Package 'QHScrnomo'

March 18, 2024

Title Construct Nomograms for Competing Risks Regression Models

Version 3.0.1

Description Nomograms are constructed to predict the cumulative incidence rate which is calculated after adjusting for competing causes to the event of interest. K-fold cross-validation is implemented to validate predictive accuracy using a competing-risk version of the concordance index. Methods are as described in: Kattan MW, Heller G, Brennan MF (2003).

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports cmprsk, Hmisc

Depends R (>= 3.5.0), rms

LazyData true

URL https://github.com/ClevelandClinicQHS/QHScrnomo

BugReports https://github.com/ClevelandClinicQHS/QHScrnomo/issues

VignetteBuilder knitr

NeedsCompilation yes

Author Michael Kattan [aut], Changhong Yu [aut], Xinge Ji [aut], Zajichek Alex [cre]

Maintainer Zajichek Alex <zajicha2@ccf.org>

Repository CRAN

Date/Publication 2024-03-18 21:20:02 UTC

R topics documented:

`anova.cmprsk` *ANOVA Table for a Competing Risks Regression Model*

Description

Uses the `anova.rms` method to construct an analysis of variance table for the competing risks regression model fit from `crr.fit`.

Usage

```
## S3 method for class 'cmprsk'
anova(object,...)
```

Arguments

`object`	A model fit by `crr.fit`
`...`	Not used

Value

An `anova.rms` matrix

Note

This function requires that the `rms` package is attached

Author(s)

Changhong Yu. Department of Quantitative Health Sciences, Cleveland Clinic

See Also

crr.fit anova.rms

Examples

```
dd <- datadist(prostate.dat)
options(datadist = "dd")
prostate.f <- cph(Surv(TIME_EVENT,EVENT_DOD == 1) ~ TX  + rcs(PSA,3) +
           BX_GLSN_CAT +  CLIN_STG + rcs(AGE,3) +
           RACE_AA, data = prostate.dat,
           x = TRUE, y = TRUE, surv = TRUE,time.inc = 144)
prostate.crr <- crr.fit(prostate.f, cencode = 0, failcode = 1)
anova(prostate.crr)
```

---

cindex *Concordance Index Calculation (C-Index)*

Description

Computes the concordance index for a predictor as a discrimination metric for binary, time-to-event, and competing risks outcomes.

Usage

```
cindex(
  prob,
  fstatus,
  ftime,
  type = "crr",
  failcode = 1,
  cencode = 0,
  tol = 1e-20
)
```

Arguments

prob	A risk score (typically a probability giving the risk of event failure)
fstatus	The event status
ftime	The event times. Applies when the type argument is "survival" or "crr"
type	The outcome type: "logistic" for binary, "survival" for ordinary time-to-event, and "crr" for competing risks outcomes. Defaults to "crr".
failcode	The value of fstatus that indicates the event of interest. Defaults to 1.
cencode	The censoring event code. Defaults to 0.
tol	Error tolerance (not used)

Value

A named vector with following elements:

N	Total number of observations in the input data
n	Number of observations used for calculation
usable	Total number of usable pairs
concordant	Number of concordant pairs
cindex	The concordance index: number of concordant pairs divided by the total number of usable pairs

Author(s)

Changhong Yu, Michael Kattan, Brian Wells, Amy Nowacki.

Examples

```
dd <- datadist(prostate.dat)
options(datadist = "dd")
prostate.f <- cph(Surv(TIME_EVENT,EVENT_DOD == 1) ~ TX  + rcs(PSA,3) +
           BX_GLSN_CAT +  CLIN_STG + rcs(AGE,3) +
           RACE_AA, data = prostate.dat,
           x = TRUE, y = TRUE, surv = TRUE,time.inc = 144)
prostate.crr <- crr.fit(prostate.f, cencode = 0, failcode = 1)

# Cross-validated predictions
prostate.dat$preds.cv.prostate.crr.120 <- tenf.crr(prostate.crr, time = 120, fold = 2)

## calculate the competing-risks version of concordance index
with(prostate.dat, cindex(preds.cv.prostate.crr.120,
                          ftime = TIME_EVENT,
                          fstatus =EVENT_DOD, type = "crr"))["cindex"]
```

---

crr.fit *Fit A Competing Risks Regression Model*

Description

Fits a competing risks regression model using the crr function from an existing cph object which can then be used to construct a nomogram.

Usage

```
crr.fit(fit, cencode = 0, failcode = 1)
```

Arguments

fit	A Cox proportional hazards regression model constructed from cph (by Frank Harrell)
cencode	The value of the status column that indicates a censored observation
failcode	The value of the status column that indicates the event of interest

Value

Returns a list of class cmprsk, with components:

coef	the estimated regression coefficients
loglik	log pseudo-liklihood evaluated at coef
lscore	derivitives of the log pseudo-likelihood evaluated at coef
inf	-second derivatives of the log pseudo-likelihood
var	estimated variance covariance matrix of coef
res	matrix of residuals giving the contribution to each score (columns) at each unique failure time (rows)
uftime	vector of unique failure times
bfitj	jumps in the Breslow-type estimate of the underlying sub-distribution cumulative hazard (used by predict.crr())
tfs	the tfs matrix (output of tf(), if used)
converged	TRUE if the iterative algorithm converged.
cencode	the value of the status indicator that indicates a censored observation
failcode	the value of the status indicator that indicates an event of interest
cph.f	regular survival model fitted by cph which is saved for function nomogram.crr to adjust lp for competing risks
cphdat	data used for cph model, where all predictors are represented in numeric format, which is used by function tenf.crr to do ten fold cross-validation

Note

This function requires that the rms package is attached

Author(s)

Michael W. Kattan, Ph.D. and Changhong Yu. Department of Quantitative Health Sciences, Cleveland Clinic

References

Michael W. Kattan, Glenn Heller and Murray F. Brennan (2003). A competing-risks nomogram for sarcoma-specific death following local recurrence. Statistics in Medicine. Stat Med. 2003;22:3515-3525.

See Also

cph crr nomogram.crr

Examples

```
dd <- datadist(prostate.dat)
options(datadist = "dd")
prostate.f <- cph(Surv(TIME_EVENT,EVENT_DOD == 1) ~ TX  + rcs(PSA,3) +
           BX_GLSN_CAT +  CLIN_STG + rcs(AGE,3) +
           RACE_AA, data = prostate.dat,
           x = TRUE, y = TRUE, surv = TRUE,time.inc = 144)
prostate.crr <- crr.fit(prostate.f, cencode = 0, failcode = 1)
prostate.crr
```

---

groupci *Assess Calibration for a Competing Risks Endpoint*

Description

Uses cuminc to estimate the cumulative incidence at a given time point within subgroups of a continuous variable (often predicted failure probabilities from a crr.fit model).

Usage

```
groupci(
  x,
  ftime,
  fstatus,
  u,
  cencode = 0,
  failcode = 1,
  ci = TRUE,
  m = 50,
  g = NULL,
  cuts = NULL,
  pl = TRUE,
  conf.int = 0.95,
  xlab = NULL,
  ylab = NULL,
  xlim = c(0, 1),
  ylim = c(0, 1),
  lty = 1,
  add = FALSE,
  cex.subtitle = FALSE,
  ab = TRUE,
  a = 0,
  b = 1,
  ...
)
```

Arguments

x	A numeric variable to assess calibration for
ftime	The event time variable. See cuminc.
fstatus	The event status variable. See cuminc.
u	A single time point to assess calibration at
cencode	The censoring event code. See cuminc.
failcode	The value of fstatus that indicates the event of interest
ci	Should the failure probability be assessed? Defaults to TRUE. If FALSE, the event-free probability is assessed.
m	Minimum number of observations in each group. See cut2.
g	Number of quantile groups. See cut2.
cuts	Actual cut points to use for x. See cut2.
pl	Should the calibration curve be plotted? Defaults to TRUE.
conf.int	Confidence limit on error bars. Defaults to 0.95. Set to FALSE to suppress.
xlab	The x-axis label. Uses label or name of calling argument if not specified.
ylab	The y-axis label. Uses a default label is none specified.
xlim	The x-axis limits. Defaults to c(0, 1).
ylim	The y-axis limits. Defaults to c(0, 1).
lty	Line type for connecting estimates and error bars
add	Defaults to FALSE. Set to TRUE to add to an existing plot.
cex.subtitle	Character size for subtitle (default 0.7). Defaults to FALSE to suppress.
ab	Should a reference line be added? See abline.
a	The intercept for the reference line. See abline.
b	The slope for the reference line. See abline.
...	Other arguments passed to lines and errbar.

Details

To divide x, the function first looks for cuts, then g, then m.

Value

A matrix with a row for each group of x:

x	Mean value of x
n	Number of observations
events	Number of events (of type failcode)
ci	Estimated cumulative incidence (or event-free probability if ci=FALSE)
std.err	Estimated standard error for the ci value

If pl=TRUE, a calibration plot is also displayed.

Author(s)

Changhong Yu, Michael Kattan, Ph.D
Department of Quantitative Health Sciences
Cleveland Clinic

See Also

cuminc pred.ci cut2

Examples

```
dd <- datadist(prostate.dat)
options(datadist = "dd")
prostate.f <- cph(Surv(TIME_EVENT,EVENT_DOD == 1) ~ TX  + rcs(PSA,3) +
           BX_GLSN_CAT +  CLIN_STG + rcs(AGE,3) +
           RACE_AA, data = prostate.dat,
           x = TRUE, y = TRUE, surv = TRUE,time.inc = 144)
prostate.crr <- crr.fit(prostate.f, cencode = 0, failcode = 1)

# Cross-validated predictions
prostate.dat$preds.cv.prostate.crr.120 <- tenf.crr(prostate.crr, time = 120, fold = 2)

with(prostate.dat,
     groupci(preds.cv.prostate.crr.120, ftime = TIME_EVENT,
             fstatus =EVENT_DOD, g = 5, u = 120,
             xlab = "Nomogram predicted 10-year cancerspecific mortality",
             ylab = "Observed predicted 10-year cancerspecific mortality")
)
```

Newlabels.cmprsk *Change the Predictor Labels for a Competing Risks Regression Model*

Description

Uses the Newlabels function to change the labels predictors when constructing a nomogram.

Usage

```
## S3 method for class 'cmprsk'
Newlabels(fit, labels, ...)
```

Arguments

fit	A model fit by crr.fit
labels	A character vector specifying the new labels for variables in a fit.
...	Other arguments for Newlabels

Details

To give new labels for all variables, you can specify labels of the form `labels = c("Age in Years","Cholesterol")`, where the list of new labels is assumed to be the length of all main effect-type variables in the fit and in their original order in the model formula. You may specify a named vector to give new labels in any order for a subset of the variables, e.g., `labels = c(age = "Age in Years", chol = "Cholesterol")`.

Value

A new `crr.fit` object with adjusted labels

Author(s)

Changhong Yu. Department of Quantitative Health Sciences, Cleveland Clinic

See Also

`Newlevels.cmprsk`

Examples

```
dd <- datadist(prostate.dat)
options(datadist = "dd")
prostate.f <- cph(Surv(TIME_EVENT,EVENT_DOD == 1) ~ TX  + rcs(PSA,3) +
           BX_GLSN_CAT +  CLIN_STG + rcs(AGE,3) +
           RACE_AA, data = prostate.dat,
           x = TRUE, y = TRUE, surv = TRUE,time.inc = 144)
prostate.crr <- crr.fit(prostate.f, cencode = 0, failcode = 1)
prostate.g <- Newlabels(
prostate.crr,
c(
  TX = 'Treatment options',
  BX_GLSN_CAT = 'Biopsy Gleason Score Sum',
  CLIN_STG = 'Clinical stage'
)
)
```

Newlevels.cmprsk *Change the Level Labels of Categorical Predictors for a Competing Risks Regression Model*

Description

Uses the `Newlevels` function to change the labels predictors when constructing a nomogram.

Usage

```
## S3 method for class 'cmprsk'
Newlevels(fit, levels, ...)
```

Arguments

fit	A model fit by crr.fit
levels	A list of named vectors specifying the new level labels for categorical predictors.
...	Other arguments for Newlevels

Value

A new crr.fit object with adjusted labels on the factor levels

Note

This will override parms and datadist information that were stored with the fit.

Author(s)

Changhong Yu. Department of Quantitative Health Sciences, Cleveland Clinic

See Also

Newlabels.cmprsk

Examples

```
dd <- datadist(prostate.dat)
options(datadist = "dd")
prostate.f <- cph(Surv(TIME_EVENT,EVENT_DOD == 1) ~ TX  + rcs(PSA,3) +
           BX_GLSN_CAT +  CLIN_STG + rcs(AGE,3) +
           RACE_AA, data = prostate.dat,
           x = TRUE, y = TRUE, surv = TRUE,time.inc = 144)
prostate.crr <- crr.fit(prostate.f, cencode = 0, failcode = 1)
prostate.g <- Newlevels(prostate.crr, list(TX = c('Treatment 1', 'Treatment 2', 'Treatment 3')))
```

---

nomogram.crr *Construct a Nomogram for a Competing Risks Regression Model*

Description

Draws a partial nomogram from a crr.fit object that can be used to manually obtain predicted values from from a competing risks regression model.

Usage

```
nomogram.crr(
  fit,
  failtime,
  ci = TRUE,
  ...,
  adj.to,
  lp = TRUE,
  lp.at,
  lplabel = "Linear Predictor",
  fun.at,
  fun.lp.at,
  funlabel = "Predicted Value",
  fun.side,
  interact = NULL,
  intercept = 1,
  conf.int = FALSE,
  col.conf = c(1, 12),
  conf.space = c(0.08, 0.2),
  conf.lp = c("representative", "all", "none"),
  est.all = TRUE,
  abbrev = FALSE,
  minlength = 4,
  maxscale = 100,
  nint = 10,
  label.every = 1,
  force.label = FALSE,
  xfrac = 0.35,
  cex.axis = 0.85,
  cex.var = 1,
  col.grid = FALSE,
  vnames = c("labels", "names"),
  varname.label = TRUE,
  varname.label.sep = "=",
  ia.space = 0.7,
  tck = -0.009,
  lmgp = 0.4,
  omit = NULL,
  naxes,
  points.label = "Points",
  total.points.label = "Total Points",
  total.sep.page = FALSE,
  total.fun,
  verbose = FALSE,
  total.min,
  total.max,
  mikeomit = NULL
)
```

Arguments

fit	A model fit by crr.fit
failtime	A vector of time points to display failure probability axes for.
ci	Should the failure probability be displayed? Defaults to TRUE. If FALSE, the event-free probability is displayed.
...	Settings of variables to use in constructing axes. If datadist was in effect, the default is to use pretty(total range, nint) for continuous variables, and the class levels for discrete ones. For legend.nomabbrev, ... specifies optional parameters to pass to the legend. Common ones are bty = "n" to suppress drawing the box. You may want to specify a non-proportionally spaced font (e.g., courier) number if abbreviations are more than one letter long. This will make the abbreviation definitions line up (e.g., specify font = 2, the default for courier). Ignored for print.
adj.to	If datadist was not defined for all predictors, it is required to define adjustment settings for the undefined ones (e.g. adj.to=list(age=50, sex="female")).
lp	Set to FALSE to suppress creation of an axis for scoring $X\beta$. Defaults to TRUE.
lp.at	If lp=TRUE, specifies a vector of settings of $X\beta$. Default is to use pretty(range of linear predictors, nint).
lplabel	label for linear predictor axis. Default is "Linear Predictor".
fun.at	Function values to label on axis. Default fun evaluated at lp.at. If more than one fun was specified, using a vector for fun.at will cause all functions to be evaluated at the same argument values. To use different values, specify a list of vectors for fun.at, with elements corresponding to the different functions (lists of vectors also applies to fun.lp.at and fun.side).
fun.lp.at	If you want to evaluate one of the functions at a different set of linear predictor values than may have been used in constructing the linear predictor axis, specify a vector or list of vectors of linear predictor values at which to evaluate the function. This is especially useful for discrete functions. The presence of this attribute also does away with the need for nomogram to compute numerical approximations of the inverse of the function. It also allows the user-supplied function to return factor objects, which is useful when e.g. a single tick mark position actually represents a range. If the fun.lp.at parameter is present, the fun.at vector for that function is ignored.
funlabel	Label for fun axis. If more than one function was given but funlabel is of length one, it will be duplicated as needed. If fun is a list of functions for which you specified names (see the final example below), these names will be used as labels.
fun.side	A vector or list of vectors of side parameters for the axis function for labeling function values. Values may be 1 to position a tick mark label below the axis (the default), or 3 for above the axis. If for example an axis has 5 tick mark labels and the second and third will run into each other, specify fun.side=c(1,1,3,1,1) (assuming only one function is specified as fun).
interact	When a continuous variable interacts with a discrete one, axes are constructed so that the continuous variable moves within the axis, and separate axes represent

levels of interacting factors. For interactions between two continuous variables, all but the axis variable must have discrete levels defined in `interact`. For discrete interacting factors, you may specify levels to use in constructing the multiple axes. For continuous interacting factors, you must do this. Examples: `interact=list(age=seq(10,70,by=10), treat=c("A","B","D"))`.

`intercept` For models such as the ordinal logistic model with multiple intercepts, specifies which one to use in evaluating the linear predictor.

`conf.int` Confidence levels to display for each scoring. Default is `FALSE` to display no confidence limits. Setting `conf.int` to `TRUE` is the same as setting it to `c(0.7, 0.9)`, with the line segment between the 0.7 and 0.9 levels shaded using gray scale.

`col.conf` Colors corresponding to `conf.int`. Use fractions for gray scale (for UNIX S-PLUS).

`conf.space` A 2-element vector with the vertical range within which to draw confidence bars, in units of 1=spacing between main bars. Four heights are used within this range (8 for the linear predictor if more than 16 unique values were evaluated), cycling them among separate confidence intervals to reduce overlapping.

`conf.lp` Default is `"representative"` to group all linear predictors evaluated into deciles, and to show, for the linear predictor confidence intervals, only the mean linear predictor within the deciles along with the median standard error within the deciles. Set `conf.lp="none"` to suppress confidence limits for the linear predictors, and to `"all"` to show all confidence limits.

`est.all` To plot axes for only the subset of variables named in `...{}`, set `est.all=FALSE`. Note: This option only works when zero has a special meaning for the variables that are omitted from the graph.

`abbrev` Set to `TRUE` to use the `abbreviate` function to abbreviate levels of categorical factors, both for labeling tick marks and for axis titles. If you only want to abbreviate certain predictor variables, set `abbrev` to a vector of character strings containing their names.

`minlength` Applies if `abbrev=TRUE`. Is the minimum abbreviation length passed to the `abbreviate` function. If you set `minlength=1`, the letters of the alphabet are used to label tick marks for categorical predictors, and all letters are drawn no matter how close together they are. For labeling axes (interaction settings), `minlength=1` causes `minlength=4` to be used.

`maxscale` Default maximum point score is 100.

`nint` Number of intervals to label for axes representing continuous variables. See `pretty`.

`label.every` Specify `label.every=i` to label on every `i`th tick mark.

`force.label` Set to `TRUE` to force every tick mark intended to be labeled to have a label plotted (whether the labels run into each other or not)

`xfrac` Fraction of horizontal plot to set aside for axis titles

`cex.axis` Character size for tick mark labels

`cex.var` Character size for axis titles (variable names)

col.grid — If col.grid=1, no gray scale is used, but an ordinary line is drawn. If 0<col.grid<1, a col (gray scale) of col.grid is used to draw vertical reference lines for major axis divisions and col.grid/2 for minor divisions. The default is col.grid=FALSE, i.e., reference lines are omitted. Specifying col.grid=TRUE is the same as specifying a gray scale level of col.grid=.2 (5 for Windows S-PLUS).

vnames — By default, variable labels are used to label axes. Set vnames="names" to instead use variable names.

varname.label — In constructing axis titles for interactions, the default is to add "(interacting.varname=level)" on the right. Specify varname.label=FALSE to instead use "(level)".

varname.label.sep — If varname.label=TRUE, you can change the separator to something other than = by specifying this parameter.

ia.space — When multiple axes are draw for levels of interacting factors, the default is to group combinations related to a main effect. This is done by spacing the axes for the second to last of these within a group only 0.7 (by default) of the way down as compared with normal space of 1 unit.

tck — See tck under par.

lmgp — Spacing between numeric axis labels and axis (see par for mgp)

omit — Vector of character strings containing names of variables for which to suppress drawing axes. Default is to show all variables.

naxes — Maximum number of axes to allow on one plot. If the nomogram requires more than one "page", the "Points" axis will be repeated at the top of each page when necessary.

points.label — A character string giving the axis label for the points scale

total.points.label — A character string giving the axis label for the total points scale

total.sep.page — Set to TRUE to force the total points and later axes to be placed on a separate page

total.fun — A user-provided function that will be executed before the total points axis is drawn. Default is not to execute a function. This is useful e.g. when total.sep.page=TRUE and you wish to use locator to find the coordinates for positioning an abbreviation legend before it's too late and a new page is started (i.e., total.fun=function()print(locator(1)

verbose — Set to TRUE to get printed output detailing how tick marks are chosen and labeled for function axes. This is useful in seeing how certain linear predictor values cannot be solved for using inverse linear interpolation on the (requested linear predictor values, function values at these lp values). When this happens you will see NAs in the verbose output, and the corresponding tick marks will not appear in the nomogram.

total.min — Setting the minimal value in the total point axis on the nomogram.

total.max — Setting the maximal value in the total point axis.

mikeomit — The predictor variables specified by their names here will not be shown in the nomogram. The predicted outcome based on this reduced nomogram would be the same as if users were using the full version of the nomogram by entering the some values for the predictors remaining in the reduced nomogram but adjusted values for the hiden predictors so that 0 points will be achieved from these hiden predictor variables in the full nomogram.

Value

A list of class "nomogram" that contains information used in plotting the axes. Please see nomogram for details.

Author(s)

Changhong Yu, Michael Kattan, Ph.D
Department of Quantitative Health Sciences
Cleveland Clinic

Frank Harrell
Department of Biostatistics
Vanderbilt University
<f.harrell@vanderbilt.edu>

References

Banks J: Nomograms. Encylopedia of Statistical Sciences, Vol 6. Editors: S Kotz and NL Johnson. New York: Wiley; 1985.

Lubsen J, Pool J, van der Does, E: A practical device for the application of a diagnostic or prognostic function. Meth. Inform. Med. 17:127–129; 1978.

Wikipedia: Nomogram, https://en.wikipedia.org/wiki/Nomogram.

Michael W. Kattan, Glenn Heller and Murray F. Brennan (2003). A competing-risks nomogram for sarcoma-specific death following local recurrence. Statistics in Medicine. Stat Med. 2003;22:3515-3525.

See Also

nomogram nomogram.mk6 crr.fit

Examples

```
data(prostate.dat)
dd <- datadist(prostate.dat)
options(datadist = "dd")
prostate.f <- cph(Surv(TIME_EVENT,EVENT_DOD == 1) ~ TX  + rcs(PSA,3) +
           BX_GLSN_CAT +  CLIN_STG + rcs(AGE,3) +
           RACE_AA, data = prostate.dat,
           x = TRUE, y= TRUE, surv=TRUE,time.inc = 144)
prostate.crr <- crr.fit(prostate.f,cencode = 0,failcode = 1)
## make a CRR nomogram
nomogram.crr(prostate.crr,failtime = 120,lp=FALSE,
funlabel = "Predicted 10-year cumulative incidence")
```

nomogram.mk6 *Construct a Nomogram for a Regression Model*

Description

Draws a partial nomogram that can be used to manually obtain predicted values. This is a modified version of nomogram.

Usage

```
nomogram.mk6(
  fit,
  ...,
  adj.to,
  lp = TRUE,
  lp.at,
  lplabel = "Linear Predictor",
  fun,
  fun.at,
  fun.lp.at,
  funlabel = "Predicted Value",
  fun.side,
  interact = NULL,
  intercept = 1,
  conf.int = FALSE,
  col.conf = c(1, 12),
  conf.space = c(0.08, 0.2),
  conf.lp = c("representative", "all", "none"),
  est.all = TRUE,
  abbrev = FALSE,
  minlength = 4,
  maxscale = 100,
  nint = 10,
  label.every = 1,
  force.label = FALSE,
  xfrac = 0.35,
  cex.axis = 0.85,
  cex.var = 1,
  col.grid = NULL,
  vnames = c("labels", "names"),
  varname.label = TRUE,
  varname.label.sep = "=",
  ia.space = 0.7,
  tck = NA,
  tcl = -0.25,
  lmgp = 0.4,
  omit = NULL,
```

```
  naxes,
  points.label = "Points",
  total.points.label = "Total Points",
  total.sep.page = FALSE,
  total.fun,
  verbose = FALSE,
  cap.labels = FALSE,
  total.min,
  total.max,
  survtime,
  mikeomit = NULL
)
```

Arguments

fit	A regression model that was created with library(rms) in effect, and (usually) with options(datadist = "object.name") in effect.
...	Settings of variables to use in constructing axes. If datadist was in effect, the default is to use pretty(total range, nint) for continuous variables, and the class levels for discrete ones. For legend.nomabbrev, ... specifies optional parameters to pass to legend. Common ones are bty = "n" to suppress drawing the box. You may want to specify a non-proportionally spaced font (e.g., courier) number if abbreviations are more than one letter long. This will make the abbreviation definitions line up (e.g., specify font = 2, the default for courier). Ignored for print.
adj.to	If you didn't define datadist for all predictors, you will have to define adjustment settings for the undefined ones, e.g. adj.to= list(age = 50, sex = "female").
lp	Set to FALSE to suppress creation of an axis for scoring $X\beta$.
lp.at	If lp=TRUE, lp.at may specify a vector of settings of $X\beta$. Default is to use pretty(range of linear predictors, nint).
lplabel	label for linear predictor axis. Default is "Linear Predictor".
fun	on another axis. If more than one transformation is plotted, put them in a list, e.g. list(function(x) x/2, function(x) 2*x). Any function values equal to NA will be ignored.
fun.at	function values to label on axis. Default fun evaluated at lp.at. If more than one fun was specified, using a vector for fun.at will cause all functions to be evaluated at the same argument values. To use different values, specify a list of vectors for fun.at, with elements corresponding to the different functions (lists of vectors also applies to fun.lp.at and fun.side).
fun.lp.at	If you want to evaluate one of the functions at a different set of linear predictor values than may have been used in constructing the linear predictor axis, specify a vector or list of vectors of linear predictor values at which to evaluate the function. This is especially useful for discrete functions. The presence of this attribute also does away with the need for nomogram to compute numerical approximations of the inverse of the function. It also allows the user-supplied

function to return `factor` objects, which is useful when e.g. a single tick mark position actually represents a range. If the `fun.lp.at` parameter is present, the `fun.at` vector for that function is ignored.

`funlabel` label for `fun` axis. If more than one function was given but funlabel is of length one, it will be duplicated as needed. If `fun` is a list of functions for which you specified names (see the final example below), these names will be used as labels.

`fun.side` a vector or list of vectors of `side` parameters for the `axis` function for labeling function values. Values may be 1 to position a tick mark label below the axis (the default), or 3 for above the axis. If for example an axis has 5 tick mark labels and the second and third will run into each other, specify `fun.side=c(1,1,3,1,1)` (assuming only one function is specified as `fun`).

`interact` When a continuous variable interacts with a discrete one, axes are constructed so that the continuous variable moves within the axis, and separate axes represent levels of interacting factors. For interactions between two continuous variables, all but the axis variable must have discrete levels defined in `interact`. For discrete interacting factors, you may specify levels to use in constructing the multiple axes. For continuous interacting factors, you must do this. Examples: `interact = list(age = seq(10,70,by=10),treat = c("A","B","D"))`.

`intercept` for models such as the ordinal logistic model with multiple intercepts, specifies which one to use in evaluating the linear predictor.

`conf.int` confidence levels to display for each scoring. Default is `FALSE` to display no confidence limits. Setting `conf.int` to `TRUE` is the same as setting it to `c(0.7, 0.9)`, with the line segment between the 0.7 and 0.9 levels shaded using gray scale.

`col.conf` colors corresponding to `conf.int`. Use fractions for gray scale(for UNIX S-PLUS).

`conf.space` a 2-element vector with the vertical range within which to draw confidence bars, in units of 1=spacing between main bars. Four heights are used within this range (8 for the linear predictor if more than 16 unique values were evaluated), cycling them among separate confidence intervals to reduce overlapping

`conf.lp` default is `"representative"` to group all linear predictors evaluated into deciles, and to show, for the linear predictor confidence intervals, only the mean linear predictor within the deciles along with the median standard error within the deciles. Set `conf.lp = "none"` to suppress confidence limits for the linear predictors, and to `"all"` to show all confidence limits.

`est.all` To plot axes for only the subset of variables named in `...`, set `est.all = FALSE`. Note: This option only works when zero has a special meaning for the variables that are omitted from the graph

`abbrev` Set to `TRUE` to use the `abbreviate` function to abbreviate levels of categorical factors, both for labeling tick marks and for axis titles. If you only want to abbreviate certain predictor variables, set `abbrev` to a vector of character strings containing their names.

`minlength` `abbreviate` function. If you set `minlength = 1`, the letters of the alphabet are used to label tick marks for categorical predictors, and all letters are drawn no

matter how close together they are. For labeling axes (interaction settings), `minlength = 1` causes `minlength = 4` to be used

`maxscale` default maximum point score is 100

`nint` number of intervals to label for axes representing continuous variables. See `pretty`.

`label.every` Specify `label.every = i` to label on every `i`th tick mark

`force.label` set to `TRUE` to force every tick mark intended to be labeled to have a label plotted (whether the labels run into each other or not)

`xfrac` fraction of horizontal plot to set aside for axis titles

`cex.axis` character size for tick mark labels

`cex.var` character size for axis titles (variable names)

`col.grid` If left unspecified, no vertical reference lines are drawn. Specify a vector of length one (to use the same color for both minor and major reference lines) or two (corresponding to the color for the major and minor divisions, respectively) containing colors, to cause vertical reference lines to the top points scale to be drawn. For R, a good choice is `col.grid = gray(c(0.8, 0.95))`.

`vnames` By default, variable labels are used to label axes. Set `vnames = "names"` to instead use variable names.

`varname.label` In constructing axis titles for interactions, the default is to add `(interacting.varname = level)` on the right. Specify `varname.label = FALSE` to instead use `"(level)"`.

`varname.label.sep`
If `varname.label = TRUE`, you can change the separator to something other than = by specifying this parameter.

`ia.space` When multiple axes are draw for levels of interacting factors, the default is to group combinations related to a main effect. This is done by spacing the axes for the second to last of these within a group only 0.7 (by default) of the way down as compared with normal space of 1 unit.

`tck` see `tck` under `par`

`tcl` length of tick marks in nomogram

`lmgp` spacing between numeric axis labels and axis (see `par` for `mgp`)

`omit` vector of character strings containing names of variables for which to suppress drawing axes. Default is to show all variables.

`naxes` maximum number of axes to allow on one plot. If the nomogram requires more than one "page", the "Points" axis will be repeated at the top of each page when necessary.

`points.label` a character string giving the axis label for the points scale

`total.points.label`
a character string giving the axis label for the total points scale

`total.sep.page` set to `TRUE` to force the total points and later axes to be placed on a separate page

`total.fun` a user-provided function that will be executed before the total points axis is drawn. Default is not toe xecute a function. This is useful e.g. when `total.sep.page = TRUE` and you wish to use `locator` to find the coordinates for positioning an abbreviation legend before it's too late and a new page is started (i.e., `total.fun = function() print(locator(1))`).

verbose	set to TRUE to get printed output detailing how tick marks are chosen and labeled for function axes. This is useful in seeing how certain linear predictor values cannot be solved for using inverse linear interpolation on the (requested linear predictor values, function values at these lp values). When this happens you will see NAs in the verbose output, and the corresponding tick marks will not appear in the nomogram.
cap.labels	logical: should the factor labels have their first letter capitalized?
total.min	the minimum point for the total point axis
total.max	the maxmum point for the total point axis
survtime	specified survival time for the predicted survival probability
mikeomit	a modified version of omit

Value

A nomogram

Note

Used internally for nomogram.crr. See nomogram.

References

Banks J: Nomograms. Encylopedia of Statistical Sciences, Vol 6. Editors: S Kotz and NL Johnson. New York: Wiley; 1985.

Lubsen J, Pool J, van der Does, E: A practical device for the application of a diagnostic or prognostic function. Meth. Inform. Med. 17:127–129; 1978.

Wikipedia: Nomogram, https://en.wikipedia.org/wiki/Nomogram.

See Also

nomogram.crr rms nomogram plot.summary.rms axis pretty approx

Examples

```
data(prostate.dat)
dd <- datadist(prostate.dat)
options(datadist = "dd")
prostate.f <- cph(Surv(TIME_EVENT,EVENT_DOD == 1) ~ TX  + rcs(PSA,3) +
           BX_GLSN_CAT +  CLIN_STG + rcs(AGE,3) +
           RACE_AA, data = prostate.dat,
           x = TRUE, y= TRUE, surv=TRUE,time.inc = 144)
## make a cph nomogram
nomogram.mk6(prostate.f, survtime=120, lp=FALSE,
funlabel = "Predicted 10-year cumulative incidence")
```

pred.ci *Extract Cumulative Incidence Estimates at a Specified Time Point*

Description

Extracts the cumulative incidence estimates from a cuminc object for the cause of interest at a specified time point into a data.frame.

Usage

```
pred.ci(cum, tm1, failcode = 1)
```

Arguments

cum	A cuminc object
tm1	A single time point to return the cumulative incidence at
failcode	The value of the status column that indicates the event of interest

Value

A data.frame with 3 columns:

Group	The group name. If the group argument was used in the cuminc fit, there will be one row per group. Otherwise this is non-informative.
CI.Prob	The cumulative incidence probability at the desired time point
CI.Var	The estimated variance of the cumulative incidence estimate

Author(s)

Michael W. Kattan, Ph.D. and Changhong Yu.
Department of Quantitative Health Sciences, Cleveland Clinic

See Also

cuminc

Examples

```
cum <- cmprsk::cuminc(prostate.dat$TIME_EVENT, prostate.dat$EVENT_DOD, cencode = 0)
pred.ci(cum, 60, failcode = 1)
```

predict.cmprsk *Calculate the Failure Time Probability from a Competing Risks Regression Model*

Description

Computes the predicted probability of the event of interest at a specified time point for a competing risks regression model fit by crr.fit. This function is adapted from predict.crr.

Usage

```
## S3 method for class 'cmprsk'
predict(object, newdata = NULL, time, lps, ...)
```

Arguments

object	A model fit by crr.fit
newdata	A data.frame for prediction containing values of covariates in the model. If missing, the model development dataset (object$cphdat) is used.
time	A single time point to calculate the failure probability
lps	Should the linear predictor be returned instead of the failure probability? Defaults to FALSE.
...	Additional arguments such as cov2 as in crr

Value

A vector of failure probabilities at the specified time point (or linear predictors if lps=TRUE) with length equal to the number of rows in newdata

Author(s)

Michael W. Kattan, Ph.D. and Changhong Yu.
Department of Quantitative Health Sciences, Cleveland Clinic

References

Fine JP and Gray RJ (1999) A proportional hazards model for the subdistribution of a competing risk. JASA 94:496-509.

See Also

crr.fit, predict.crr

Examples

```
dd <- datadist(prostate.dat)
options(datadist = "dd")
prostate.f <- cph(Surv(TIME_EVENT,EVENT_DOD == 1) ~ TX  + rcs(PSA,3) +
           BX_GLSN_CAT +  CLIN_STG + rcs(AGE,3) +
           RACE_AA, data = prostate.dat,
           x = TRUE, y = TRUE, surv = TRUE,time.inc = 144)
prostate.crr <- crr.fit(prostate.f, cencode = 0, failcode = 1)
predict(prostate.crr, time = 60)
```

---

prostate.dat *Prostate cancer data set*

Description

This is an artificial prostate cancer dataset used for illustrating the usage of functions in QHScrnomo

Usage

```
prostate.dat
```

Format

prostate.dat**:**

A data frame with 2000 observations on the following 9 variables:

UNIQID patient ID

TX Treatment options of prostate cancera with levels EBRT, PI, RP

PSA Pre-treatment PSA levels

BX_GLSN_CAT Biopsy Gleason Score Sum. a factor with levels 1 for 2-6 2 for 7 and 3 for 8-10

CLIN_STG Clinical stage with levels T1, T2, T3

AGE Age at treatment date

RACE_AA patient ethnicity.a factor with levels 0 for other and 1 for African American

TIME_EVENT follow up time in months

EVENT_DOD followup status, 0 - censored, 1 - died of prostate cancer, 2 - died of other causes

Details

This is a simulated dataset

sas.cmprsk *Generate a Prediction Equation for a Competing Risks Regression Model*

Description

Uses Function to generate and print the linear predictor to the console or file which can be hard-coded into a function for evaluation on the scale of the original data. A time point can optionally be specified to retrieve the base sub-cumulative rate at that time point (i.e, the failure probability when all covariate values are 0).

Usage

```
sas.cmprsk(f, time = NA, baseonly = FALSE, file = "", append = FALSE)
```

Arguments

f	A model fit by crr.fit
time	A single time point to calculate the failure probability
baseonly	Should we only display the failure probability at the specified time? Defaults to FALSE.
file	An optional connection or character string naming the file to print to.
append	Only used if the file argument is specified. If TRUE, the output will be appended to file, otherwise it will overwritten.

Value

A printed equation using cat (invisible NULL)

Author(s)

Changhong Yu. Department of Quantitative Health Sciences, Cleveland Clinic

See Also

crr.fit sascode Function

Examples

```
dd <- datadist(prostate.dat)
options(datadist = "dd")
prostate.f <- cph(Surv(TIME_EVENT,EVENT_DOD == 1) ~ TX  + rcs(PSA,3) +
           BX_GLSN_CAT +  CLIN_STG + rcs(AGE,3) +
           RACE_AA, data = prostate.dat,
           x = TRUE, y = TRUE, surv = TRUE,time.inc = 144)
prostate.crr <- crr.fit(prostate.f, cencode = 0, failcode = 1)
sas.cmprsk(prostate.crr, time = 60)
```

summary.cmprsk *Summary of a Competing Risks Regression Model*

Description

Uses the summary.rms method to construct a summary for the competing risks regression model fit from crr.fit.

Usage

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

Arguments

object	A model fit by crr.fit
...	Other arguments for summary.rms

Value

A summary.rms matrix

Note

This function requires that the rms package is attached

Author(s)

Changhong Yu. Department of Quantitative Health Sciences, Cleveland Clinic

See Also

crr.fit summary.rms

Examples

```
dd <- datadist(prostate.dat)
options(datadist = "dd")
prostate.f <- cph(Surv(TIME_EVENT,EVENT_DOD == 1) ~ TX  + rcs(PSA,3) +
           BX_GLSN_CAT +  CLIN_STG + rcs(AGE,3) +
           RACE_AA, data = prostate.dat,
           x = TRUE, y = TRUE, surv = TRUE,time.inc = 144)
prostate.crr <- crr.fit(prostate.f, cencode = 0, failcode = 1)
summary(prostate.crr)
```

tenf.crr *Obtain K-Fold Cross-Validated Predictions*

Description

Computes "out-of-sample" predictions by K-fold cross-validation for each observation in the modeling data set from a crr.fit object.

Usage

```
tenf.crr(fit, time = NULL, lps = FALSE, fold = 10, trace = TRUE)
```

Arguments

fit	A model fit by crr.fit
time	A single time point to calculate the failure probability
lps	Should the linear predictor be returned instead of the failure probability? Defaults to FALSE.
fold	The number of folds. Defaults to 10.
trace	Should the progress of cross-validation be printed to the console? Defaults to TRUE.

Value

A vector of failure probabilities at the specified time point (or linear predictors if lps=TRUE) with length equal to the number of rows in the original data set.

Author(s)

Changhong Yu, Michael Kattan, Ph.D
Department of Quantitative Health Sciences
Cleveland Clinic

See Also

crr.fit crr

Examples

```
dd <- datadist(prostate.dat)
options(datadist = "dd")
prostate.f <- cph(Surv(TIME_EVENT,EVENT_DOD == 1) ~ TX  + rcs(PSA,3) +
           BX_GLSN_CAT +  CLIN_STG + rcs(AGE,3) +
           RACE_AA, data = prostate.dat,
           x = TRUE, y = TRUE, surv = TRUE,time.inc = 144)
prostate.crr <- crr.fit(prostate.f, cencode = 0, failcode = 1)
tenf.crr(prostate.crr, time = 120, fold = 2)
```

Index