Package ‘eha’

September 6, 2023

Encoding UTF-8
Version 2.11.0
Date 2023-09-06
Title Event History Analysis

BugReports https://github.com/goranbrostrom/eha/issues
License GPL (>= 2)
LazyData yes
ByteCompile yes
URL https://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 7.2.3
Suggests knitr, rmarkdown, bookdown
VignetteBuilder knitr, utils
Author Göran Broström [aut, cre], Jianming Jin [ctb]
Repository CRAN
Date/Publication 2023-09-06 16:10:02 UTC
### R topics documented:

<table>
<thead>
<tr>
<th>Package/Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>eha-package</td>
<td>4</td>
</tr>
<tr>
<td>aftreg</td>
<td>5</td>
</tr>
<tr>
<td>aftreg.fit</td>
<td>7</td>
</tr>
<tr>
<td>age.window</td>
<td>9</td>
</tr>
<tr>
<td>cal.window</td>
<td>10</td>
</tr>
<tr>
<td>check.dist</td>
<td>11</td>
</tr>
<tr>
<td>check.surv</td>
<td>12</td>
</tr>
<tr>
<td>child</td>
<td>13</td>
</tr>
<tr>
<td>compHaz</td>
<td>14</td>
</tr>
<tr>
<td>coxfunk</td>
<td>15</td>
</tr>
<tr>
<td>coxreg</td>
<td>16</td>
</tr>
<tr>
<td>coxreg.fit</td>
<td>19</td>
</tr>
<tr>
<td>coxreg2</td>
<td>21</td>
</tr>
<tr>
<td>cro</td>
<td>24</td>
</tr>
<tr>
<td>EV</td>
<td>25</td>
</tr>
<tr>
<td>fert</td>
<td>26</td>
</tr>
<tr>
<td>frail.fit</td>
<td>27</td>
</tr>
<tr>
<td>geome.fit</td>
<td>28</td>
</tr>
<tr>
<td>Gompertz</td>
<td>29</td>
</tr>
<tr>
<td>hazards</td>
<td>30</td>
</tr>
<tr>
<td>HiscoHisclass</td>
<td>30</td>
</tr>
<tr>
<td>import_strata</td>
<td>31</td>
</tr>
<tr>
<td>import_Surv</td>
<td>31</td>
</tr>
<tr>
<td>infants</td>
<td>32</td>
</tr>
<tr>
<td>join.spells</td>
<td>33</td>
</tr>
<tr>
<td>Loglogistic</td>
<td>34</td>
</tr>
<tr>
<td>Lognormal</td>
<td>35</td>
</tr>
<tr>
<td>logrank</td>
<td>36</td>
</tr>
<tr>
<td>logrye</td>
<td>37</td>
</tr>
<tr>
<td>ltx</td>
<td>38</td>
</tr>
<tr>
<td>ltx2</td>
<td>39</td>
</tr>
<tr>
<td>make.communal</td>
<td>40</td>
</tr>
<tr>
<td>makeham</td>
<td>42</td>
</tr>
<tr>
<td>male.mortality</td>
<td>43</td>
</tr>
<tr>
<td>mlreg</td>
<td>44</td>
</tr>
<tr>
<td>mort</td>
<td>47</td>
</tr>
<tr>
<td>oe</td>
<td>48</td>
</tr>
<tr>
<td>oldmort</td>
<td>48</td>
</tr>
<tr>
<td>Pch</td>
<td>50</td>
</tr>
<tr>
<td>pchreg</td>
<td>51</td>
</tr>
<tr>
<td>perstat</td>
<td>53</td>
</tr>
<tr>
<td>phfunc</td>
<td>54</td>
</tr>
<tr>
<td>phreg</td>
<td>55</td>
</tr>
<tr>
<td>phreg.fit</td>
<td>58</td>
</tr>
<tr>
<td>piecewise</td>
<td>59</td>
</tr>
<tr>
<td>plot.aftreg</td>
<td>60</td>
</tr>
</tbody>
</table>
R topics documented:

plot.coxreg .................................................. 61
plot.hazdata .................................................. 63
plot.logrank .................................................. 64
plot.phreg .................................................... 66
plot.tpchreg .................................................. 67
plot.weibreg .................................................. 69
plotHaz ....................................................... 70
print.aftreg .................................................. 72
print.coxreg .................................................. 72
print.logrank .................................................. 73
print.phreg .................................................... 74
print.riskssets ............................................... 75
print.summary.aftreg ....................................... 76
print.summary.coxreg ....................................... 76
print.summary.phreg ....................................... 77
print.summary.tpchreg ...................................... 78
print.tpchreg ................................................ 79
print.weibreg ................................................ 79
regtable ...................................................... 80
risksets ....................................................... 81
scania .......................................................... 82
summary.aftreg ............................................... 83
summary.coxreg ............................................... 84
summary.phreg ............................................... 85
summary.tpchreg ............................................ 85
summary.weibreg ............................................ 86
SurvSplit ...................................................... 87
swedeaths ..................................................... 88
swepop .......................................................... 89
table.events ................................................... 90
toBinary ......................................................... 91
toDate .......................................................... 92
toTime .......................................................... 93
toTpch .......................................................... 94
tpchreg ........................................................ 95
weibreg ........................................................ 96
weibreg.fit .................................................... 99
Weibull .......................................................... 100
wfunk ........................................................... 101

Index .......................................................... 103
**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:

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

Description

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

Usage

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

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 `getOption("na.action")`.
- **dist**: Which distribution? Default is "weibull", with the alternatives "gompertz", "ev", "loglogistic" and "lognormal". A special case like the exponential can be obtained by choosing "weibull" in combination with `shape = 1`.
- **init**: vector of initial values of the iteration. Default initial value is zero for all variables.
- **shape**: If positive, the shape parameter is fixed at that value. If zero or negative, the shape parameter is estimated. Stratification is now regarded as a meaningful option even if shape is fixed.
- **id**: If there are more than one spell per individual, it is essential to keep spells together by the id argument. This allows for time-varying covariates.
param  Which parametrization should be used? The lifeAcc uses the parametrization given in the vignette, while the lifeExp uses the same as in the survreg function.

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

singular.ok  Not used.
model  Not used.
x  Return the design matrix in the model object?
y  Return the response in the model object?

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

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

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

\[ b - zbeta \]

is changed to

\[ b + zbeta \]

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

Value
A list of class "aftreg" 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.
`aftreg.fit`  

#### Terms
- **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.
- **param**: The parametrization.

#### Author(s)
Göran Broström

#### See Also
- `coxreg`, `phreg`, `survreg`

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

---

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

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

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

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)

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. If the selection gives an empty result, NULL is returned, with no warning.

Author(s)

Göran Broström

See Also

cal.window, coxreg, aftreg

Examples

dat <- data.frame(enter = 0, exit = 5.731, event = 1, x = 2)
window <- c(2, 5.3)
dat.trim <- age.window(dat, window)
Calendar time cut of survival data

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

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

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

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

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

Author(s)
Göran Broström

See Also
`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**

```r
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

**See Also**

`coxreg` and `phreg`.

**Examples**

```r
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.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 mortality, Skellefteå, Sweden 1850–1900.

Description

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

Usage

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.

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/.

Examples

fit <- coxreg(Surv(enter, exit, event) ~ sex + socBranch, data = child, coxph = TRUE)
summary(fit)
Graphical comparison of cumulative hazards

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

Usage
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
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

```
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`
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.
covars  The covariates.
ttr  Total Time at Risk.
levels  List of levels of factors.
formula  The calling formula.
bootstrap  The (matrix of) bootstrap replicates, if requested on input. It is up to the user to
do whatever desirable with this sample.
call  The call.
method  The method.
n.strata  Number of strata.
convergence  Did the optimization converge?
fail  Did the optimization fail? (Is NULL if not).

Warning

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

Note

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

Author(s)

Göran Broström

References


See Also

coxph, risksets

Examples

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),
  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
**Description**

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

**Usage**

```r
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))
```

---

### coxreg2

**Cox regression**

**Description**

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

**Usage**

```r
coxreg2(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 `options()`$na.action.
- `init` vector of initial values of the iteration. Default initial value is zero for all variables.
- `method` Method of treating ties, "efron" (default), "breslow", "mppl" (maximum partial partial likelihood), or "ml" (maximum likelihood).
control

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

`singular.ok`

Not used

`model`

Not used

`center`

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 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.

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

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

Number of events in data.

Used by extractor functions.

The Surv vector.

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 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. Suppling 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

cro

See Also
coxph, risksets

Examples

dat <- data.frame(time= c(4, 3,1,1,2,2,3),
                  status=c(1,1,0,1,0,0),
                  x= c(0, 2,1,1,0,0),
                  sex= c(0, 0,0,0,1,1))
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

---

cro

Creates a minimal representation of a data frame.

Description

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

Usage

cro(dat, response = 1)

Arguments

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

Details

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

Value

A list with components

y The response.
covar A data frame with unique rows of covariates.
keys Pointers from y to covar, connecting each response with its covariate vector.
Note
This function is based on suggestions by Anne York and Brian Ripley.

Author(s)
Göran Broström

See Also
match, paste

Examples

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

---

**EV**

### The EV Distribution

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

**Usage**

```r
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

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

for $x \geq 0$.

Value

dEV gives the density, pEV gives the distribution function, qEV gives the quantile function, hEV gives the hazard function, HEV gives the cumulative hazard function, and rEV generates random deviates. Invalid arguments will result in return value NaN, with a warning.

---

**fert**  
*Marital fertility nineteenth century*

---

Description

Birth intervals for married women with at least one birth, 19th northern Sweden

Usage

data(fert)

Format

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

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

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

Source

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

References

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)

frail.fit

Frailty experiment

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

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
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 \( a \to a/b \), so that \( b \) is a true scale parameter (for any fixed \( a \)), and \( b \) is an 'AFT parameter'. If param = "rate", then \( b \to 1/b \).
Value

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

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

hazards | Get baseline hazards atoms from fits from

Description

Get baseline hazards atoms from fits from

Usage

hazards(x, cum = TRUE, ...)

Arguments

x | A reg object.
cum | Logical: Should the cumulative hazards be returned?
... | Additional arguments for various methods.

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.

HiscoHisclass | HISCO to HISCLASS transformation

Description

HISCO to HISCLASS transformation

Usage

HiscoHisclass(hisco, status = NULL, relation = NULL, urban = NULL, debug = FALSE)
Arguments

- hisco: Hisco codes to be transformed to hisclass.
- status: Optional standard description of status.
- relation: Relation between owner of occupation and self.
- urban: Logical, "Is residence in an urban area?"
- debug: Logical, prints intermediate values if TRUE.

Value

A vector of hisclass codes, same length as input hisco.

Author(s)

Göran Broström with translation and modification of a Stata do.

References


---

**import_strata**

*strata function imported from survival*

**Description**

This function is imported from the *survival* package. See *strata*.

---

**import_Surv**

*Surv function imported from survival*

**Description**

This function is imported from the *survival* package. See *Surv*. 
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

```r
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

```r
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

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 \( \text{length}(n) > 1 \), the length is taken to be the number required.
log, log.p logical; if TRUE, probabilities \( p \) are given as \( \log(p) \).
prop proportionality constant in the extended Loglogistic distribution.

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

\[
h(x) = \left( \frac{b}{a} \right) \left( \frac{x}{a} \right)^{b-1} \exp \left( \frac{x}{a} \right)^b
\]
for \( x \geq 0 \).

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

The Lognormal Distribution

Description

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

Usage

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

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

for $x \geq 0$.

Value

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

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

*The Log-rank test*

Description

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

Usage

```r
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

```r
fit <- logrank(Y = Surv(enter, exit, event), group = civ, data = oldmort[oldmort$region == "town", ])
fit
```
Rye prices, Scania, southern Sweden, 1801-1894.

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)
ltx

LaTeX printing of regression results.

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
For printing confidence limits, use `ltx2`.

Author(s)
Göran Broström.

See Also
- `ltx2`, `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")
```

---

**ltx2**  
*LaTeX alternative printing of regression results.*

Description

This (generic) function prints the LaTeX code of the results of a fit from `coxreg`, `phreg`, `tpchreg`, or `aftreg`.

Usage

```r
ltx2(
  x,
  caption = NULL,
  label = NULL,
  dr = NULL,
  digits = max(options()$digits - 4, 4),
  conf = 0.95,
  keep = NULL,
  ...
)
```

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.  
- `conf`  
  Confidence intervals level.  
- `keep`  
  Number of covariates to present.  
- `...`  
  Not used.

Value

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

Note

Resulting tables contain estimated hazard ratios and confidence limits instead of regression coefficients and standard errors as in \texttt{ltx}.

Author(s)

Göran Broström.

See Also

\texttt{xtable}, \texttt{coxreg}, \texttt{phreg}, \texttt{phreg}, \texttt{aftreg}, and \texttt{ltx}.

Examples


data(oldmort)
fit <- coxreg(Surv(enter, exit, event) ~ sex, data = oldmort)
ltx2(fit, caption = "A test example.", label = "tab:test1")

make.communal

\textit{Put communals in "fixed" data frame}

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

\begin{verbatim}
make.communal(
  dat,
  com.dat,
  communal = TRUE,
  start,
  period = 1,
  lag = 0,
  surv = c("enter", "exit", "event", "birthdate"),
  tol = 1e-04,
  fortran = TRUE
)
\end{verbatim}
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 start and lag.
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).
The Gompertz-Makeham Distribution

Description

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

Usage

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 \( \text{length}(n) > 1 \), the length is taken to be the number required.
log, log.p logical; if TRUE, probabilities p are given as \( \log(p) \).

Details

The 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.
**Description**

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

**Usage**

```r
data(male.mortality)
```

**Format**

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

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

**Details**

The interesting explanatory covariate is `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 `mort`.

**Source**

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

**References**

[https://www.umu.se/enheten-for-demografi-och-aldrandeforskning/](https://www.umu.se/enheten-for-demografi-och-aldrandeforskning/)

**Examples**

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

**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 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.

**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**

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))
mlreg( Surv(time, status) ~ x + strata(sex), data = dat) #stratified model
# Same as:
mortality in ages 40-60, nineteenth century

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

Usage
data(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.
sez  Socio-economic status, a factor with levels lower, upper

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

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

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

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

```r
data(mort)
fit <- coxreg(Surv(enter, exit, event) ~ ses, data = mort)
summary(fit)
```

---

**oe**

*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

```r
data(oldmort)
summary(oldmort)
## maybe str(oldmort) ; plot(oldmort) ...```
**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**

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

**Arguments**

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

**Details**

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

$$0 < c_1 < \cdots < c_n < \infty, n \geq 0$$

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

**Value**

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

**Note**

the parameter `levels` must have length at least 1, and the number of cut points must be one less than the number of levels.
**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),
  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).
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
## Not run:
dat <- age.window(oldmort, c(60, 80))
fit <- pchreg(Surv(enter, exit, event) ~ ses.50 + sex,
data = dat, cuts = seq(60, 80, by = 4))
summary(fit)

fit.cr <- coxreg(Surv(enter, exit, event) ~ ses.50 + sex, data = dat)
check.dist(fit.cr, fit, main = "Cumulative hazards")

## End(Not run)
```

---

### perstat

#### Period statistics

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))
```

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`
Loglihood 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 cosidered in the partial derivatives.
- `dist`: Which distriution? 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

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
)

phreg        Parametric Proportional Hazards Regression
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

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

where S0 is some standardized survivor function.

Value

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

**coefficients**
Fitted parameter estimates.
cuts
hazards
var
loglik
score
linear.predictors
means
w.means
n
n.events
terms
assign
wald.test
y
isF
covars
ttr
levels
formula
call
method
convergence
fail
pfixed

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.

Author(s)
Göran Broström

See Also
phreg

Descriptive

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.
plot.aftreg

Value
A list with components

- events: Vector of number of events
- exposure: Vector of total exposure time
- intensity: Vector of hazards, \( \text{intensity} = \frac{\text{events}}{\text{exposure}} \)

Author(s)
Göran Broström

See Also
perstat

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 = ...)`
**plot.coxreg**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>main</td>
<td>Header for the plot</td>
</tr>
<tr>
<td>xlim</td>
<td>x limits</td>
</tr>
<tr>
<td>ylim</td>
<td>y limits</td>
</tr>
<tr>
<td>xlab</td>
<td>x label</td>
</tr>
<tr>
<td>ylab</td>
<td>y label</td>
</tr>
<tr>
<td>col</td>
<td>Colors?</td>
</tr>
<tr>
<td>lty</td>
<td>Line types?</td>
</tr>
<tr>
<td>printLegend</td>
<td>Should legend be printed? Default is yes.</td>
</tr>
<tr>
<td>new.data</td>
<td>At which covariate values?</td>
</tr>
<tr>
<td>...</td>
<td>Extra parameters passed to 'plot'</td>
</tr>
</tbody>
</table>

**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 <- rlogis(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

## 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

x              A coxreg object
fn             What should be plotted? Default is "cumhaz", and the other choices are "surv", "log", and "loglog".
conf.int       logical or a value like 0.95 (default for one curve).
fig            logical. If TRUE the plot is actually drawn, otherwise only the coordinates of the curve(s) are returned.
xlim           Start and end of the x axis.
ylim           Start and end of the y axis.
main           A headline for the plot
xlab           Label on the x axis.
ylab           Label on the y axis.
col            Color of the curves. Defaults to 'black'.
lty             Line type(s).
printLegend    Either a logical or a text string; if TRUE, a legend is printed at a default place, if FALSE, no legend is printed. Otherwise, if a text string, it should be one of "bottomleft", "bottomright", "topleft", etc., see legend for all possible choices.
...             Other parameters to pass to the plot.

Value

An object of class hazdata containing the coordinates of the curve(s).
Description

Baseline hazards estimates.

Usage

## 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 = "",
  col = "black",
  lty = 1,
  printLegend = TRUE,
  ...
)

Arguments

x     A hazdata object, typically the 'hazards' element in the output from link{coxreg} with method = "ml" or method = "mpp1" 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

```r
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

```r
## S3 method for class 'logrank'
plot(
    x,
    fn = c("cum", "surv", "log", "loglog"),
    xlim = NULL,
    ylim = NULL,
    main = NULL,
```
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.
- **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[oldmort$region == "town", ])
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

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

plot.tpchreg  Plots output from a tpchreg regression

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, eg, `par(mfrow = ...)`
- `main`: Header for the plot
- `xlim`: x limits
- `ylim`: y limits
- `xlab`: x label
- `ylab`: y label
- `new.data`: At which covariate values?
- `...`: Extra parameters passed to `plot`

Details

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

Value

No return value
Author(s)
Göran Broström

See Also
phreg, weibreg

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**

```r
plotHaz(
  sp,  
  pp,  
  interval,  
  main = NULL,  
  xlab = "Time",  
  ylab = "Cum. hazards",  
  leglab,  
  col = c("blue", "red"),  
  lty = 1:2,  
  ylim,  
  log = ",",  
  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"
plotHaz

xlab Label on x axis (default "Time")
ylab Label on y axis (default "Cum. Hazards")
leglab Labels in legend.
col Line colors. should be NULL (black lines) or of length 2
lty line types.
ylim Y limits for the plot.
log Argument sent to plot, defaults to "".
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)
print.aftreg  

Prints aftreg objects

Description

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

Usage

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

Arguments

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

Value

No value is returned.

Note

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

Author(s)

Göran Broström

See Also

- `phreg`, `print.coxph`

print.coxreg  

Prints coxreg objects

Description

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

Usage

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

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

Details

Doesn’t work with three-way and higher interactions, in which case print.coxph is used.

Value

No value is returned.

Author(s)

Göran Broström

See Also

coxreg, print.coxph

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), ...)

Arguments

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

Value

No value is returned.

Note

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

Author(s)

Göran Broström

See Also

phreg, print.coxph
print.risksets

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

Description

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

Usage

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

Arguments

- `x`  
  An object of class 'risksets'.

- `...`  
  Not used for the moment.

Value

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

Note

There is no summary.risksets yet. On the TODO list.

Author(s)

Göran Broström

See Also

risksets

Examples

```r
rs <- with(mort, risksets(Surv(enter, exit, event)))
print(rs)
```
print.summary.aftreg  Prints summary.aftreg objects

Description
Prints summary.aftreg objects

Usage

## 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` or `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`

Description

Prints `summary.phreg` objects

Usage

```r
## 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.
- **digits**: Output format.
- **...**: Other arguments.

Value

No value is returned.

Author(s)

Göran Broström
print.summary.tpchreg

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

Prints tpchreg objects

Description

More "pretty-printing"

Usage

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

Arguments

x A tpchreg object, typically the result of running tpchreg
digits Output format.
...

Details

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

Value

No value is returned.

Author(s)

Göran Broström

See Also

tpchreg, print.coxreg

print.weibreg

Prints weibreg objects

Description

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

Usage

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

- \(x\) A `weibreg` object
- `digits` Precision in printing
- `...` Not used.

**Value**

No value is returned.

**Note**

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

**Author(s)**

Göran Broström

**See Also**

`weibreg`, `print.coxph`

---

**Description**

Retrieves regression tables

**Usage**

```r
regtable(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.
### Description

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

### Usage

```r
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

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

---

scania  

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

Description

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

Usage

   data(scania)
Format

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

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

Details

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

Source

The Scanian Economic Demographic Database, Lund University, Sweden.

References

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

Examples

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

____________

*summary.aftreg*  
*Prints aftreg objects*

____________

Description

Prints aftreg objects

Usage

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

Arguments

- **object**  A aftreg object
- **...**  Additional ...
Author(s)

Göran Broström

See Also

print.coxreg

Examples

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

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

## 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[oldmort$region == "town", ])  
summary(fit)
```

---

Summary for tpchreg objects

## S3 method for class `tpchreg`

```
summary(object, ci = FALSE, level = 0.95, ...)  
```

### Arguments

- `object` A tpchreg object
- `ci` FALSE, level = 0.95, ...

### Examples

```r
```
summary.weibreg

Arguments

- `object`: A `tpchreg` object.
- `ci`: Logical. If `TRUE`, confidence limits are given instead of se’s.
- `level`: Confidence level, used if `ci`.
- `...`: Additional ...

Author(s)

- Göran Broström

See Also

- `tpchreg`

Examples

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

---

**summary.weibreg**

*Prints a weibreg object*

Description

This is the same as `print.weibreg`

Usage

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

Arguments

- `object`: A `weibreg` object
- `...`: Additional ...

Author(s)

- Göran Broström

See Also

- `print.weibreg`
SurvSplit

Examples

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

SurvSplit  Split a survival object at specified durations.

Description

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

Usage

SurvSplit(Y, cuts)

Arguments

- **Y**: A survival object, a matrix with two or three columns.
- **cuts**: The cut points, must be strictly positive and distinct.

Value

A list with components

- **Y**: The new survival object with three columns, i.e., in 'counting process' form.
- **ivl**: Interval No., starting from leftmost, (0, cuts[1]) or similar.
- **idx**: Row number for original Y row.

Note

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

Author(s)

Göran Broström

See Also

`survSplit, age.window`
Examples

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

## The function is currently defined as
function(Y, cuts){
  if (NCOL(Y) == 2) Y <- cbind(rep(0, NROW(Y)), Y)
  indat <- cbind(Y, 1:NROW(Y), rep(-1, NROW(Y)))
  colnames(indat) <- c("enter", "exit", "event", "idx", "ivl")
  n <- length(cuts)
  cuts <- sort(cuts)
  if ((cuts[1] <= 0) || (cuts[n] == Inf))
    stop("'cuts' must be positive and finite.")
  cuts <- c(0, cuts, Inf)
  n <- n + 1
  out <- list()
  indat <- as.data.frame(indat)
  for (i in 1:n){
    out[[i]] <- age.window(indat, cuts[i:(i+1)])
    out[[i]]$ivl <- i
    out[[i]] <- t(out[[i]])
  }
  Y <- matrix(unlist(out), ncol = 5, byrow = TRUE)
  colnames(Y) <- colnames(indat)
  list(Y = Y[, 1:3],
       ivl = Y[, 5],
       idx = Y[, 4]
  )
}
```

---

swedeaths

*Swedish death data, 1969-2020.*

Description

A data frame containing data on the number of deaths by sex, age and year, Sweden 1969-2020.

Usage

```r
data(swedeaths)
```

Format

A data frame with 5 variables and 10504 observations.

```r
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 number of deaths by sex and age. The original data set is in wide form and then converted to long format.

Source

See Also
swepop, tpchreg

Examples
summary(swedeaths)
## maybe str(swedeaths) ...

---


Description
A data frame containing data on the population size by sex, age and year, Sweden 1969-2020.

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 coverted to factors. The variable 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 reshape.

Source


See Also

swedeaths

Examples

summary(swepop)
## maybe str(swepop) ...

---

### table.events

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

Description

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

Usage

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

Arguments

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

Value

A list with components

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

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`, `glm`.

Examples

```r
enter <- rep(0, 4)
ext <- 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.
toTime

Usage
toDate(times)

Arguments
times a vector of durations

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

Author(s)
Göran Broström

See Also
toTime

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.
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**

```r
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

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.
weibreg

Note

The interpretation of cuts is different from that in `hpch`. This is intentional.

See Also

`oe`.

Examples

```r
sw <- swepop
sw$deaths <- swdeaths$deaths
fit <- tpchreg(oe(deaths, pop) ~ strata(sex) + I(year - 2000),
              time = age, last = 101, data = sw[sw$year >= 2000, ])
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(eps = 1e-04, maxiter = 10, trace = FALSE),
  singular.ok = TRUE,
  model = FALSE,
  x = FALSE,
  y = TRUE,
  center = TRUE
)
```

Arguments

- `formula`: a formula object, with the response on the left of a `~` operator, and the terms on the right. The response must be a survival object as returned by the `Surv` function.
weibreg

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.

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

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

singular.ok Not used.

model Not used.

x Return the design matrix in the model object?

y Return the response in the model object?

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

Details

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

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

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 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).
weibreg

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(scale), but nothing else. If you don’t like this, set center = FALSE.

Note

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

Author(s)

Göran Broström

See Also

phreg, coxreg, print.weibreg

Examples

dat <- data.frame(time = c(4, 3, 1, 1, 2, 2, 3),
                  status = c(1, 1, 1, 0, 1, 1, 0),
                  x = c(0, 2, 1, 1, 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

```r
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 log likelihood with only scale and shape in the model, the second the final maximum.
- `score linear.predictors` Score test statistic at initial values
  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.
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**

```r
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**

`weibreg`, `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**

```r
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

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

This is in correspondence with \texttt{dweibull}.

Value
A list with components
- \texttt{f} The log likelihood. Present if \texttt{ord} >= 0
- \texttt{fp} The score vector. Present if \texttt{ord} >= 1
- \texttt{fpp} The negative of the hessian. Present if \texttt{ord} >= 2

Author(s)
Göran Broström

See Also
- \texttt{weibreg}
Index

* cluster
toBinary, 91

* datasets
child, 13
fert, 26
infants, 32
logrye, 37
male.mortality, 43
mort, 47
oldmort, 48
scania, 82
swedeaths, 88
swepop, 89

* distribution
check.dist, 11
compHaz, 14
coxfunk, 15
EV, 25
Gompertz, 29
Loglogistic, 34
Lognormal, 35
makeham, 42
Pch, 50
phfunc, 54
plotHaz, 70
wfunk, 101

* dplot
plot.aftreg, 60
plot.phreg, 66
plot.tpchreg, 67
plot.weibreg, 69

* manip
check.surv, 12
cro, 24
join.spells, 33
SurvSplit, 87
toTpch, 94

* nonparametric
perstat, 53
piecewise, 59

* printing
ltx, 38
ltx2, 39

* print
summary.aftreg, 83
summary.coxreg, 84
summary.weibreg, 86

* regression
aftreg, 5
aftreg.fit, 7
coxreg, 16
coxreg.fit, 19
coxreg2, 21
mlreg, 44
phreg, 51
phreg.fit, 58
print.aftreg, 72
print.logrank, 73
print.phreg, 74
print.weibreg, 79
tpchreg, 95
weibreg, 96
weibreg.fit, 99

* summary
print.risksets, 75
summary.tpchreg, 85

* survival
aftreg, 5
aftreg.fit, 7
age.window, 9
cal.window, 10
check.surv, 12
coxfunk, 15
coxreg, 16
coxreg.fit, 19
coxreg2, 21
geome.fit, 28
<table>
<thead>
<tr>
<th>Function</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>join.spells</td>
<td>33</td>
</tr>
<tr>
<td>logrank</td>
<td>36</td>
</tr>
<tr>
<td>make.community</td>
<td>40</td>
</tr>
<tr>
<td>mlreg</td>
<td>44</td>
</tr>
<tr>
<td>pchreg</td>
<td>51</td>
</tr>
<tr>
<td>perstat</td>
<td>53</td>
</tr>
<tr>
<td>phfunc</td>
<td>54</td>
</tr>
<tr>
<td>phreg</td>
<td>55</td>
</tr>
<tr>
<td>phreg.fit</td>
<td>58</td>
</tr>
<tr>
<td>piecewise</td>
<td>59</td>
</tr>
<tr>
<td>plot.aftreg</td>
<td>60</td>
</tr>
<tr>
<td>plot.hazdata</td>
<td>63</td>
</tr>
<tr>
<td>plot.logrank</td>
<td>64</td>
</tr>
<tr>
<td>plot.phreg</td>
<td>66</td>
</tr>
<tr>
<td>plot.tpchreg</td>
<td>67</td>
</tr>
<tr>
<td>plot.weibreg</td>
<td>69</td>
</tr>
<tr>
<td>print.aftreg</td>
<td>72</td>
</tr>
<tr>
<td>print.coxreg</td>
<td>72</td>
</tr>
<tr>
<td>print.logrank</td>
<td>73</td>
</tr>
<tr>
<td>print.phreg</td>
<td>74</td>
</tr>
<tr>
<td>print.summary.aftreg</td>
<td>76</td>
</tr>
<tr>
<td>print.summary.coxreg</td>
<td>76</td>
</tr>
<tr>
<td>print.summary.phreg</td>
<td>77</td>
</tr>
<tr>
<td>print.summary.tpchreg</td>
<td>78</td>
</tr>
<tr>
<td>print.tpchreg</td>
<td>79</td>
</tr>
<tr>
<td>print.weibreg</td>
<td>79</td>
</tr>
<tr>
<td>regtable</td>
<td>80</td>
</tr>
<tr>
<td>risksets</td>
<td>81</td>
</tr>
<tr>
<td>summary.aftreg</td>
<td>83</td>
</tr>
<tr>
<td>summary.coxreg</td>
<td>84</td>
</tr>
<tr>
<td>summary.phreg</td>
<td>85</td>
</tr>
<tr>
<td>summary.tpchreg</td>
<td>85</td>
</tr>
<tr>
<td>summary.weibreg</td>
<td>86</td>
</tr>
<tr>
<td>table.events</td>
<td>90</td>
</tr>
<tr>
<td>tobinary</td>
<td>91</td>
</tr>
<tr>
<td>toDate</td>
<td>92</td>
</tr>
<tr>
<td>toTime</td>
<td>93</td>
</tr>
<tr>
<td>tpchreg</td>
<td>95</td>
</tr>
<tr>
<td>weibreg</td>
<td>96</td>
</tr>
<tr>
<td>weibreg.fit</td>
<td>99</td>
</tr>
<tr>
<td>Weibull</td>
<td>100</td>
</tr>
<tr>
<td>wfunk</td>
<td>101</td>
</tr>
</tbody>
</table>

* table |

<table>
<thead>
<tr>
<th>Function</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>aftreg</td>
<td>5, 7–10, 12, 34, 38–41, 61, 76</td>
</tr>
<tr>
<td>aftreg.fit</td>
<td>7</td>
</tr>
<tr>
<td>age.window</td>
<td>9, 10, 87</td>
</tr>
<tr>
<td>cal.window</td>
<td>9, 10, 41</td>
</tr>
<tr>
<td>check.dist</td>
<td>11, 10, 41</td>
</tr>
<tr>
<td>check.surv</td>
<td>12, 34</td>
</tr>
<tr>
<td>child</td>
<td>13</td>
</tr>
<tr>
<td>compHaz</td>
<td>14</td>
</tr>
<tr>
<td>coxph</td>
<td>4, 17, 18, 22, 24, 41, 56, 97</td>
</tr>
<tr>
<td>coxreg</td>
<td>4, 7, 9–12, 14, 15, 16, 19, 20, 28, 34, 36, 38–41, 45, 46, 52, 56, 58, 70, 71, 73, 74, 77, 81, 82, 92, 97, 98</td>
</tr>
<tr>
<td>coxreg.fit</td>
<td>19</td>
</tr>
<tr>
<td>coxreg2</td>
<td>21</td>
</tr>
<tr>
<td>cro</td>
<td>24</td>
</tr>
<tr>
<td>dEV (EV)</td>
<td>25</td>
</tr>
<tr>
<td>dgompertz (Gompertz)</td>
<td>29</td>
</tr>
<tr>
<td>dllogis (Loglogistic)</td>
<td>34</td>
</tr>
<tr>
<td>dlnorm (Lognormal)</td>
<td>35</td>
</tr>
<tr>
<td>dmakeham (makeham)</td>
<td>42</td>
</tr>
<tr>
<td>dpch (Pch)</td>
<td>50</td>
</tr>
<tr>
<td>dweibull</td>
<td>100, 102</td>
</tr>
<tr>
<td>eha (eha-package)</td>
<td>4</td>
</tr>
<tr>
<td>eha-package</td>
<td>4</td>
</tr>
<tr>
<td>EV</td>
<td>25</td>
</tr>
<tr>
<td>fert</td>
<td>26</td>
</tr>
<tr>
<td>frail.fit</td>
<td>27</td>
</tr>
<tr>
<td>geome.fit</td>
<td>28</td>
</tr>
<tr>
<td>glm</td>
<td>92</td>
</tr>
<tr>
<td>Gompertz</td>
<td>29, 56</td>
</tr>
<tr>
<td>gompertz (Gompertz)</td>
<td>29</td>
</tr>
<tr>
<td>hazards</td>
<td>14, 30</td>
</tr>
<tr>
<td>HEV (EV)</td>
<td>25</td>
</tr>
<tr>
<td>hEV (EV)</td>
<td>25</td>
</tr>
<tr>
<td>Hgompertz (Gompertz)</td>
<td>29</td>
</tr>
<tr>
<td>hgompertz (Gompertz)</td>
<td>29</td>
</tr>
<tr>
<td>HiscoHisclass</td>
<td>30</td>
</tr>
<tr>
<td>Hllogis (Loglogistic)</td>
<td>34</td>
</tr>
<tr>
<td>hllogis (Loglogistic)</td>
<td>34</td>
</tr>
<tr>
<td>Hlnorm (Lognormal)</td>
<td>35</td>
</tr>
<tr>
<td>hlnorm (Lognormal)</td>
<td>35</td>
</tr>
<tr>
<td>Hmakeham (makeham)</td>
<td>42</td>
</tr>
<tr>
<td>hmakeham (makeham)</td>
<td>42</td>
</tr>
<tr>
<td>Hpch (Pch)</td>
<td>50</td>
</tr>
<tr>
<td>hpch</td>
<td>96</td>
</tr>
<tr>
<td>hpch (Pch)</td>
<td>50</td>
</tr>
</tbody>
</table>
swedeaths, 88, 90
swepop, 89, 89

table.events, 82, 90
toBinary, 91
toDate, 92, 94
toTime, 93, 93
toTpch, 94
tpchreg, 38, 39, 48, 68, 78, 79, 86, 89, 95

weibreg, 70, 80, 96, 99–102
weibreg.fit, 99
Weibull, 97, 100
wfunk, 101