Package ‘rankhazard’

April 29, 2025

Version 1.1.1

Date 2025-04-28

Title Rank-Hazard Plots

Imports survival

Suggests rms

Description Rank-hazard plots Karvanen and Harrell (2009) <DOI:10.1002/sim.3591> visualize the relative importance of covariates in a proportional hazards model. The key idea is to rank the covariate values and plot the relative hazard as a function of ranks scaled to interval [0,1]. The relative hazard is plotted in respect to the reference hazard, which can bee.g. the hazard related to the median of the covariate.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2025-04-29 06:40:02 UTC

Author Juha Karvanen [aut],
Nanni Ultima [aut, cre]

Maintainer Nanni Ultima <nanni.ultima@gmail.com>

Contents

rankhazard-package *Rank-hazard plots*

Description

Rank-hazard plots visualize the relative importance of covariates in a proportional hazards model. The key idea is to rank the covariate values and plot the relative hazard as a function of ranks scaled to interval [0,1]. The relative hazard is the hazard plotted in respect to the reference hazard, which can be e.g. the hazard related to the median of the covariate. Transformation to scaled ranks allows plotting of covariates measured in different units in the same graph, which helps in the interpretation of the epidemiological relevance of the covariates. Rank-hazard plots show the difference of hazards between the extremes of the covariate values present in the data and can be used as a tool to check if the proportional hazards assumption leads to reasonable estimates for individuals with extreme covariate values. Alternative covariate definitions or different transformations applied to covariates can be also compared using rank-hazard plots.

Details

Package:	rankhazard
Type:	Package
Version:	1.1.1
Date:	2025-04-28
License:	GPL 2
LazyLoad:	yes

The function `rankhazardplot` receives a `coxph` (package `survival`) object or a `cph` (package `rms`) object as an argument and creates a rank-hazard plot of the covariates. The reference points for the relative hazards and legend texts can be provided as optional arguments. Plotting parameters such as, lwd, lty, col and pch are passed to the plotting commands.

Author(s)

Juha Karvanen <juha.karvanen@iki.fi> , Nanni Ultima <nanni.ultima@gmail.com>

References

Karvanen J., Harrell F. E., Jr. 2009 Visualizing covariates in proportional hazards model. *Statistics in Medicine*, **28**, 1957–1966.

Examples

```
library(survival)
model <- coxph(Surv(time,status) ~ age + sex + ph.karno, data = cancer, x=TRUE)
rankhazardplot(model, data=cancer)
library(rms)
model2 <- cph(Surv(time,status) ~ age + sex + ph.karno, data = cancer, x=TRUE)
```

```
dd <- datadist(cancer)
options(datadist="dd")
rankhazardplot(model2, data=cancer)
```

rankhazardplot *Create a Rank-hazard plot*

Description

Creates a rank-hazard plot. Plots the relative hazards (or the logarithm of the relative hazards) for each covariate of a Cox proportional hazards model fitted by coxph or cph.

Usage

```
rankhazardplot(...)

## S3 method for class 'coxph'
rankhazardplot(coxphobj, data, select = NULL, refpoints = NULL,
   CI_level = 0.95, x_CI = NULL, draw.confint = FALSE,
   legendtext = NULL, axistext = NULL, legendlocation = "top",
   axistextposition = -0.1, reftick = TRUE, refline = FALSE,
   col.refline = 1, lwd.refline = 1, lty.refline = 2,
   ylab = NULL, ylim = NULL, yticks = NULL, yvalues = NULL,
   xtext =TRUE, plottype = "hazard", axes = TRUE, na.rm = TRUE,
   draw = TRUE, return = FALSE, col = NULL, lwd = 1, lty = 1,
   pch = NULL, cex = 1, bg = "transparent", pt.lwd = 1,
   col.CI = col, lty.CI = lty +1, lwd.CI = lwd, add = FALSE,
   graphsbefore = 0, args.legend = NULL, ...)

## S3 method for class 'cph'
rankhazardplot(cphobj, data, select = NULL, refpoints = NULL,
   CI_level = 0.95, x_CI = NULL, draw.confint = FALSE,
   legendtext = NULL, axistext = NULL, legendlocation = "top",
   axistextposition = -0.1, reftick = TRUE, refline = FALSE,
   col.refline = 1, lwd.refline = 1, lty.refline = 2,
   ylab = NULL, ylim = NULL, yticks = NULL, yvalues = NULL,
   xtext =TRUE, plottype = "hazard", axes = TRUE, na.rm = TRUE,
   draw = TRUE, return = FALSE, col = NULL, lwd = 1, lty = 1,
   pch = NULL, cex = 1, bg = "transparent", pt.lwd = 1,
   col.CI = col, lty.CI = lty +1, lwd.CI = lwd, add = FALSE,
   graphsbefore = 0, args.legend = NULL, ...)

## Default S3 method:
rankhazardplot(x, coefs = NULL, xp = NULL, refvalues = NULL,
    refpoints = NULL, confinterval = NULL, select = 1,
    legendtext = NULL, axistext = NULL, legendlocation = "top",
    axistextposition = -0.1, reftick = TRUE, refline = FALSE,
    col.refline = 1, lwd.refline = 1, lty.refline = 2,
    ylab = NULL, ylim = NULL, yticks = NULL, yvalues = NULL,
    xtext =TRUE,  plottype = "hazard",axes = TRUE, na.rm = TRUE,
    col = NULL, lwd = 1, lty = 1, pch = NULL, cex = 1,
    bg = "transparent", pt.lwd = 1, draw.confint = NULL,
    col.CI = col, lty.CI = lty +1, lwd.CI = lwd, add = FALSE,
    graphsbefore = 0, args.legend = NULL, ...)
```

Arguments

`coxphobj` An object of class coxph created by function coxph from the package `survival`. The object should have been created with `x = TRUE`. Otherwise `x_CI` must be given.

`cphobj` An object of class `cph` created by function `cph` from the package `rms`. The object should have been created with the option `x = TRUE`. Otherwise `x_CI` must be given.

`data` A data frame that contains the covariates in the model. It can be the data used in fitting the Cox proportional hazards model or new data. It can contain more covariates than there are in the model. Used with the argument `coxphobj` or `cphobj`.

`select` A vector with the order numbers of the covariates to be plotted. The order numbers are defined by the order of the covariates in the model. It can be used with the argument `coxphobj`, `cphobj` or `confinterval`. When used with `coxphobj` or `cphobj` the default is `NULL` and all covariates in the model are plotted. With `confinterval` the default is the first covariate.

`x` A data frame that contains the covariates to be plotted with one column for each covariate. Used with the argument `xp` or `coefs`. When used with `xp`, `x` defines the values to be printed on the x-axis. When used with `coefs`, `x` the predictions are computed as a product of `coefs` and `x`. The dimensions of `x` and `xp` must be the same. The number of covariates given in `x` and the length of the `coefs` must be the same. If `return = TRUE` `x` is returned in the list. See section `Value`.

`xp` A data frame that contains the predictions ("terms") for the covariates to be plotted. The order of the covariates must be the same as in `x`. If `return = TRUE` `xp` is returned in the list. See section `Value`.

`coefs` A vector of Cox regression coefficients for the covariates to be plotted. The order of the covariates must be the same as in `x`.

`refpoints` A vector of reference points given in the same order as the covariates in the model. A reference point is the value of the covariate where the reference hazard is calculated to compute the relative hazards as a quotient of the hazard and the reference hazard. Consequently, at the reference point the value of the relative hazard is 1 (and the value of the logarithm of the relative hazard is 0). If the `select` argument is in use the reference points are given for the selected covariates only and in the same order as the selection made. If `NULL`, the medians of each covariate are used as reference points. With factors the default is the reference level of the factor. If the reference point for the selected covariate is `NA`, the default is used. When plotting the confidence intervals the reference point for factors can be changed only by using a model that has been fitted with a data that has a re-leveled factor.

`refvalues` A vector of reference values. A reference value of a covariate is the predicted value ("terms") at the reference point (see `refpoints`). Used and needed only with the argument `xp`. Otherwise calculated by the `rankhazardplot` function with the values of the `refpoints`. If `return = TRUE` `refvalues` is returned in the list. See section `Value`.

`confinterval` A list that contains all information to plot confidence intervals. If given to the `rankhazardplot.default` function the confidence intervals are always plotted. If `return = TRUE` `confinterval` is returned in the list. See section `Value`.

`draw.confint` If `TRUE` the confidence intervals are plotted. Can be used with `coxphobj` or `cphobj`. By default `FALSE`.

`CI_level` A number between 0 and 1 that defines the level of the confidence interval for (the logarithm of) the relative hazard. By default 0.95.

`x_CI` A data frame of the covariate data. Needed if the `coxphobj$x` or `cphobj$x` does not exist. The number and the order of the columns must be same as `as.data.frame(coxphobj$x)` or `as.data.frame(cphobj$x)`.

`col.CI` Defines the colour of the graph of confidence intervals. By default the same as colour of the relative hazard. See documentation for `par`.

`lty.CI` Defines the type of the graph of confidence intervals. By default `lty + 1`. See documentation for `par`.

`lwd.CI` Defines the width of the graph of confidence intervals. By default the same as width of the relative hazard. See documentation for `par`.

`legendtext` A vector of covariate names for the legend box and the summary. If `NULL` and `axistext = NULL`, the names are from the columns of `xp`, `coefs`, `attr(coxphobj$terms,"term.labels'` or `attr(cphobj$terms, "term.labels")`. If `NULL`, `axistext` is used if it is given. If the `select` argument is in use the names are given for the selected covariates only and in the same order as the selection made.

`legendlocation` A keyword that determines the location where the legend box is printed. By default "top". See `Details` in the documentation for `legend`. If `NULL`, the legend box is not printed and it can be added by function `legend`.

`args.legend` A list of additional arguments to be passed to `legend`. Names of the list are used as argument names. These values are used when the legend is drawn. For example if `legend` and `x` are specified in this list, those will replace the values of `legendtext` and `legendlocation` respectively.

`axistext` A vector of covariate names and units for the x-axis. If `NULL` and `legendtext = NULL`, the names are from the columns of `x` or the same as in the `data`. If `NULL`, `legendtext` is used if it is given. If the `select` argument is in use the names are given for the selected covariates only and in the same order as the selection made.

`axistextposition`

A number that defines the x-coordinate where the axis texts are placed. Adjustment is right alignment. By default -0.1. The bigger the size of the plot is, the closer to zero the value can be. The maximum is 0.

`xtext` By default `TRUE` and the labels and the quartiles of the covariates are printed on the x-axis. If the text on the x-axis should be altered in some way (e.g. `font` or `cex`) the printing of the labels must be prevented by giving `xtext = FALSE`. After that the labels can be added by `mtext`

`plottype` A string that defines the scale for the y-axis. Either `"hazard"` for the relative hazard with log-scale or `"loghazard"` for the logarithm of the relative hazard with linear scale.

`ylab` A string that defines the label of the y-axis. When `plottype = "hazard"` the default is "relative hazard". When `plottype = "loghazard"` the default is the "logarithm of the relative hazard". See documentation for `plot`.

`ylim` A numerical vector that defines the range of the y-axis. By default the range is the minimum and the maximum of (the logarithm of) the relative hazards. See documentation for `plot.window`.

`yticks` A vector that determines the places for the ticks that are plotted to the y-axis. When `plottype = "hazard"` it is recommended that evenly calculated ticks are used before and after the reference e.g. `c(seq(0.5, 1, by=0.1),2:7)`, due to the logarithmic scale of the y-axis. If `NULL` the ticks are computed using `pretty`.

`yvalues` A vector that determines which values are printed on the y-axis. If `NULL` the values of the `yticks` are used. It is recommended that 1 is in the `yvalues` if the `plottype = "hazard"` and 0 if the `plottype = "loghazard"`. In addition to this the `yvalues` should be a subset of `yticks`.

`axes` By default `TRUE` and the axes are plotted. If `FALSE` axes are not plotted. NOTE when adding the axes yourself: On the rank-hazard plot the x-axis is from 0 to 1 and the places for quartiles are 0.25, 0.5 and 0.75.

`na.rm` By default `TRUE` and only complete cases are plotted. Complete cases are cases that have information on all covariates that are used in fitting the model. If `FALSE` all available cases for each variable are plotted.

`reftick` By default `TRUE` and the reference tick is emboldened. For the relative hazard the tick is at 1 and for the logarithm of the relative hazard at 0.

`refline` If `TRUE` the reference line is drawn. The line is horizontal at the same place as the reference tick (see `reftick`). By default `FALSE`.

`col.refline` Defines the colour of the reference line, if `refline = TRUE`. By default 1. See documentation for `par`.

`lwd.refline` Defines the width of the reference line, if `refline = TRUE`. By default 1. See documentation for `par`.

`lty.refline` Defines the type of the reference line, if `refline = TRUE`. By default 2. See documentation for `par`.

`col` A vector that defines the colours of the lines and the points. If the vector is shorter than the number of the covariates to be plotted, the values are repeated. See documentation for `par`.

`lwd` A vector that defines the widths of the lines. If the vector is shorter than the number of the covariates to be plotted, the values are repeated. See documentation for `par`.

`lty`
A vector that defines the types of the lines. If the vector is shorter than the number of the covariates to be plotted, the values are repeated. See documentation for `par`.

`pch`
A vector that defines the characters of the points. If the vector is shorter than the number of the covariates to be plotted, the values are repeated. If the value is `NA` the points are not plotted. See documentation for `points`.

`bg`
A vector that defines the fill colour of the point. Available only for point characters `21:25`. If the vector is shorter than the number of the covariates to be plotted, the values are repeated. By default `"transparent"`. See documentation for `points`.

`pt.lwd`
A vector that defines the line width for the drawing symbols. If the vector is shorter than the number of the covariates to be plotted, the values are repeated. See documentation for `lwd` in `points`.

`cex`
A vector that defines the size of the points. If the vector is shorter than the number of covariates to be plotted, the values are repeated. See documentation for `plot.default`.

`draw`
By default `TRUE` and a rank-hazard plot and a summary are printed. If `FALSE`, no output is provided unless `return = TRUE`.

`return`
By default `FALSE` and `Value` is not returned. Used with the argument `coxphobj` or `cphobj`. If `TRUE`, `x`, `xp`, `refvalues` and `confinterval` are returned as a list. See section `Value`.

`add`
By default `FALSE` and a new plot is created. If `TRUE` graphs are added to the previous plot. In that case the amount of graphs already plotted must be specified by the argument `graphsbefore`.

`graphsbefore`
A number that specifies how many graphs have already been drawn on the plot. If `add = FALSE`, the default is 0.

`...`
Other arguments to be passed to `matplot` and `matlines` commands. For example `main`. Arguments that are already in use are `x`, `y`, `type`, `log`, `xlim`, `xlab`, `xaxt` and `yaxt` and using them will result in an error.

Details

The function `rankhazardplot` receives a `coxph` (package `survival`) object or a `cph` (package `rms`) object as an argument and creates a rank-hazard plot of the covariates. The reference points for the relative hazards and legend texts can be provided as optional arguments. Plotting parameters such as, lwd, lty, col and pch are passed to the plotting commands.

Rank-hazard plots visualize the relative importance of covariates in a proportional hazards model. The key idea is to rank the covariate values and plot the relative hazard as a function of ranks scaled to interval [0,1]. The relative hazard is the hazard plotted in respect to the reference hazard, which can be e.g. the hazard related to the median of the covariate.

The labels on the x-axis show the minimum, the quartiles and the maximum of each covariate. These are real observations. If the quantile would be determined by a mean of two observations, the smaller value is choosed to be the quantile. However, if the number of the obervations is even, the default reference point is not necessarily a real observation as it is calculated as a mean of two middle observations. Hence, the median shown on the x-axis and the reference point can differ even when the default is used.

Predictions are computed by `predict.coxph`, when the function is called with the argument `coxphobj` and by `predict.rms`, when the function is called with the argument `cphobj`. Consequently, relative hazards are available for models that are supported by `predict.coxph` or `predict.rms`. For example the `pspline` transforms are supported by `predict.coxph` but not by `predict.rms`.

The upper and lower confidence limits of the Cox regression coefficients are used to calculate the confidence intervals for the relative hazards. Confidence intervals are only supported for original covariates (same as in the data), simply transformed covariates (e.g. `log`) and factors.

Rank-hazard plots can be used to visualize time-dependent models. In that case plotting can be made using `coefs` that are Cox regression coefficients of the time-dependent model. The data matrix `x` contains the values of covariates at some specific time. It is also possible to make a rank-hazard plot by the argument `xp`. In that case both `x` and `xp` must be selected so that they contain information only at some specific time. Third way to make the rank-hazard plot is to use the time-dependent model and give the values of the covariates at some specific time as the argument `data`. If the purpose is to compare relative hazards at different times, it is recommended that the same reference points are used in every plot. For example the medians of every covariate at the first measurement.

Value

If `return = TRUE` the function returns a list that contains all the information needed to draw a rank-hazard plot and confidence intervals. The list contains:

`x` A data frame that contains the covariate data.

`xp` A data frame that contains the centered predictions for all covariates in the model. Calculated by `predict.coxph` or `predict.rms`.

`refvalues` A vector that contains the centered predictions that are calculated using the reference point defaults for the covariates that don't have a given reference point. Calculated by `predict.coxph` or `predict.rms`.

`confinterval` A list that contains covariate data as a data frame `x`, predictions by terms as a data frame `xp`, reference values as a vector `refvalues`, a lower confidence interval for the predictions as a data frame `low`, a lower confidence interval for the reference values as a vector `lowrefvalues`, an upper confidence interval for the predictions as a data frame `upp` and an upper confidence interval for the reference values as a vector `upprefvalues`. In the list there is also the vector `select_CI` that has indices of the covariates in the model for which the confidence intervals are provided.

The covariates for which the confidence intervals are provided are original (same as in the data), simply transformed (e.g. `log`) and factors. The predictions are calculated as a product of `coefs` and `x`. The upper and lower confidence limits of the Cox regression coefficients are used to calculate the confidence intervals for the relative hazards. NB: values aren't centered and for that reason e.g. `xp` and `confinterval$xp` are not the same.

Author(s)

Juha Karvanen <juha.karvanen@iki.fi>, Nanni Ultima <nanni.ultima@gmail.com>

References

Karvanen J., Harrell F. E., Jr. 2009 Visualizing covariates in proportional hazards model. *Statistics in Medicine*, **28**, 1957–1966.

See Also

coxph, cph,predict.coxph, predict.rms

Examples

```
library(survival)
library(rms)

data(pbc)
# new status variable
pbc$statusbin <- ifelse(pbc$status==0, 0, NA)
pbc$statusbin <- ifelse(pbc$status==2, 1, pbc$statusbin)
pbc$edema2 <- as.factor(pbc$edema)

### different ways to make a rank-hazard plot ###

coxmodel1 <- coxph(Surv(time, statusbin) ~ age + protime +
                  as.factor(edema), data = pbc, x = TRUE)

par(mar = c(4, 5, 4, 2) + 0.1)
rankhazardplot(coxmodel1, data = pbc,
               main = "Rank-hazardplot by coxphobj")

dd <- datadist(pbc)
options(datadist = 'dd')
cmodel1 <- cph(Surv(time, statusbin) ~ age + protime + edema2,
               data = pbc, x = TRUE)
rankhazardplot(cmodel1, data = pbc,
               main = "Rank-hazardplot by cphobj")

output1 <- rankhazardplot(coxmodel1, data = pbc, draw = FALSE,
                          return = TRUE)

rankhazardplot(x = output1$x, xp = output1$xp,
               refvalues = output1$refvalues,
               main = "Rank-hazardplot by x and xp")
rankhazardplot(x = output1$x[1:2], coefs = coxmodel1$coef[1:2],
               main = "Rank-hazardplot by x and coefs \n with returned data")
rankhazardplot(x = pbc[c("age","protime")], coefs = coxmodel1$coef[1:2],
               main = "Rank-hazardplot by x and coefs \n with original data")

rankhazardplot(x = output1$x, xp = output1$xp, refvalues = output1$refvalues,
               main = "Rank-hazardplot by x and xp")
rankhazardplot(x = output1$x[1:2], coefs = coxmodel1$coef[1:2],
               main = "Rank-hazardplot by x and coefs \n with returned data")
rankhazardplot(x = pbc[c("age","protime")], coefs = coxmodel1$coef[1:2],
               main = "Rank-hazardplot by x and coefs \n with original data")
```

```
### changing legend ###

rankhazardplot(coxmodel1, data = pbc,
               main = "Rank-hazardplot by coxphobj",
               legendtext = c("age", "protime", "edema"),
               legendlocation = "topleft")

rankhazardplot(coxmodel1, data = pbc,
               main = "Rank-hazardplot by coxphobj",
               legendtext = c("age", "protime", "edema"),
               legendlocation = "topleft",
               args.legend = list(cex = 0.8, bty = "o"))

### selecting covariates ###

coxmodel2 <- coxph(Surv(time, statusbin) ~ age + protime +
                     as.factor(edema) + bili + albumin + copper +
                     ast + as.factor(stage), data = pbc, x = TRUE)

par(mar = c(9, 5, 4, 2) + 0.1)
rankhazardplot(coxmodel2, data = pbc, args.legend= list(cex = 0.5),
               main = "Too much information?")

par(mar = c(4, 5, 4, 2) + 0.1)
rankhazardplot(coxmodel2, data = pbc, select = c(1, 5, 4),
               main = "How to select covariates")

### highlighting the reference hazard ###

par(mfrow = c(1, 2))
rankhazardplot(coxmodel2, select = c(1, 5, 4), data = pbc,
               refline = TRUE, plottype = "hazard",
               lty.refline = 3, main = "Reference line at 1")
rankhazardplot(coxmodel2, select = c(1, 5, 4), data = pbc,
               refline = TRUE, plottype = "loghazard",
               col.refline = "darkblue",
               main = "Reference line at 0")

### using graphical parameters ###

# Compare the two following plots
rankhazardplot(coxmodel2, select = c(1, 3, 4, 5), data = pbc,
               main = "By default")
par(mar = c(5, 5, 4, 2) + 0.1)
rankhazardplot(coxmodel2, select = c(1, 3, 4, 5), data = pbc,
               ylim = c(0.4, 10), ylab = "Relative hazard",
               yvalues = c(0.4, 1, 2, 4, 6, 10),
               yticks = c(seq(0.4, 1, by = 0.1), 2:10),
               col = c("darkgreen","navyblue","maroon3",1),
               pch = 18:21, lwd = 2, lty = c(1, 1, 2, 2),
               cex = 0.9, bg = "yellow", pt.lwd = 2,
               main = "Graphical arguments in use",
               axistext = c("age", "stage", "bilirubin", "albumin"),
               args.legend = list(cex = 0.8,
                 legend = c("age", "factor(stage)", "bilirubin", "albumin")))

### comparing covariates from different models ###

# transforms #
# same model as coxmodel2, only bilirubin is transformed by logarithm
logmodel <- coxph(Surv(time, statusbin) ~ age + protime +
                    as.factor(edema) + log(bili) + albumin +
                    copper + ast + as.factor(stage), data = pbc, x = TRUE)
# same model as coxmodel2, only a pspline is fitted to bilirubin
coxspline <- coxph(Surv(time, statusbin) ~ age + protime +
                     as.factor(edema) + pspline(bili) + albumin +
                     copper + ast + as.factor(stage), data = pbc, x = TRUE)

par(mar = c(3, 5, 4, 2) + 0.1, mfrow = c(1, 1))

rankhazardplot(coxmodel2, data = pbc, select = 4,
               legendtext = c( "bili","log(bili)", "pspline(bili)"),
               args.legend = list(col = 1:3, pch = 0:2),
               ylim = c(0.336, 10.7), cex.main = 0.8,
               main = "Transforming has a great impact on
               interpreting the effect of the bilirubin")
# For next graph we want the same data that has been used
# in fitting the model. We also want log-transformed values on the x-axis.
outputlog <- rankhazardplot(logmodel, data = pbc,
                            return = TRUE, draw = FALSE)
rankhazardplot(xp = outputlog$xp["log(bili)"], select = 4,
               x = log(outputlog$x["bili"]), add = TRUE,
               refvalues = outputlog$ref["log(bili)"],
               graphsbefore = 1, axistext = "log(bili)")
rankhazardplot(coxspline, data = pbc, select = 4, add = TRUE,
               graphsbefore = 2)

# comparing models that differ by one covariate #
# same model as coxmodel2, only age is left out
# how does it affect relative hazards for stage?
coxmodel3 <- coxph(Surv(time, statusbin) ~ protime +
                     as.factor(edema) + log(bili) + albumin +
                     copper + ast + as.factor(stage), data = pbc,
                   x = TRUE)

par(mar = c(2, 5, 4, 2) + 0.1)
rankhazardplot(coxmodel2, data = pbc, select = 8,
               args.legend = list(x = "bottomright", col = 1:2,
                legend = c("age in the model","age not in the model"),
                pch = 0:2, cex = 0.7),
               axistext = "stage", cex.main = 0.8,
               main = "Comparing the relative risk of 'stage' between
               similar models that differ by covariate 'age'")
rankhazardplot(coxmodel3, data = pbc, select = 7, add = TRUE,
               graphsbefore = 1)
```

```
### changing reference points ###

# factors with non-numerical levels #
coxmodel4 <- coxph(Surv(time, statusbin) ~ age + sex,
                   data = pbc, x = TRUE)
par(mfrow = c(1, 2))
rankhazardplot(coxmodel4, data = pbc,
               main = "Reference points by default",
               refline = TRUE, ylim = c(0.4, 3.8))
rankhazardplot(coxmodel4, refpoints = c(40, "f"), data = pbc,
               main = "Different reference points",
               refline = TRUE, ylim = c(0.4, 3.8))

# with select argument and changing only part of reference points #
par(mfrow = c(1, 1))
rankhazardplot(coxmodel2, data = pbc, select = c(7, 1),
               refpoints = c(100, NA), ylim = c(0.5, 4.3),
               refline = TRUE,
               main = "Reference point for age by default")

# using coefs argument #
coefs <- c(coxmodel2$coef["bili"],logmodel$coef["log(bili)"])
xlog <- data.frame(outputlog$x["bili"], log(outputlog$x["bili"]))
par(mar = c(3, 5, 4, 2) + 0.1)
rankhazardplot(x = xlog, coefs = coefs, refpoints = c(1.2, log(1.2)),
               legendtext = c("bili", "log(bili)"),
               main = c("Reference point is 1.2, the maximum",
                        "\n value for 'normal' total bilirubin"))

# using add argument #
# this way the values on x-axis are always bili, not log(bili)
par(mar = c(2, 5, 4, 2) + 0.1)
rankhazardplot(coxmodel2, data = pbc, select = 4, refpoints = 1.2,
               ylim = c(0.375, 9.64),
               args.legend = list(col = 1:2, pch = 0:1),
               legendtext = c("bili", "log(bili)"))
rankhazardplot(logmodel, data = pbc, select = 4, refpoints = 1.2,
               add = TRUE, graphsbefore = 1)

# using xp argument #
output1_new_ref <- rankhazardplot(coxmodel1, data = pbc,
                                  refpoints = c(40, 10,0),
                                  draw = FALSE, return = TRUE)
rankhazardplot(x = output1_new_ref$x, xp = output1_new_ref$xp,
               refvalues = output1_new_ref$refvalues,
               main = "How to change the reference \n points when using xp")

### confidence intervals ###

par(mar = c(2, 5, 4, 2) + 0.1)
rankhazardplot(confinterval = output1$conf,
               main = c("By argument confinterval,",
                        "\n 95 per cent confidence intervals"))

rankhazardplot(coxmodel1, data = pbc, draw.confint = TRUE,
               select = 1, col.CI = "red", lwd = 2, lwd.CI = 1,
               main = "By argument confint and \n using graphical arguments")

rankhazardplot(coxmodel1, data = pbc, draw.confint = TRUE,
               select = 1, refpoints = 40,
               main = "By argument confint and \n changing reference point")

rankhazardplot(coxmodel2, data = pbc, draw.confint = TRUE,
               select = 5, col.CI = 2, lty.CI = 3, cex = 0.7,
               main = "99 per cent confidence intervals",
               CI_level = 0.99)

rankhazardplot(coxmodel2, data = pbc, draw.confint = TRUE,
               select = 5, col.CI = 2, lty.CI = 3, cex = 0.7,
               main = "95 per cent confidence intervals",
               ylim = c(0.208, 10.1))

### data in start-stop format ###

data(cgd)
timemodel <- coxph(Surv(tstart, tstop, status) ~ treat + height +
                     steroids, data = cgd, x = TRUE)
# steroids and height are in the model only to make
# the example plot more interesting
rankhazardplot(timemodel, data = cgd[cgd$enum == 1,],
               main = "Covariate values at study entry")
```

Index