

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

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

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.
logrank

The Log-rank test

Description
Performs the log-rank test on survival data, possibly stratified.

Usage
logrank(Y, group, data = parent.frame())

Arguments
Y               a survival object as returned by the Surv function.
group           defines the groups to be compared. Coerced to a factor.
data            a data.frame in which to interpret the variables.

Value
A list of class logrank with components
test.statistic  The logrank (score) test statistic.
df              The degrees of freedom of the test statistic.
p.value         The p value of the test.
hazards         A list of two-column matrices, describing event times and corresponding hazard atoms in each stratum (class 'hazdata').
call            The call

Note
The test is performed by fitting a Cox regression model and reporting its score test. With tied data, this might be slightly different from the true logrank test, but the difference is unimportant in practice.

Author(s)
Göran Broström

See Also
coxreg, print.logrank.

Examples
fit <- logrank(Y = Surv(enter, exit, event), group = civ, data = oldmort)
fit
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 `coxreg`, `phreg`, `tpchreg`, or `aftreg`, similar to what `xtable` does for fits from other functions.

Usage

```r
ltx(
  x, 
  caption = NULL, 
  label = NULL, 
  dr = NULL, 
  digits = max(options()$digits - 4, 3),
  ...
)
```

Arguments

- **x**: The output from a call to `coxreg`, `tpchreg`, or `aftreg`
- **caption**: A suitable caption for the table.
- **label**: A label used in the LaTeX code.
- **dr**: Output from a `drop1` call.
- **digits**: Number of digits to be printed.
- **...**: Not used.

Details

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

Value

LaTeX code version of the results from a run with `coxreg`, `phreg`, `phreg`, or `aftreg`.

Note

There is no method in `xtable` for `coxreg` and friends.

Author(s)

Göran Broström.

See Also

- `xtable`, `coxreg`, `phreg`, `phreg`, and `aftreg`. 
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 calendar 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

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.
male.mortality

Usage

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 surving 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)
mlreg

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.
eses  Socio-economic status, a factor with levels lower, upper

Details

The interesting explanatory covariate is ses (socioeconomic status), which is a time-varying co-variate. 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 mort.

Source

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

References

https://www.umu.se/enheten-for-demografi-och-aldraneforskning/

Examples

data(male.mortality)
fit <- coxreg(Surv(enter, exit, event) ~ ses, data = male.mortality)
summary(fit)

mlreg

ML proportional hazards regression

Description

Maximum Likelihood estimation of proportional hazards models. Is deprecated, use coxreg instead.
Usage

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
)

Arguments

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 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.
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 cameleon 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.
- 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 Walt 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.
- bootstrap The bootstrap sample, if requested on input.
- sigma Present if a frailty model is fitted. Equals the estimated frailty standard deviation.
sigma.sd          The standard error of the estimated frailty standard deviation.
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 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, 1, 1, 1, 0),
                  x = c(0, 2, 1, 1, 1, 0, 1),
                  sex = c(0, 0, 0, 1, 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
```
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(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, Umeå University, Umeå, Sweden.

References
https://www.umu.se/enheten-for-demografi-och-aldrandeforskning/

Examples
data(mort)
fit <- coxreg(Surv(enter, exit, event) ~ ses, data = mort)
summary(fit)
Create an oe object

Description

Create an oe ("occurrence/exposure") object, used as a response variable in a model formula specifically in `tpchreg`.

Usage

```r
oe(count, exposure)
```

Arguments

- `count` Number of events, a non-negative integer-valued vector.
- `exposure` exposure time corresponding to count. A positive numeric vector.

See Also

- `tpchreg`.

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

Details
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 \( \text{length}(n) > 1 \), 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.

---

**pchreg**

**Piecewise Constant Proportional Hazards Regression**

Description

Proportional hazards model with piecewise constant baseline hazard(s). Allows for stratification and left truncated and right censored data.

Usage

```r
pchreg(
  formula = formula(data),
  data = parent.frame(),
  na.action = getOption("na.action"),
  cuts = NULL,
  init,
  control = list(eps = 1e-08, maxiter = 20, trace = FALSE),
)```
pchreg

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.
cuts 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.
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?

Value

A list of class "pchreg" with components

coefficients Fitted parameter estimates.
cuts Cut points (NULL if no cut points).
hazards The estimated constant levels.
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. 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).
perstat

n.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).

Author(s)

Göran Broström

See Also

phreg, coxreg, link{aftreg}.

Examples

```r
fit <- pchreg(Surv(enter, exit, event) ~ ses.50 + sex, data = oldmort, 
cuts = seq(60, 100, by = 5))
summary(fit)
fit.cr <- coxreg(Surv(enter, exit, event) ~ ses.50 + sex, data = oldmort)
compHaz(fit.cr, fit)
```

Description

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

Usage

```r
perstat(surv, period, age = c(0, 200))
```
**phfunc**

**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

**Author(s)**

Göran Broström

**See Also**

piecewise

---

**phfunc**  
*Log likelihood 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**

```r
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 considered in the partial derivatives.
- **dist**: Which distribution? 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 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

- **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**

```r
phreg(
  formula = formula(data),
  data = parent.frame(),
  na.action = getOption("na.action"),
  dist = "weibull",
  cuts = NULL,
  init,
  shape = 0,
  param = c("canonical", "rate"),
  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 "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.
- **cuts**: Only used with dist = "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. Only relevant for the Weibull and Extreme Value distributions.
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?

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

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

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

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 coponent 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. 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).

n.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)

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.

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.
piecewise

Author(s)
Göran Broström

See Also
phreg

---

piecewise

Piecewise hazards

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

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)
```

plot.coxreg

Plot method for `coxreg` objects

Description

A plot of a baseline function of a `coxreg` fit is produced, one curve for each stratum. A wrapper for `plot.survfit` in `survival`.

Usage

```r
## S3 method for class 'coxreg'
plot(
  x,
  fn = c("cum", "surv", "log", "loglog"),
  conf.int = FALSE,
  fig = TRUE,
  xlim = NULL,
  ylim = NULL,
  main = NULL,
  xlab = "Duration",
  ylab = "",
  col = 1,
  lty = 1,
  printLegend = TRUE,
  ...
)
```
Arguments

\texttt{x} \quad \text{A coxreg object}

\texttt{fn} \quad \text{What should be plotted? Default is "cumhaz", and the other choices are "surv", "log", and "loglog".}

\texttt{conf.int} \quad \text{logical or a value like 0.95 (default for one curve).}

\texttt{fig} \quad \text{logical. If TRUE the plot is actually drawn, otherwise only the coordinates of the curve(s) are returned.}

\texttt{xlim} \quad \text{Start and end of the x axis.}

\texttt{ylim} \quad \text{Start and end of the y axis.}

\texttt{main} \quad \text{A headline for the plot}

\texttt{xlab} \quad \text{Label on the x axis.}

\texttt{ylab} \quad \text{Label on the y axis.}

\texttt{col} \quad \text{Color of the curves. Defaults to 'black'.}

\texttt{lty} \quad \text{Line type(s).}

\texttt{printLegend} \quad \text{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 \texttt{legend} for all possible choices.}

\ldots \quad \text{Other parameters to pass to the plot.}

Value

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

\texttt{plot.hazdata} \quad \textit{Plots of hazdata objects.}

Description

Baseline hazards estimates.

Usage

\texttt{## S3 method for class 'hazdata'
plot(
  x,
  strata = NULL,
  fn = c("cum", "surv", "log", "loglog"),
  fig = TRUE,
  xlim = NULL,
  ylim = NULL,
  main = NULL,
  xlab = "",
  ylab = "",
  ...}

\begin{verbatim}
col = "black",
lty = 1,
printLegend = TRUE,
...)
\end{verbatim}

**Arguments**

- **x**: A hazdata object, typically the 'hazards' element in the output from `link{coxreg}` with `method = "ml"` or `method = "mppl"` or `coxph = FALSE`.
- **strata**: Stratum names if there are strata present.
- **fn**: Which type of plot? Allowed values are "cum" (or "cumhaz"), "surv" (or "sur"), "log", or "loglog". The last two plots the cumulative hazards on a log (y) scale or a log-log (xy) scale, respectively.
- **fig**: Should a plot actually be produced? Default is TRUE.
- **xlim**: Horizontal plot limits. If NULL, calculated by the function.
- **ylim**: Vertical plot limits. If NULL, set to \(c(0,1)\) for a plot of the survival function.
- **main**: A heading for the plot.
- **xlab**: Label on the x axis.
- **ylab**: Label on the y-axis.
- **col**: Color of the lines. May be a vector of length equal to No. of strata.
- **lty**: Line type(s). May be a vector of length equal to No. of strata.
- **printLegend**: Logical or character; should a legend be produced? Defaults to TRUE. If character, it should be one of `bottomleft`, `bottomright`, etc, see `legend`.
- **...**: Anything that `plot.default` likes...

**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).

**Note**

- \(x\) is a list where each element is a two-column matrix. The first column contains failure times, and the second column contains the corresponding 'hazard atoms'.

**Author(s)**

Göran Broström
Examples

time0 <- numeric(50)
group <- c(rep(0, 25), rep(1, 25))
x <- runif(50, -0.5, 0.5)
time1 <- rexp(50, exp(group))
event <- rep(1, 50)
fit <- coxreg(Surv(time0, time1, event) ~ x + strata(group), method = "ml")
plot(fit$hazards, col = 1:2, fn = "surv", xlab = "Duration")
## Same result as:
## plot(fit, col = 1:2, fn = "sur", xlab = "Duration")

plot.logrank

Plots of hazdata objects.

Description

Baseline hazards estimates.

Usage

## S3 method for class 'logrank'
plot(x,
     fn = c("cum", "surv", "log", "loglog"),
     xlim = NULL,
     ylim = NULL,
     main = NULL,
     xlab = "",
     ylab = "",
     col = "black",
     lty = 1,
     printLegend = TRUE,
     ...
)

Arguments

x A logrank object, typically the 'hazards' element in the output from link{logrank}.
fn Which type of plot? Allowed values are "cum" (or "cumhaz"), "surv" (or "sur"),
"log", or "loglog". The last two plots the cumulative hazards on a log (y) scale
or a log-log (xy) scale, respectively.
xlim Horizontal plot limits. If NULL, calculated by the function.
ylim Vertical plot limits. If NULL, set to c(0, 1) for a plot of the survival function.
main A heading for the plot.
**plot.phreg**

- `xlab` Label on the x axis.
- `ylab` Label on the y-axis.
- `col` Color of the lines. May be a vector of length equal to No. of strata.
- `lty` Line type(s). May be a vector of length equal to No. of strata.
- `printLegend` Logical or character; should a legend be produced? Defaults to TRUE. If character, it should be one of `bottomleft`, `bottomright`, etc, see `legend`.
- `...` Anything that `plot.default` likes...

**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).

**Note**

`x` is a list where each element is a two-column matrix. The first column contains failure times, and the second column contains the corresponding 'hazard atoms'.

**Author(s)**

Göran Broström

**Examples**

```r
fit <- logrank(Surv(enter, exit, event), group = civ, data = oldmort)
plot(fit)
```

---

**plot.phreg**  
*Plots output from a phreg regression*

**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,
  score = 1,
  fig = TRUE,
  ...
)
```

Arguments

- `x` A `phreg` object
- `fn` Which function should be plotted? Default is the hazard function(s).
- `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.
- `score` Multiplication factor for the hazard function.
- `fig` logical, should the graph be drawn? If FALSE, data is returned.
- `...` Extra parameters passed to `plot` and `lines`.

Value

No return value if `fig = TRUE`, otherwise the cumulative hazards function (coordinates), given `fn = "cum"`.

Note

Reference hazard is given by the fit; zero for all covariates, and the reference category for factors.
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)
```

Description

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

Usage

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

Arguments

- `x`  A `tpchreg` object
- `fn`  Which functions should be plotted? Default is the hazard function.
- `log`  character, "" (default), "y", or "xy".
- `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.
... Extra parameters passed to 'plot' and 'lines'.

Value
No return value.

Author(s)
Göran Broström

See Also
tpchreg

---

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
```r
## 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, e.g., `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`

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)
```
plotHaz

Graphical comparing of cumulative hazards

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

Usage
plotHaz(
  sp,
  pp,
  interval,
  main = NULL,
  xlab = "Time",
  ylab = "Cum. hazards",
  col = c("blue", "red"),
  lty = 1:2,
  ylim,
  printLegend = TRUE
)

Arguments

sp  An object of type "coxreg" or "phreg", typically output from coxreg or phreg.
pp  An object of type "coxreg" or "phreg", typically output from coxreg or phreg.
interval  Time interval for the plot, if missing, calculated from sp.
main  Header for the plot. Default is distribution and "cumulative hazard function"
xlab  Label on x axis (default "Time")
ylab  Label on y axis (default "Cum. Hazards")
col  Line colors. should be NULL (black lines) or of length 2
lty  line types.
ylim  Y limits for the plot.
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
See Also

check.dist, coxreg and phreg.

Examples

data(mort)
op <- par(mfrow = c(1, 2))
fit.cr <- coxreg(Surv(enter, exit, event) ~ ses, data = mort)
fit.w <- phreg(Surv(enter, exit, event) ~ ses, data = mort)
fit.g <- phreg(Surv(enter, exit, event) ~ ses, data = mort,
dist = "gompertz")
plotHaz(fit.cr, fit.w, interval = c(0, 20), main = "Weibull")
plotHaz(fit.cr, fit.g, main = "Gompertz")
par(op)

Description

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

Usage

## 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
Description

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

Usage

## 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.

Value

No value is returned.

Author(s)

Göran Broström

See Also

coxreg, print.coxph
print.logrank

Prints logrank objects

Description

The result of logrank is printed.

Usage

## S3 method for class 'logrank'
print(x, digits = max(options()$digits - 4, 6), ...)

Arguments

- **x**: A logrank object
- **digits**: Precision in printing
- **...**: Not used.

Value

The input is returned invisibly.

Author(s)

Göran Broström

See Also

logrank, coxreg

print.phreg

Prints phreg objects

Description

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

Usage

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


print.risksets

Arguments

x A phreg object
digits Precision in printing
...

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'.
...

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.summary.aftreg

S3 method for class 'summary.aftreg'
print(x, digits = max(getOption("digits") - 3, 3), short = FALSE, ...)

Arguments

x A summary.aftreg object, typically the result of running summary.aftreg,
summary on a phreg object.
digits Output format.
short Logical: If TRUE, short output, only regression.
... Other arguments.

Value
No value is returned.

Author(s)
Göran Broström

See Also
aftreg, summary.aftreg
print.summary.coxreg  Prints summary.coxreg objects

Description
Prints summary.coxreg objects

Usage
## S3 method for class 'summary.coxreg'
print(x, digits = 3, short = FALSE, ...)

Arguments
x                  A summary.coxreg object, typically the result of running summary.coxreg, summary on a coxreg object.
digits             Output format.
short              Logical, short or long (default) output?
...                Other arguments.

Value
No value is returned.

Author(s)
Göran Broström

See Also
coxreg, summary.coxreg

print.summary.phreg  Prints summary.phreg objects

Description
Prints summary.phreg objects

Usage
## S3 method for class 'summary.phreg'
print(x, digits = max(getOption("digits") - 3, 3), ...)

Arguments
x                  A summary.phreg object, typically the result of running summary.phreg, summary on a phreg object.
Arguments

x A summary.phreg object, typically the result of running summary.phreg, summary on a phreg object.
digits Output format.
... Other arguments.

Value

No value is returned.

Author(s)

Göran Broström

See Also

phreg, summary.phreg

print.summary.tpchreg  Prints summary.tpchreg objects

Description

Prints summary.tpchreg objects

Usage

## S3 method for class 'summary.tpchreg'
print(x, digits = max(getOption("digits") - 3, 3), ...)

Arguments

x A summary.tpchreg object, typically the result of running summary.tpchreg, summary on a tpchreg object.
digits Output format.
... Other arguments.

Value

No value is returned.

Author(s)

Göran Broström

See Also

tpchreg, summary.tpchreg
print.tpchreg  \hspace{1cm} \textit{Prints tpchreg objects}

\section*{Description}

More "pretty-printing"

\section*{Usage}

\begin{verbatim}
## S3 method for class 'tpchreg'
print(x, digits = max(options()$digits - 4, 3), ...)
\end{verbatim}

\section*{Arguments}

\begin{description}
\item[x] A \texttt{tpchreg} object, typically the result of running \texttt{tpchreg}
\item[digits] Output format.
\item[...] Other arguments.
\end{description}

\section*{Details}

Doesn't work with three-way or higher interactions.

\section*{Value}

No value is returned.

\section*{Author(s)}

Göran Broström

\section*{See Also}

\begin{itemize}
\item \texttt{tpchreg}, \texttt{print.coxreg}
\end{itemize}

\section*{print.weibreg  \hspace{1cm} \textit{Prints weibreg objects}}

\section*{Description}

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

\section*{Usage}

\begin{verbatim}
## S3 method for class 'weibreg'
print(x, digits = max(options()$digits - 4, 3), ...)
\end{verbatim}
**Arguments**

- `x` A `weibreg` 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**

`weibreg`, `print.coxph`

---

**Description**

Retrieves regression tables

**Usage**

```r
gretable(x, digits = 3, short = TRUE, ...)
```

**Arguments**

- `x` A `summary.XXreg` object, typically the result of running `summary.XXreg`, summary on a `XXreg` object.
- `digits` Output format.
- `short` If `TRUE`, return only coefficients table.
- `...` Other arguments.

**Value**

A character data frame, ready to print in various formats.

**Note**

Should not be used if interactions present.
risksets

Finds the compositions and sizes of risk sets

Description

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

Usage

risksets(
  x,
  strata = NULL,
  max.survs = NULL,
  members = TRUE,
  collate_sets = FALSE
)

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.
collate_sets logical. If TRUE, group information by risk sets in a list. Only if members = TRUE.

Details

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

Value

A list with components (if collate_sets = FALSE)
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  If 'members' is TRUE, 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, 2, 3, 4)
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.

References
https://www.ed.lu.se/databases

Examples
```r
data(scania)
summary(scania)
```

Description
Prints aftreg objects

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

Arguments
```r
object A aftreg object
...
Additional ...
```
Summary.coxreg

Author(s)
Göran Broström

See Also
print.coxreg

Examples

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

summary.coxreg  A summary of coxreg objects.

Description
A summary of coxreg objects.

Usage

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

Arguments

object  A coxreg object
...
Additional ...

Author(s)
Göran Broström

See Also
print.coxreg

Examples

fit <- coxreg(Surv(enter, exit, event) ~ sex + civ, data = oldmort)
summary(fit)
### summary.phreg

*A summary of phreg objects.*

**Description**

A summary of phreg objects.

**Usage**

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

**Arguments**

- `object` A phreg object
- `...` Additional ...

**Author(s)**

Göran Broström

**See Also**

`print.phreg`

**Examples**

```r
fit <- phreg(Surv(enter, exit, event) ~ sex + civ, data = oldmort)
summary(fit)
```

### summary.tpchreg

*Summary for tpchreg objects*

**Description**

Summary for tpchreg objects

**Usage**

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

object  A `tpchreg` object.
...
Additional ...

Author(s)

Göran Broström

See Also

`tpchreg`

Examples

```
## The function is currently defined as
## function (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

```r
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])
}
```

---

**swedeaths**


**Description**

A data frame containing data on the number of deaths by sex, age and year, Sweden 1968-2019.

**Usage**

```r
data(swedeaths)
```

**Format**

A data frame with 5 variables and 10504 observations.

- **age** Numerical with integer values 0-100, representing achieved age in years during the actual calendar year. The highest value, 100, represents ages 100 and above.
sex A factor with two levels, "women" and "men".
year Calendar year.
deaths Number of deaths by age, sex, and year.
id Created by the `reshape` procedure, see Details.

Details

Data are downloaded from Statistics Sweden in the form of a csv file and in that process converted to a data frame. Variable names are translated from Swedish, and some of them are converted to factors. Each numeric column contains the average population by sex and age, calculated by taking the mean value of the population size at December 31 the previous year and December 31 the current year. The original data contain the sizes at the end of each year. The original data set is in wide form and then converted to long format.

Source

Statistics Sweden, [https://scb.se](https://scb.se).

See Also

`swepop`, `tpchreg`

Examples

```r
summary(swedeaths)
## maybe str(swedeaths) ...
```

---

**swepop**  

**Description**

A data frame containing data on the population size by sex, age and year, Sweden 1968-2019.

**Usage**

data(swepop)

**Format**

A data frame with 5 variables and 10504 observations.

- **age** Numerical with integer values 0-100, representing achieved age in years during the actual calendar year. The highest value, 100, represents ages 100 and above.
- **sex** A factor with two levels, "women" and "men".
- **year** Calendar year.
- **pop** Average population by age, sex, and year.
- **id** Created by the `reshape` procedure, see Details.
Details

Data are downloaded from Statistics Sweden in the form of a csv file and converted to a data frame. Variable names are translated from Swedish, and some of them are converted to factors. The variable \texttt{pop} contains the average population by sex and age, calculated by taking the mean value of the population size at December 31 the previous year and December 31 the current year. The original data contain the sizes at the end of each year. The original data set is in wide format and converted to long format by \texttt{reshape}.

Source

Statistics Sweden, \url{https://scb.se}.

See Also

\texttt{swedeaths}

Examples

\begin{verbatim}
summary(swepop)
## maybe str(swepop) ...
\end{verbatim}

\begin{verbatim}

table.events(enter = rep(0, length(exit)), exit, event, strict = TRUE)
\end{verbatim}

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

\begin{verbatim}
table.events(enter = rep(0, length(exit)), exit, event, strict = TRUE)
\end{verbatim}

Arguments

\begin{verbatim}
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.
\end{verbatim}

Value

A list with components

\begin{verbatim}
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.
\end{verbatim}
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)
```

toBinary

Transforms a "survival" data frame into a data frame suitable for binary (logistic) regression

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
```r
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](https://cran.r-project.org/package=coxreg), [glm](https://cran.r-project.org/package=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 `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

toDate

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)

Arguments

dates A vector of dates in character form or of class Date

Value

A vector of durations, as described above.
toTpch

Author(s)

Göran Broström

See Also
toDate

Examples

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

toTime(c("1897-05-16", "1901-11-21"))
```

---

**toTpch**

*Transform survival data to tabular form*

Description

Transform a "survival data frame" to tabular form aggregating number of events and exposure time by time intervals and covariates.

Usage

toTpch(formula, data, cuts, enter = "enter", exit = "exit",
       event = "event", episode = "age")

Arguments

- **formula**: A model formula.
- **data**: A data frame with survival data.
- **cuts**: An ordered, non-negative vector of time points at which a hazard function changes value. Note that data are left truncated at cuts[1] (the smallest value) and right censored at c[n], where n is the length of cuts and cuts[n] == max(cuts).
- **enter**: Character string with the name of the variable representing left truncation values.
- **exit**: Character string with the name of the event/censoring time variable.
- **event**: Character string with the name of the event indicator variable.
- **episode**: Character string with the name of the output variable of the grouped time (a factor variable).
Details

If cuts is missing, nothing is done. Internally, this function first calls survival::survSplit and then stats::aggregate.

Value

A data frame with exposure time and number of events aggregated by time intervals and covariates. If all covariates are factors, this usually results in a huge reduction of the size of the data frame, but otherwise the size of the output may be larger than the size of the input data frame.

Note

Episodes, or parts of episodes, outside min(cuts), max(cuts) are cut off. With continuous covariates, consider rounding them so that the number of distinct observed values is not too large.

Author(s)

Göran Broström

---

tpchreg

Proportional hazards regression with piecewise constant hazards and tabular data.

Description

Proportional hazards regression with piecewise constant hazards and tabular data.

Usage

```r
tpchreg(formula, data, time, weights, last, subset, na.action, contrasts = NULL, start.coef = NULL, control = list(epsilon = 1.e-8, maxit = 200, trace = FALSE))
```

Arguments

- **formula**: a formula with `oe(count, exposure) ~ x1 + ...`
- **data**: a data frame with occurrence/exposure data plus covariates.
- **time**: the time variable, a factor character vector indicating time intervals, or numeric, indicating the start of intervals.
- **weights**: Case weights.
- **last**: If time is numeric, the closing of the last interval.
- **subset**: subset of data, not implemented yet.
- **na.action**: Not implemented yet.
- **contrasts**: Not implemented yet.
- **start.coef**: For the moment equal to zero, not used.
- **control**: list of control parameters for the optimization.
The interpretation of cuts is different from that in \texttt{hpch}. This is intentional.

See Also

\texttt{oe}.

Examples

```r
sw <- swepop
sw$deaths <- swdeaths$deaths
fit <- tpchreg(oe(deaths, pop) ~ strata(sex) + I(year - 1990), time = age, last = 101, data = sw)
summary(fit)
```

---

**weibreg**

Weibull Regression

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

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>formula</td>
<td>a formula object, with the response on the left of a \texttt{~} operator, and the terms on the right. The response must be a survival object as returned by the \texttt{Surv} function.</td>
</tr>
<tr>
<td>data</td>
<td>a \texttt{data.frame} in which to interpret the variables named in the formula.</td>
</tr>
</tbody>
</table>
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.

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

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

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("weibreg","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 componet 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
```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))
weibreg(Surv(time, status) ~ x + strata(sex), data = dat) # stratified model
```
weibreg.fit  

Weibull regression

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

Usage
weibreg.fit(X, Y, strata, offset, init, shape, control, center = TRUE)

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

Details
See weibreg for more detail.

Value
coefficients  Estimated regression coefficients plus estimated scale and shape coefficients, sorted by strata, if present.
var
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.
**Weibull**

**Author(s)**
Göran Broström

**See Also**
weibreg

---

The (Cumulative) Hazard Function of a Weibull Distribution

**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

```
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)
```

---

**wfunk**

*Loglihood function of a Weibull regression*

Description

Calculates minus the log likelihood function and its first and second order derivatives for data from a Weibull regression model. Is called by `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 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.
Details

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

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

This is in correspondence with \texttt{dweibull}.

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

- \texttt{weibreg}
Index

* cluster
toBinary, 84
* datasets
child, 12
fert, 22
infants, 27
logrye, 32
male.mortality, 36
mort, 41
oldmort, 42
scania, 75
swedeaths, 81
swepop, 82
* distribution
check.dist, 10
compHaz, 13
coxfunk, 14
EV, 21
Gompertz, 25
Loglogistic, 29
Lognormal, 30
makeham, 35
Pch, 43
phfunc, 47
plotHaz, 64
wfunk, 94
* dplot
plot.aftreg, 54
plot.phreg, 59
plot.tpchreg, 61
plot.weibreg, 62
* manip
check.surv, 11
cro, 20
join.spells, 28
SurvSplit, 80
toTpch, 87
* nonparametric
perstat, 46

piecewise, 53
* printing
ltx, 33
* print
summary.aftreg, 76
summary.coxreg, 77
summary.weibreg, 79
* regression
aftreg, 4
aftreg.fit, 7
coxreg, 15
coxreg.fit, 18
mlreg, 37
pchreg, 44
phreg, 49
phreg.fit, 52
print.aftreg, 65
print.logrank, 67
print.phreg, 67
print.weibreg, 72
tpchreg, 88
weibreg, 89
weibreg.fit, 92
* summary
print.risksets, 68
summary.tpchreg, 78
* survival
aftreg, 4
aftreg.fit, 7
age.window, 8
cal.window, 9
check.surv, 11
coxfunk, 14
coxreg, 15
coxreg.fit, 18
geome.fit, 24
join.spells, 28
logrank, 31
make.communal, 34
INDEX

mcreg, 37
pcreg, 44
perstat, 46
phfunc, 47
phreg, 49
phreg.fit, 52
piecewise, 53
plot.aftreg, 54
plot.hazdata, 56
plot.logrank, 58
plot.phreg, 59
plot.tpchreg, 61
plot.weibreg, 62
print.aftreg, 65
print.coxreg, 66
print.logrank, 67
print.phreg, 67
print.summary.aftreg, 69
print.summary.coxreg, 70
print.summary.phreg, 71
print.tpchreg, 72
print.weibreg, 72
regtable, 73
risksets, 74
summary.aftreg, 76
summary.coxreg, 77
summary.phreg, 78
summary.tpchreg, 78
summary.weibreg, 79
table.events, 83
tobinary, 84
toDate, 85
toTime, 86
tpchreg, 88
weibreg, 89
weibreg.fit, 92
Weibull, 93
wfunk, 94
* table
  tpchreg, 88

aftreg, 4, 7–10, 12, 29, 33, 35, 55, 69
aftreg.fit, 7
age.window, 8, 10, 80

cal.window, 9, 9, 35
check.dist, 10, 51, 65
check.surv, 11, 29

cchild, 12
compHaz, 13
coxph, 4, 16, 17, 35, 50, 90
coxreg, 4, 7, 9–14, 15, 18, 19, 24, 29, 31, 33,
  35, 39, 40, 46, 50, 51, 64–67, 70, 74,
  75, 85, 90, 91
coxreg.fit, 18
cro, 20
dEV (EV), 21
dgmpertz (Gompertz), 25
dlllogis (Loglogistic), 29
dlnormal (Lognormal), 30
dmakeham (makeham), 35
dpch (Pch), 43
dweibull, 93, 95
eha (eha-package), 3
eha-package, 3
EV, 21
fert, 22
frail, 23
gfunk, 24
geome.fit, 24
glm, 85
Gompertz, 25, 50
gompertz (Gompertz), 25
hazards, 13, 26
HEV (EV), 21
hEV (EV), 21
Hgmpertz (Gompertz), 25
hgmpertz (Gompertz), 25
Hlllogis (Loglogistic), 29
hlllogis (Loglogistic), 29
Hlnormal (Lognormal), 30
hlnormal (Lognormal), 30
Hmakeham (makeham), 35
hmakeham (makeham), 35
Hpch (Pch), 43
hpch (Pch), 43
Hweibull (Weibull), 93
hweibull (Weibull), 93

import_strata, 26
import_Surv, 26
infants, 27
INDEX

join.spells, 12, 28
legend, 56, 57, 59
llogis (Loglogistic), 29
lnorm (Lognormal), 30
Loglogistic, 29
Lognormal, 30
logrank, 31, 67
logrye, 32
ltx, 33
make.comunal, 34
makeham, 35
male.mortality, 36
match, 21
mlreg, 37
mort, 41
mpch (Pch), 43
oe, 42, 89
oldmort, 42
paste, 21
Pch, 43
pch (Pch), 43
phreg, 44
perstat, 46, 53
pEV (EV), 21
pgompertz (Gompertz), 25
phfunc, 47
phreg, 7, 10, 11, 13, 33, 46, 48, 49, 52, 53, 61, 
63–66, 68, 71, 80, 91
phreg.fit, 52
piecewise, 47, 53
pllogis (Loglogistic), 29
plnorm (Lognormal), 30
plot.aftreg, 54
plot.coxreg, 55
plot.default, 57, 59
plot.hazdata, 56
plot.logrank, 58
plot.phreg, 59
plot.tpchreg, 61
plot.weibreg, 62
plotHaz, 64
pmakeham (makeham), 35
ppch (Pch), 43
print.aftreg, 65
print.coxph, 65, 66, 68, 73
print.coxreg, 66, 72, 77
print.logrank, 31, 67
print.phreg, 67, 78
print.risksets, 68
print.summary.aftreg, 69
print.summary.coxreg, 70
print.summary.phreg, 70
print.summary.tpchreg, 71
print.tpchreg, 72
print.weibreg, 72, 79, 91
pweibull, 93
qEV (EV), 21
qgompertz (Gompertz), 25
qllogis (Loglogistic), 29
qlnorm (Lognormal), 30
qmakeham (makeham), 35
qpch (Pch), 43
regtable, 73
rEV (EV), 21
rgompertz (Gompertz), 25
risksets, 17, 19, 23, 24, 40, 74, 84
rllogis (Loglogistic), 29
rlnorm (Lognormal), 30
rmakeham (makeham), 35
rpch (Pch), 43
scania, 75
strata, 26
strata (import_strata), 26
summary.aftreg, 69, 76
summary.coxreg, 70, 74, 77
summary.phreg, 71, 78
summary.tpchreg, 71, 78
summary.weibreg, 79
Surv, 26, 31
Surv (import_Surv), 26
survival, 55
survreg, 5–7, 50, 90
SurvSplit, 80
survSplit, 80
swedeaths, 81, 83
sweepop, 82, 82

table.events, 75, 83
toBinary, 84
toDate, 85, 87
toTime, 86, 86
INDEX

toTpch, 87
tpchreg, 33, 42, 62, 71, 72, 79, 82, 88
weibreg, 63, 73, 89, 92–95
weibreg.fit, 92
Weibull, 90, 93
wfunk, 94