Package ‘coxphw’

February 19, 2015

Type Package

Title Weighted Estimation in Cox Regression

Version 3.0.0

Date 2014-04-07

Author R by Meinhard Ploner, Georg Heinze, Daniela Dunkler, Fortran by Georg Heinze

Maintainer Georg Heinze <georg.heinze@meduniwien.ac.at>

Description This package implements weighted estimation in Cox regression as proposed by Schemper, Wakounig and Heinze (Statistics in Medicine, 2009). Weighted Cox regression provides unbiased average hazard ratio estimates also in case of non-proportional hazards. Approximated generalized concordance probability an effect size measure for clear-cut decisions can be obtained.

Additionally estimation of nonlinear effects using fractional polynomials similar to the MFP algorithm (Royston, Sauerbrei, 2008) is provided. This feature can also be used to estimate the interaction of a covariate with a nonlinear function of time.

Depends R (>= 3.0.2), survival

NeedsCompilation yes

License GPL-2

Repository CRAN

Date/Publication 2014-04-08 16:40:22

R topics documented:

coxphw-package ......................................................... 2
biofeedback .......................................................... 4
coeff.coxphw ......................................................... 5
concord ............................................................. 5
confint.coxphw ...................................................... 7
coxphw ............................................................ 7
fp.power ............................................................ 13
gastric ............................................................. 14
plotshape .......................................................... 15
plotw .............................................................. 19
This package implements weighted estimation in Cox regression as proposed by Schenper, Wakounig and Heinze (Statistics in Medicine, 2009). Weighted Cox regression provides unbiased average hazard ratio estimates also in case of non-proportional hazards. Additionally estimation of nonlinear effects using fractional polynomials similar to the MFP algorithm (Royston, Sauerbrei, 2008) is provided. This feature can also be used to estimate the interaction of a covariate with a nonlinear function of time.

Main functions included in the coxphw package are

- `coxphw`: weighted estimation of Cox regression: either (recommended) estimation of average hazard ratios (Schemper et al., 2009), estimation of average regression effects (Xu and O'Quigley, 2000), or proportional hazards regression. Estimation of nonlinear effects using fractional polynomials is possible.
- `concord`: obtain generalized concordance probabilities with confidence intervals.
- `plotw`: plots the weights used in a weighted Cox regression analysis against time.
- `fp.power`: provides fractional polynomials as accessible function.
- `plotshape`: either visualizes a nonlinear or a time-dependent effect (possibly estimated by fractional polynomials) of a coxphw fit or computes the effect estimates at pre-specified values. It plots the relative or log relative hazard versus values of a continuous covariable.
- `wald`: obtain Wald chi-squared test statistics and p-values for one or more regression coefficients given their variance-covariance matrix.

Data sets included in the coxphw package are
Note

The SAS macro WCM with similar functionality (but without fractional polynomials) can be obtained at http://cemiis.meduniwien.ac.at/en/kb/science-research/software/statistical-software/wcmcoxphw/.

Up to Version 2.13 coxphw used a slightly different syntax (arguments: AHR, AHR.norobust, ARE, PH, normalize, censcorr, prentice, breslow, taroneware). From Version 3.0 on the old syntax is disabled.

Author(s)

R by Meinhard Ploner, Georg Heinze, Daniela Dunkler; Fortran by Georg Heinze
Maintainer: <georg.heinze@meduniwien.ac.at>

References


See Also

coxphw, concord, plotshape, plotw, fp.power, wald
Examples

## for examples see coxphw

---

biofeedback  
*Biofeedback Treatment Data*

**Description**

In this study the effect of biofeedback treatment on time until treatment success was evaluated in patients suffering from aspiration after head and neck surgery. The outcome of interest was the time from start of treatment until the patient achieved full swallowing rehabilitation (`thdur`). Patients were randomized into two groups (`bfb`): one group of patients received videoendoscopic biofeedback treatment; the other group received the conservative treatment including thermal stimulation with ice and exercises for the lips, tongue, laryngeal closure and elevation. Treatment was started as soon as the healing process after surgery was finished (`theal`).

**Usage**

data(biofeedback)

**Format**

A data frame with 33 observations on the following 6 variables:

- `id` the patient id.
- `success` of treatment within the first 100 days; either 0 = no success or 1 = success.
- `thdur` the duration of therapy in days.
- `bfb` indicates the treatment group; either 0 = conservative or 1 = biofeedback.
- `theal` time from surgery to start of therapy in days.
- `log2heal` log2-transformed time from surgery to start of therapy.

**Source**

Data were supplied by Dr. D.-M. Denk, who gave permission to freely distribute the data and use them for non-commercial purposes.

**References**

Examples

data("biofeedback")

plot(survfit(Surv(thdur, success) ~ bfb, data = biofeedback), lty = 1:2, las = 1,
xlab = "time (days)", ylab = "probability of success")

coxphw(Surv(thdur, success) ~ bfb, data = biofeedback, template = "AH")

calcuar de coeficientes de coef.coxphw

Description

This class of objects is returned by the coxphw function. Objects of this class have methods for the functions summary, print, coef, vcov, and confint.

Usage

## S3 method for class 'coxphw'
coef(object, ...)

Arguments

object

... further arguments.

Author(s)

Daniela Dunkler

See Also

coxphw

calculate concordance

Description

Compute generalized concordance probabilities with accompanying confidence intervals for objects of class coxphw or coxph.
Usage

concord(fit, digits = 4)

Arguments

fit an object of class coxphw.
digits integer indicating the number of decimal places to be used. Default is 4.

Details

The generalized concordance probability is defined as \( P(T_i < T_j | x_i = x_j + 1) \) with \( T_i \) and \( T_j \) as survival times of randomly chosen subjects with covariate values \( x_i \) and \( x_j \), respectively. Assuming that \( x_i \) and \( x_j \) are 1 and 0, respectively, this definition includes a two-group comparison.

If proportional hazards can be assumed, the generalized concordance probability can also be derived from Cox proportional hazards regression (coxphw with template = "PH" or coxph) or weighted Cox regression as suggested by Xu and O'Quigley (2000) (coxphw with template = "ARE").

If in a fit to coxphw the betafix argument was used, then for the fixed parameters only the point estimates are given.

Value

A matrix with estimates of the generalized concordance probability with accompanying confidence intervals for each explanatory variable in the model.

Author(s)

Daniela Dunkler

References


See Also

coxphw

Examples

data("gastric")
fit <- coxphw(Surv(time, status) ~ radiation, data = gastric, template = "AHR")
concord(fit)
Description

Confidence Intervals for Model Parameters

Computes confidence intervals for one or more parameters in a model fitted by coxphw. Objects of this class have methods for the functions summary, print, coef, vcov, and confint.

Usage

```r
## S3 method for class 'coxphw'
confint(object, parm, level = 0.95, ...)
```

Arguments

- `object`: a fitted model object of class `coxphw`.
- `parm`: a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
- `level`: the confidence level required.
- `...`: additional argument(s) for methods.

Value

A matrix (or vector) with columns giving lower and upper confidence limits for each parameter. These will be labeled as (1-level)/2 and 1 - (1-level)/2 in % (by default 2.5% and 97.5%).

Author(s)

Daniela Dunkler

See Also

coxphw

Description

Weighted Estimation in Cox Regression

Weighted Cox regression provides as proposed by Schemper et al. (2009) unbiased estimates of average hazard ratios also in case of non-proportional hazards. Additionally estimation of nonlinear effects using fractional polynomials similar to the MFP algorithm is provided. This feature can also be used to estimate the interaction of a covariate with a nonlinear function of time.
Usage

coxphw(formula = attr(data, "formula"), data = sys.parent(),
        template = c("AHR", "ARE", "PH"), robust = TRUE, jack = FALSE,
        betafix = NULL, alpha = 0.05, alpha.fp = c(0.2, 0.05, 0.05),
        fp.iter = 10, fp.max = 2, trunc.weights = 1, offset = NULL, maxit = 200,
        maxhs = 5, xconv = 1e-4, gconv = 1e-4, maxstep = 1, x = TRUE, y = TRUE,
        round.times.to = 0.00001, add.constant = 0, print = FALSE, sorted = FALSE,
        pc = TRUE, pc.time = TRUE, id = NULL, normalize = TRUE, ...)

Arguments

 formula a formula object with the response on the left of the operator and the model terms on the right. The response must be a survival object as returned by Surv. formula may include fp()-terms requesting the fractional polynomial functionality of coxphw.

data a data frame in which to interpret the variables named in formula.

template choose among three pre-defined templates: "AHR" requests estimation of average hazard ratios (Schemper et al., 2009), "ARE" requests estimation of average regression effects (Xu and O'Quigley, 2000) and "PH" requests Cox proportional hazards regression. Recommended and default template is "AHR".

robust if set to TRUE, the robust covariance estimate (Lin-Wei) is used; otherwise the Lin-Sasieni covariance estimate is applied. Default is TRUE.

jack if set to TRUE, the variance is based on a complete jackknife. Each individual (as identified by id) is left out in turn. The resulting matrix of DFBETA residuals D is then used to compute the variance matrix: V = D'D. Default is FALSE.

betafix can be used to restrict the estimation of one or more regression coefficients to pre-defined values. A vector with one element for each model term as given in formula is expected (with an identical order as in formula). If estimation of a model term is requested, then the corresponding element in betafix has to be set to NA, otherwise it should be set to the fixed parameter value. The default value is betafix = NULL, yielding unrestricted estimation of all regression coefficients.

alpha the significance level (1-\(\alpha\)), 0.05 as default.

alpha.fp a vector of length three which specifies p-values to include a fractional polynomial fp()-term: alpha.fp[1] defines the threshold p-value for keeping at least the linear term in the model, alpha.fp[2] defines the threshold p-value for the nonlinear term, and alpha.fp[3] defines the threshold p-value for the complexity of the nonlinear term (either a fractional polynomial of degree 1 or 2). Default is c(0.20, 0.05, 0.05).

fp.iter number of iterations for the big fractional polynomial-loop, default is 10. This may be important if several fp()-terms are included in the model as inclusion or exclusion of one term based on the threshold p-value may increase or decrease the significance of others.

fp.max highest power of fp()-terms applied to all fp()-terms. Select 1 or 2, with 2 as default.
trunc.weights specifies a quantile at which the (combined normalized) weights are to be truncated. It can be used to increase the precision of the estimates, particularly if `template = "AHR"` or "ARE" is used. Default is 1 (no truncation). Recommended value is 0.95 for mild truncation.

offset specifies a variable which is included in the model but its parameter estimate is fixed at 1. A numeric vector is expected. It can also be used to include case weights into the analysis, by supplying the logarithm of these weights in the offset argument.

normalize if set to TRUE, weights are normalized such that their sum is equal to the number of events. This may speed up or enable convergence, but has no consequences on the estimated regression coefficients.

maxit maximum number of iterations. Default is 200.

maxhs maximum number of step-halvings per iterations. Default is 5. The increments of the parameter vector in one Newton-Raphson iteration step are halved, unless the new likelihood is greater than the old one, maximally doing `maxhs` halvings.

xconv specifies the maximum allowed change in standardized parameter estimates to declare convergence. Default is 0.0001.

gconv specifies the maximum allowed change in score function to declare convergence. Default is 0.0001.

maxstep specifies the maximum change of (standardized) parameter values allowed in one iteration. Default is 1.

x requests copying explanatory variables into the output object. Default is TRUE.

y requests copying survival information into the output object. Default is TRUE.

round.times.to rounds survival times to the nearest number that is a multiple of `round.times.to`. This may improve numerical stability if very small survival times are used (mostly in simulations). Default is 0.00001.

add.constant this number will be added to all times. It can be used if some survival times are exactly 0. Default is 0.

print requests echoing of intermediate results (particularly, for estimation with `fp()`-terms). Default is FALSE.

sorted if set to TRUE, the data set will not be sorted prior to passing it to FORTRAN. This may speed up computations. Default is FALSE.

pc if set to TRUE, it will transform the model matrix to speed up convergence during `fp()`-mode. Default is TRUE.

pc.time if set to TRUE, it will transform time-dependent covariates (i.e., interactions of covariates with functions of time) to speed up convergence. Default is TRUE.

id a vector of subject identification integer numbers starting from 1. These IDs are used for computing the robust covariance matrix. If `id = NA` (the default) the program assumes that each line of the data set refers to a distinct subject.

... additional arguments.
Details

If Cox’s proportional hazards regression is used in the presence of non-proportional hazards, i.e., with underlying time-dependent hazard ratios of prognostic factors, the average relative risk for such a factor is under- or overestimated and testing power for the corresponding regression parameter is reduced. In such a situation weighted estimation provides a parsimonious alternative to more elaborate modelling of time-dependent effects. Weighted estimation in Cox regression extends the tests by Breslow and Prentice to a multi-covariate situation as does the Cox model to Mantel’s logrank test. Weighted Cox regression can also be seen as a robust alternative to the standard Cox estimator, reducing the influence of outlying survival times on parameter estimates.

Three pre-defined templates can be requested: 1) "AHR", i.e., estimation of average hazard ratios (Schemper et al., 2009) using Prentice weights with censoring correction and robust variance estimation; 2) "ARE", i.e., estimation of average regression effects (Xu and O’Quigley, 2000) using censoring correction and robust variance estimation; or 3) "PH", i.e., Cox proportional hazards regression using robust variance estimation.

Breslow’s tie-handling method is used by the program, other methods to handle ties are currently not available.

A fit of coxphw with template = "PH" will yield identical estimates as a fit of coxph using Breslow’s tie handling method and robust variance estimation (using cluster).

If robust = FALSE, the program estimates the covariance matrix using the Lin (1991) and Sasieni (1993) sandwich estimate $A^{-1}BA^{-1}$ with $-A$ and $-B$ denoting the sum of contributions to the second derivative of the log likelihood, weighted by $w(t_j)$ and $w(t_j)^2$, respectively. This estimate is independent from the scaling of the weights and reduces to the inverse of the information matrix in case of no weighting. However, it is theoretically valid only in case of proportional hazards. Therefore, since application of weighted Cox regression usually implies a violated proportional hazards assumption, the robust Lin-Wei covariance estimate is used by default (robust = TRUE).

Estimation of nonlinear effects of continuous variables using fractional polynomials similar to the MFP algorithm (Royston and Altman, 1994; Royston and Sauerbrei, 2008) is available. (It is similar, but not identical to mfp, e.g., tests are based on robust Wald test statistics instead of likelihood ratio statistics.) Estimation of fractional polynomials can also be applied to a survival time in a time-by-covariate interaction. A pretransformation (by shifting to positive values and dividing by scaling factor [a power of 10] such that the standard deviation is approximately equal to one) may be applied to continuous variables for which fractional polynomials are requested. For examples see the documentation of plotshape.

betafix and estimation of fractional polynomials cannot be requested at the same time. If betafix is requested for one or more regression coefficients, no estimates on inference are given for these coefficients as they were not estimated in the model. The global Wald test only relates to those variables for whom regression coefficients were estimated.

Value

A list with the following components:

- coefficients: the parameter estimates.
- var: the estimated covariance matrix.
- df: the degrees of freedom.
- ci.lower: the lower confidence limits of exp(beta).
ci.upper  the upper confidence limits of exp(beta).
prob      the p-values.
linear.predictors  the linear predictors.
n      the number of observations.
dfbeta.resid  matrix of DFBETA residuals.
iter      the number of iterations needed to converge.
method.ties  the ties handling method.
PTcoefs  matrix with scale and shift used for pretransformation of fp(\cdot)-terms.
fp.ind     if fp(\cdot)-terms were used, a matrix with all transformations of fractional polynomials used in the fitting process.
ind      if fp(\cdot)-terms were used, an indicator which variables from all transformations of fractional polynomials used in the fitting process were selected in the final model.
cov.j      the covariance matrix computed by the jackknife method (only computed if jack = TRUE).
cov.lw     the covariance matrix computed by the Lin-Wei method (robust covariance)
cov.ls     the covariance matrix computed by the Lin-Sasieni method.
cov.method  the method used to compute the (displayed) covariance matrix and the standard errors. This method is either "jack" if jack = TRUE, or "Lin-Wei" if jack = FALSE.
w.matrix  a matrix with four columns according to the number of uncensored failure times. The first column contains the failure times, the remaining columns (labeled w.raw, w.obskm, and w) contain the raw weights, the weights according to the inverse of the Kaplan-Meier estimates with reverse status indicator and the normalized product of both.
wald     Wald-test statistics.
means      the means of the covariates.
offset.values  offset values.
dataline  the first dataline of the input data set (required for plotfp).
x      the explanatory variables.
y      the response.
alpha     the significance level = 1 - confidence level.
template  the requested template.
formula  the model formula.
betafix  the betafix vector.
call      the function call.
Note

The SAS macro WCM with similar functionality (but without nonlinear estimation using fractional polynomials) is offered for download at http://cemiis.meduniwien.ac.at/en/kb/science-research/software/statistical-software/wcmcoxphw/.

Up to Version 2.13 coxphw used a slightly different syntax (arguments: AHR, AHR.norobust, ARE, PH, normalize, censcorr, prentice, breslow, taroneware). From Version 3.0 on the old syntax is disabled.

Author(s)

Georg Heinze, Meinhard Ploner, Daniela Dunkler

References


See Also

`concord, plotshape, plotw, coxph`

Examples

data("gastric")

# weighted estimation of average hazard ratio
fit1 <- coxphw(Surv(time, status) ~ radiation, data = gastric, template = "AHR")
summary(fit1)
fit1$cov lw  # robust covariance
fit1$cov ls  # Lin-Sasieni covariance


```r
# unweighted estimation, include interaction with years
# ('radiation' must be included in formula!)
gastric$years <- gastric$time / 365.25
fit2 <- coxphw(Surv(years, status) ~ radiation + years : radiation, data = gastric,
               template = "PH")
summary(fit2)

# or equivalently
# coxphw(Surv(years, status) ~ radiation * years, data = gastric, template = "PH")

# select best fp(1) for interaction with time using fp-functionality
# (don't include 'radiation' in formula if using fp(time)!!)
fit3 <- coxphw(Surv(time, status) ~ fp(time) : radiation, data = gastric, template = "PH",
               alpha.fp = c(1, 0.05, 0.05), fp.max = 1)
summary(fit3)
```

---

**fp.power**  
*Provides Fractional Polynomials as Accessible Function*

**Description**

Provides fractional polynomials as accessible function.

**Usage**

```r
fp.power(z, a, b = NULL)
```

**Arguments**

- `z`  
a scalar or vector of positive numerical values.
- `a`  
first power.
- `b`  
optional second power.

**Details**

The function returns fp(a) of z (and optionally fp(b) of z).

**Value**

A matrix with one or two columns (if a second power b was specified), and number of rows equal to the length of z. The columns are sorted by descending power.

**Author(s)**

Georg Heinze
References


See Also

`coxphw`

Examples

```r
fp.power(z = c(1, 4, 6), a = 1)
fp.power(z = c(1, 4, 6), a = 0.5)
fp.power(z = c(1, 4, 6), a = 0.5, b = 0.5)
fp.power(z = c(1, 4, 6), a = 0, b = 2)
```

---

**gastric**

*Gastric Cancer Data*

Description

A data set of survival times of patients with locally advanced, nonresectable gastric carcinoma. The patients were either treated with chemotherapy plus radiation or chemotherapy alone.

Usage

```r
data(gastric)
```

Format

A data frame with 90 observations on the following 4 variables:

- **id**: unique patient id.
- **radiation**: treatment of either 0 = chemotherapy alone or 1 = chemotherapy plus radiation.
- **time**: survival time in days.
- **status**: 0 = censored or 1 = death.

Source


[http://www.mayo.edu/research/documents/gastrichtml/DOC-10027680](http://www.mayo.edu/research/documents/gastrichtml/DOC-10027680)
**plotshape**

**References**


**Examples**

data("gastric")

plot(survfit(Surv(time, status) ~ radiation, data = gastric), lty = 1:2, las = 1,
    xscale = 365.25, xlab = "time (years)", ylab = "survival distribution function")

coxph(Surv(time, status) ~ radiation, data = gastric, template = "AHR")

---

**plotshape**  
*Compute and Plot the Relative or Log Relative Hazard Versus Values of a Continuous Covariable*

**Description**

This function either visualizes a nonlinear or a time-dependent effect of a coxphw object (possibly estimated by the fractional polynomial functionality of coxphw) or obtains the effect estimates at specified values of this variables. It plots/prints the relative or log relative hazard versus values of a continuous covariable.

**Usage**

plotshape(obj, plot.x = NA, ref = NA, plot = TRUE, variable = NULL,
    treatment = NULL, ref.type = "value", addci = TRUE, exp = FALSE,
    xlab = NULL, ylab = NULL, pval = FALSE, digits = 4, ...)

**Arguments**

- **obj** an output object of coxphw, which may include fp()-terms.
- **plot.x** the data values for the continuous variable (e.g., plot.x = 30:70).
- **ref** a reference value. The log relative hazard at this value will be 0. (e.g., ref= 50).
- **plot** if set to TRUE (default), requests a plot, otherwise, a matrix with relative or log relative hazards at values specified in plot.x are printed.
- **variable** name of the variable with the nonlinear effect (use ").
- **treatment** variable which is in interaction with variable (or fp(variable))(use ").
- **ref.type** "value" for a simple nonlinear effect of variable without interaction,  
  "interaction.time" for interaction of treatment with time (or fp(time)),  
  "interaction.treat" for interaction of treatment with fp(variable), or any value to specify the level of the treatment variable for which the fractional polynomial of variable should be plotted.
addci: confidence intervals are obtained. Default is TRUE.
exp: if set to TRUE (default), the log relative hazard is given, otherwise the relative hazard is requested.
xlab: label for x-axis of plot, uses variable specified in variable as default.
ylab: label for y-axis of plot, uses "relative hazard" or "log relative hazard" as default.
pval: if plot = FALSE, add Wald-test p-values to effect estimates at values of plot.x. Default is set to FALSE.
digits: if plot = FALSE, number of printed digits. Default is 4.
... further parameters, to be used for plots (e.g., scaling of axes).

Details

This function can be used to depict the estimated nonlinear or time-dependent effect of an object of class coxphw (possibly using fractional polynomials fp()). It supports simple nonlinear effects as well as interaction effects of continuous variables with binary covariates (see examples section).

Value

A list with the following components:

xbeta: a matrix with estimates of (log) relative hazard.
cilower: lower confidence limits.
ciupper: upper confidence limits.
p: p-value, only if plot = FALSE and pval = TRUE.
alpha: the significance level = 1 - confidence level.
plot.x: the data values of the continuous variable.
exp: an indicator if log relative hazard or relative hazard was obtained.

Author(s)

Georg Heinze, Meinhard Ploner, Daniela Dunkler

References


See Also

coxphw
Examples

```r
# Example 1
set.seed(30091)
n <- 300
x <- 1:n
ture.func <- function(x) 3 * (x / 100)^2 - log(x / 100) - 3 * x / 100
x <- round(rnorm(n = x) * 10 + 40, digits = 0)
time <- rexp(n = n, rate = 1) / exp(ture.func(x))
event <- rep(x = 1, times = n)
futime <- runif(n = n, min = 0, max = 30000)
event <- (time < futime) + 0
time[event == 0] <- futime[event == 0]
my.data <- data.frame(x, time, event)

fitahr <- coxph(Surv(time, event) ~ fp(x), alpha.fp = c(1, 0.05, 0.05), data = my.data,
                   template = "AHR", maxit = 200, print = TRUE)

# estimated function
plot.x <- quantile(x, probs = 0.05):quantile(x, probs = 0.95)
yahr <- plotshape(fitahr, plot.x = plot.x, ref = median(x), variable = "x",
                   ref.type = "value", addci = FALSE, plot = TRUE, type = "l")

# true function
lines(x = plot.x, true.func(plot.x) - true.func(median(plot.x)), lty = 2)

legend("topleft", legend="AHR estimates", "true"), bty = "n", lty = 1:2, inset = 0.05)

# Example 2: select best fp(1) model, without testing in RA2 algorithm
set.seed(512364)
n <- 200
x <- 1:n
ture.func <- function(x) 2.5 * log(x) - 2
x <- round(rnorm(x) * 10 + 40, digits = 0)
x <- round(runif(x) * 60 + 10, digits = 0)
time <- round(100000 * rexp(n = n, rate = 1) / exp(ture.func(x)), digits = 1)
event <- rep(x = 1, times = n)
my.data <- data.frame(x, time, event)

fit <- coxph(Surv(time, event) ~ fp(x), alpha.fp = c(1, 1, 1), data = my.data, fp.max = 1,
              template = "AHR", print = FALSE)

# estimated function
plot.x <- quantile(x, probs = 0.05):quantile(x, probs = 0.95)
y <- plotshape(fit, plot.x = plot.x, ref = 40, ref.type = "value", variable = "x", type = "p")

# true function
lines(x = plot.x, y = true.func(plot.x) - true.func(40), col = "blue")

legend("topleft", legend=c("estimates", "true"), col = c("black", "blue"), bty = "n",
```
inset = 0.1, lty = c(NA, 1), pch=c(1, NA))

# Example 3: interaction of continuous variable with binary "treatment" variable
set.seed(75315)
n <- 200
trt <- rbinom(n = n, size = 1, prob = 0.5)
x <- 1:n
true.func <- function(x) 2.5*log(x)-2
x <- round(rnorm(n = x) * 10 + 40, digits = 0)
x <- round(runif(n = x) * 60 + 10, digits = 0)
time <- 100 * rexp(n = n, rate = 1) / exp(true.func(x)) /
4 * trt - (true.func(x) / 4)^2 * (trt==0)
event <- rep(x = 1, times = n)
my.data <- data.frame(x, trt, time, event)
fit <- coxph(Surv(time, event) ~ fp(x) * trt, data = my.data, alpha.fp=c(1, 1, 0.05),
              template = "AHR", print = FALSE)

# plots the interaction of trt with x (the effect of trt dependent on the values of x)
plot.x <- quantile(x, probs = 0.05):quantile(x, probs = 0.95)
y <- plotshape(fit, variable = "x", treatment = "trt", ref.type = "interaction.treat",
              plot.x = plot.x, main = "interaction of trt with x", type = "l")

# plots the effect of x in subjects with trt = 0
y0 <- plotshape(fit, variable = "x", treatment = "trt", ref.type = 0, plot.x = plot.x,
              type = "l", ref = median(x), main = "effect of x in subjects with trt = 0")

# plots the effect of x in subjects with trt = 1
y1 <- plotshape(fit, variable = "x", treatment = "trt", ref.type = 1, plot.x = plot.x,
              type = "l", ref = median(x), main = "effect of x in subjects with trt = 1")

# Example 4: interaction of binary "treatment" variable with function of time
set.seed(23917)
# time <- 100 * rexp(n = n, rate = 1) / exp((true.func(x) / 10)^2 / 2000 * trt + trt)
# event <- rep(x = 1, times = n)
# my.data <- data.frame(x, trt, time, event)
# plot.x <- seq(from = 1, to = 100, by = 1)
#
# fitahr <- coxph(Surv(time, event) ~ fp(time) * trt + x, alpha.fp = c(1, 1, 0.05),
#               data = my.data, fp.max = 2, template = "AHR")
# yahr <- plotshape(fitahr, variable = "time", treatment = "trt",
#               ref.type = "interaction.time", plot.x = plot.x, plot = FALSE)
#
# fitph <- coxph(Surv(time, event) ~ fp(time) * trt + x, alpha.fp = c(1, 1, 0.05),
#               data = my.data, fp.max = 2, template = "PH")
# yph <- plotshape(fitph, variable = "time", treatment = "trt",
#               ref.type = "interaction.time", plot.x = plot.x, plot = FALSE)
#
# plot(x = plot.x, y = yahr$xbeta, xlab = "time", ylab = "log relative hazard", pch = 1,
**Description**

This function plots the weights used in a weighted Cox regression analysis against time.

**Usage**

```r
plotw(x, rank = FALSE, log = FALSE, legendxy = NULL,...)
```

**Arguments**

- `x` a `coxphw` object.
- `rank` if set to TRUE, plots the weights against ranked time. Default is FALSE.
- `log` if set to TRUE, shows logarithm of weights. Default is FALSE.
- `legendxy` an optional vector of length 2 of the x and y co-ordinates to be used to position the legend.
- `...` additional arguments for plotting

**Details**

The function plots the survival weights, i.e., the left-continuous survivor function estimates, the censoring weights, i.e., estimates of the follow-up distribution obtained by Kaplan-Meier estimation with reversed meaning of the status indicator and the combined normalized weights, i.e. the product of the survival and the censoring weights, rescaled to a mean of 1.

**Value**

No output value.

**Author(s)**

Georg Heinze, Daniela Dunkler

**See Also**

`coxphw`
Examples

data("gastric")

# weighted estimation of average hazard ratio
fit1 <- coxphw(Surv(time, status) ~ radiation, data = gastric, template = "AHR")
plotw(fit1)

# estimation of average regression effect by inverse probability of censoring weights;
# truncate weights at 95th percentile
fit2 <- coxphw(Surv(time, status) ~ radiation, data = gastric, template = "ARE",
               trunc.weights = 0.95)
plotw(fit2)

print.coxphw

Print Method for Objects of Class coxphw

Description

This class of objects is returned by the coxphw function. Objects of this class have methods for the functions summary, print, coef, vcov, and confint.

Usage

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

Arguments

x

object of class coxphw.

... further arguments.

Details

If betafix was requested for one or more regression coefficients, no estimates on inference are given for these coefficients as they were not estimated in the model. The global Wald test only relates to those variables for whom regression coefficients were estimated.

Author(s)

Georg Heinze, Daniela Dunkler

See Also

coxphw
**Pretransformation function**

**Description**

Provides automatic pretransformation of variables (to well-scaled and nonzero values).

**Usage**

\[ PT(z) \]

**Arguments**

\[ z \]

a vector of numerical values.

**Details**

The function transforms a variable by shifting to positive values, and dividing by scaling factor (a power of 10) such that the standard deviation is approximately equal to 1.

**Value**

\[ (z + \text{shift}) / \text{scale} \]

**Author(s)**

Georg Heinze

**See Also**

coxphw

**Examples**

\[ PT(z = c(-6, -1, 4, 6)) \]
summary.coxphw

Summary Method for Objects of Class coxphw

Description
This class of objects is returned by the coxphw function. Objects of this class have methods for the functions summary, print, coef, vcov, and confint.

Usage

```r
## S3 method for class 'coxphw'
summary(object, print = TRUE, ...)
```

Arguments

- `object` object of class coxphw.
- `print` print summary. Default is TRUE.
- `...` further arguments.

Details

If `betafix` was requested for one or more regression coefficients, no estimates on inference are given for these coefficients as they were not estimated in the model. The global Wald test only relates to those variables for whom regression coefficients were estimated.

Author(s)
Georg Heinze, Daniela Dunkler

See Also
coxphw

vcov.coxphw

Obtain the Variance-Covariance Matrix for a Fitted Model Object of Class coxphw

Description

This class of objects is returned by the coxphw function. Objects of this class have methods for the functions summary, print, coef, vcov, and confint.

Usage

```r
## S3 method for class 'coxphw'
vcov(object, ...)
```
Arguments

object object of class coxphw.

... further arguments.

Author(s)

Daniela Dunkler

See Also

coxphw

Description

Obtain Wald chi-squared test statistics and p-values for one or more regression coefficients given their variance-covariance matrix.

Usage

wald(coeff, cov, index = NULL, h0 = NULL)

Arguments

coeff a vector with regression coefficients.

cov a variance-covariance matrix.

index an integer specifying which parameters should be jointly tested. Default is to test all parameters given in coeff and cov.

h0 a vector with the same length as coeff stating the null hypothesis for the test. Default is 0 for all coefficients.

Details

The test is based on the assumption that the coefficients asymptotically follow a (multivariate) normal distribution with mean coeff and a variance-covariance matrix cov.

Value

A vector with the following components:

chi2 the Wald-test statistic.

df degrees of freedom.

p p-value.
Author(s)
   Daniela Dunkler

See Also
   coxphw
Index

*Topic datasets
  biofeedback, 4
gastric, 14
*Topic math
  fp.power, 13
  PT, 21
*Topic package
  coxphw-package, 2
*Topic regression
  coxphw, 7
  coxphw-package, 2
*Topic survival
  concord, 5
  coxphw, 7
  coxphw-package, 2
  fp.power, 13
  plotshape, 15
  plotw, 19
  wald, 2, 3, 23
*Topic utilities
  coef.coxphw, 5
  confint.coxphw, 7
  print.coxphw, 20
  summary.coxphw, 22
  vcov.coxphw, 22

biofeedback, 4
cluster, 10
coef.coxphw, 5
concord, 2, 3, 5, 12
confint.coxphw, 7
coxph, 10, 12
coxphw, 2, 3, 5–7, 7, 14, 16, 19–24
coxphw-package, 2
fp.power, 2, 3, 13
gastric, 14
mfp, 10

plotshape, 2, 3, 10, 12, 15
plotw, 2, 3, 12, 19
print.coxphw, 20
PT, 21
summary.coxphw, 22
Surv, 8
vcov.coxphw, 22
wald, 2, 3, 23