Package 'QHScrnomo'

November 5, 2021

Type Package

Title Construct Nomogram for Competing Risk Models

Version 2.2.0

Date 2021-10-27

Depends R (>= 3.5.0),Hmisc,rms,cmprsk

Suggests rmarkdown, knitr, survival

Description Nomogram is constructed to predict cumulative incidence rate, which is calculated by adjusting for competing causes other than that of interest. K-fold cross-validation function is implemented to validate the performance. Competing risk version of concordance index is calculated. Methods are as described in: Kattan MW, Heller G, Brennan MF (2003) <doi:10.1002/sim.1574>.

License GPL-2

NeedsCompilation yes

Config/testthat/edition 3

RoxygenNote 7.1.2

VignetteBuilder knitr

Encoding UTF-8

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

Maintainer Xinge Ji <jix@ccf.org>

Repository CRAN

Date/Publication 2021-11-05 13:20:02 UTC

R topics documented:

`addOffset4ModelFrame` *Internal function to Calculate Offset for Data*

Description

Internal function to Calculate Offset for Data

Usage

```
addOffset4ModelFrame(Terms, newdata, offset = 0)
```

Arguments

`Terms`	term value
`newdata`	data frame for prediction. Each row of the data frame contains values of covariates that are required in the crr model. If missing, the original data set that was used to develop the crr model will be used for prediction.
`offset`	offset value

Details

Internally calculate offset for a data

Value

newdata

anova.cmprsk *anova table for competing risks regression*

Description

generate anova table for crr

Usage

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

Arguments

object	a competing risks regression model object built from funciton crr.fit
...	other arguments

Details

generate anova table for competing risks regression model

Value

anova table in matrix

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)
## anova test
anova(prostate.crr)
```

cindex *Concordance index calculation*

Description

Calculate concordance index

Usage

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

Arguments

prob	predicted risk of failure event, either probability or risk score
fstatus	failure(event) variable
ftime	follow-up time variable for survival or competing risks predictions
type	type of regression models corresponding to different type of outcomes. 'logis' is the default value for binary outcome, 'surv' for ordinary survival outcome and 'crr' for competing risks outcome.
failcode	coding for failure(event). 1 is the default value.
cencode	coding for censoring. 0 is the defaul
tol	error tolerance. the default value is 1e-20.

Details

to calculate the discrimination metric, concordance index for binary, time-to event and competing risks outcomes

Value

a vector of returned values.

N	the total number of observations in the input data
n	the nonmissing number of observations that was used f or calculation
usable	the total number of usable pairs.
oncordant	the number of concordant pairs
cindex	the concordance index that equal to the number of concordant pairs divided by the total number of usable pairs.

Author(s)

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

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)

## ten fold cross validation
prostate.dat$preds.tenf.cv.prostate.crr.120 <-
                                      tenf.crr(prostate.crr,time = 120)

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

`crr.fit` *Fit Competing Risks Regression Model Fits a competing risks regression model from an existing Cox proportional hazards object and allows a nomogram to be constructed from the competing risks regression object.*

Description

This function uses the `crr` function in the `cmprsk` package to construct a competing risk regression object.

Usage

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

Arguments

`fit`	a Cox proportional hazards regression model constructed from `cph` in rms library (by Frank Harrell)
`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

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 and cmprsk libraries are 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

```
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)
## anova test
anova(prostate.crr)
## hazards ratio
summary(prostate.crr)

## ten fold cross validation
prostate.dat$preds.tenf.cv.prostate.crr.120 <-
                                      tenf.crr(prostate.crr,time = 120)

## make a CRR nomogram
nomogram.crr(prostate.crr,failtime = 120,lp=FALSE,
funlabel = "Predicted 10-year cumulative incidence")

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

## generate the calibration curve for predicted 10-year cancer
## specific mortality

with(prostate.dat,
     groupci(preds.tenf.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")
)
```

---

groupci *make calibration curve for competing risks endpoint*

Description

Cumulative Incidence Estimates vs. a Continuous Variable

Usage

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

Arguments

x	a continuous variable
ftime	vector of follow-up time
fstatus	vector of failure status
cencode	value indicating cencering.
failcode	value indicating event of interest
ci	logical flag to output event free probability if setting FALSE
m	desired minimum number of observations in a group
g	number of quantile groups
cuts	actual cuts in x, e.g. c(0,1,2) to use [0,1), [1,2].
u	time for which to estimate cumulative incidence
pl	TRUE to plot results
conf.int	defaults to .95 for 0.95 confidence bars. Set to FALSE to suppress bars
xlab	if pl=TRUE, is x-axis label. Default is label(x) or name of calling argument
ylab	if pl=TRUE, is y-axis label. Default is constructed from u and time units
xlim	range of x axis
ylim	range of y axis

lty	line time for primary line connecting estimates
add	set to TRUE if adding to an existing plot
cex.subtitle	character size for subtitle. Default is .7. Use FALSE to suppress subtitle.
ab	TRUE to add a 45 degree line
...	plotting parameters to pass to the plot and errbar functions

Details

Function to divide a continuous variable x (e.g. age, or predicted cumulative incidence at time u created by predict.cmprsk into g quantile groups, get cumulative incidence estimates at time u (a scaler), and to return a matrix with columns x=mean x in quantile, n=number of subjects, events=no. events, and ci= cumulattive incidence at time u, std.err = standard error. Instead of supplying g, the user can supply the minimum number of subjects to have in the quantile group (m, default=50). If cuts is given (e.g. cuts=c(0,.1,.2,...{},.9,.1)), it overrides m and g.

Value

matrix with columns named x (mean predictor value in interval), n (sample size in interval), events (number of events in interval), ci (cumulative incidence estimate), std.err (standard error of cumulative incidence)

Note

This function is adapted from Harrell's function.

Author(s)

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

See Also

cuminc,pred.ci

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)

## ten fold cross validation
```

```
prostate.dat$preds.tenf.cv.prostate.crr.120 <-
                                        tenf.crr(prostate.crr,time = 120)

with(prostate.dat,
     groupci(preds.tenf.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 predictor labels for a model fit*

Description

Change labels of predictors

Usage

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

Arguments

`fit`	a model fit
`labels`	a character vector specifying new labels for variables in a fit. 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 random order or for a subset of the variables, e.g., labels=c(age="Age in Years",chol="Cholesterol")
`...`	other arguments

Details

This method function was written for competing risks regression model for facilitating to change the labels of predictors when construct a nomogram. It is used for the generic function `Newlabels`

Value

a new model fit object with the levels adjusted.

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)
prostate.g <- Newlabels(
    prostate.crr,
    c(
        TX = 'Treatment options',
        BX_GLSN_CAT = 'Biopsy Gleason Score Sum',
        CLIN_STG = 'Clinical stage'
    )
)
```

Newlevels.cmprsk *Change levels of categorical variable for a model fit*

Description

##' Change levels of categorical variable

Usage

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

Arguments

fit	a model fit
levels	a list of named vectors specifying new level labels for categorical predictors. This will override parms as well as datadist information (if available) that were stored with the fit.
...	other arguments

Details

This method function was written for competing risks regression model for facilitating to change the levels of categorical predictors when construct a nomogram. It is used for the generic function Newlevels

Value

returns a new model fit object with the levels adjusted.

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)
prostate.g <- Newlevels(prostate.crr,
    list(TX=c('Treatment 1','Treatment 2', 'Treatment 3')))
```

`nomo2.crr` *Estimate Cumulative Incidence Rate*

Description

Calculate Estimated Cumulative Incidence Rate Calculate predicted cumulative incidence rate based on a competing risks regression model.

Usage

```
nomo2.crr(x, f.crr, time)
```

Arguments

`x`	a vector of sum of linear predictors for each subject.
`f.crr`	a saved model fitted by function `crr.fit`
`time`	expected evaluation time

Details

This function is usually used to transform regular failure probabilities to competing risks adjusted probabilities, when a nomogram of competing risks regression model is constructed started from a regular survival model. It is not often called externally.

Value

a vector with each element being the predicted cumulative incidence rate at the expected time.

Note

internal function

Author(s)

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

See Also

pred2.crr crr.fit crr

---

nomogram.crr *Draw a Nomogram*

Description

Draws a partial nomogram that can be used to manually obtain predicted values from a regression model that was fitted with rms in effect.

Usage

```
nomogram.crr(
  fit,
  failtime = NULL,
  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 competing risks regression model fit that was created with function crr.fit.
failtime	the expected failure time for calculating cumalative incidence.
ci	logical flag to output cumulative incidence or event free probability if setting FALSE.
...	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.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 ith 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.

Details

The nomogram does not have lines representing sums, but it has a reference line for reading scoring points (default range 0–100). Once the reader manually totals the points, the predicted values can be read at the bottom. Non-monotonic transformations of continuous variables are handled (scales wrap around), as are transformations which have flat sections (tick marks are labeled with ranges).

Value

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

Note

internal use only

Author(s)

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

Draw a Competing Risks Nomogram

Draws a partial nomogram adjusting for competing risks for a cox ph survival model.

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

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, crr.fit, pred2.crr, nomo2.crr

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 *Draw a Nomogram with modified function* Nomogram

---

Description

Draw a Nomogram from a regression model

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 fit 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

Details

a modified version of nomogram in rms package

Value

a nomogram object

Note

internal use only. please reference to nomogram details.

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

rms, 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 *Calculate Cumulative Incidence*

Description

Calculate Cumulative Incidence

Usage

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

Arguments

cum	a object from function cuminc
tm1	expected failure time
failcode	value indicating the event of interest

Details

Extract cumulative incidence and its variance from a object generated from function cuminc.

Value

a data frame with 3 columns.

column 3:	Group name.
column 2:	Cumulative Incidence Probability.
column 3:	Variance

Author(s)

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

See Also

cuminc

Examples

```
data(prostate.dat) # get demo data set
cum <- cuminc(prostate.dat$TIME_EVENT,prostate.dat$EVENT_DOD,
              cencode = 0)
# calculate the expected cumulative incidence by 5 year for death from
# prostate cancer
# Here, code for cause A is 'DOA'.
pred.ci(cum,60,failcode = 1)
```

pred2.crr *Predict Cumulative Incidence Rate*

Description

Predict Cumulative Incidence Rate

Usage

```
pred2.crr(f.crr, lp, time)
```

Arguments

f.crr	a saved competing risks regression model created by function crr.fit
lp	a scalar being the sum of linear predictors for a single subject.
time	expected time point, at which cumulative incidence rate will be assessed.

Details

Calculate the predicted cumulative incidence rate based on a saved competing risks regression model. The cumulative incidence is adjusted for other competing causes rather than the event of interest.

Value

Return the predicted cumulative incidence rate.

Author(s)

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

See Also

predict.crr crr.fit

`pred3.crr` *Predict cumulative incidence used internally*

Description

Predict cumulative incidence used internally an internal function.

Usage

```
pred3.crr(z, cov1, cov2, time, lps = FALSE)
```

Arguments

`z`	the fitter crr model
`cov1`	covarite matrix 1
`cov2`	covariate matrix 2
`time`	time point at which the prediction will make
`lps`	logical flag. if the liner predictor be generated.

Details

Internally used only.

Value

A list. See `crr` for details.

Note

an internal function called by `crr.fit`.

Author(s)

changhong

predict.cmprsk *Calculate Predicted Competing Risks Probability*

Description

Calculate Predicted Competing Risks Probability.

Usage

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

Arguments

object	a saved crr model fit crated by function crr.fit
newdata	data frame for prediction. Each row of the data frame contains values of covariates that are required in the crr model. If missing, the original data set that was used to develop the crr model will be used for prediction.
time	expected time point for evaluating the competing risks probability.
lps	set TRUE to return linear predictor values instead of failure probabilities.
...	other arguments

Details

Calculate predicted probabilities for a competing risks regression model, which is fitted by function crr.fit.

Value

A vector with the length equal to the number of rows in the data frame, which was used to make prediction. Each element corresponds to a predicted failure probability at the expected time point.

Note

This function is adapted from function predict.crr in package cmprsk.

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

```
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)
prostate.dat$pred.60 <- predict(prostate.crr, time=60)
```

---

predictDesign *Internal function to Calculate Predicted Competing Risks Probability.*

Description

Internal function to Calculate Predicted Competing Risks Probability.

Usage

```
predictDesign(
  fit,
  newdata = NULL,
  type = c("lp", "x", "data.frame", "terms", "adjto", "adjto.data.frame",
    "model.frame"),
  se.fit = FALSE,
  conf.int = FALSE,
  conf.type = c("mean", "individual"),
  incl.non.slopes = NULL,
  non.slopes = NULL,
  kint = 1,
  na.action = na.keep,
  expand.na = TRUE,
  center.terms = TRUE,
  ...
)
```

Arguments

fit	a saved crr model fit crated by function
newdata	data frame for prediction. Each row of the data frame contains values of covariates that are required in the crr model. If missing, the original data set that was used to develop the crr model will be used for prediction.
type	type of output desired
se.fit	return standard error
conf.int	specify conf.int as a positive fraction to obtain upper and lower confidence intervals (e.g., 'conf.int=0.95').
conf.type	specifies the type of confidence interval.
incl.non.slopes	set to 'TRUE' to include multiple intercepts, default is 'NULL'
non.slopes	a vector of multiple intercepts
kint	a single integer specifying the number of the intercept to use in multiple-intercept models.
na.action	Function to handle missing values in 'newdata'.
expand.na	set to 'FALSE' to keep the naresid from having any effect
center.terms	set to 'FALSE' to suppress subtracting adjust-to values from columns of the design matrix before computing terms with 'type="terms"'.
...	ignored crr.fit

Details

Calculate predicted probabilities for a competing risks regression model

Value

A vector with the length equal to the number of rows in the data frame, which was used to make prediction. Each element corresponds to a predicted failure probability at the expected time point.

See Also

predictrms,

---

prostate.dat *Prostate cancer data set*

Description

This is an artificial prostate cancer dataset used for illustrating the usages of functions in R package and QHScrnomo

Format

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 data set.

Examples

```
data(prostate.dat)
```

---

sas.cmprsk *generate prediction equation for a competing risks regression models*

Description

Generate an equation to calculate X beta from a crr model fit If specify a time point, the function also generates the subcumulative rate at the time point.

Usage

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

Arguments

`f`	a model fit from the competing risks regression.
`time`	time point
`baseonly`	logical variable. If true, only base survival probability will be printed.
`file`	A connection, or a character string naming the file to print to
`append`	logical. Only used if the argument file is the name of file. If TRUE output will be appended to file; otherwise, it will overwrite the content of file

Details

f should be fitted by the function crr.fit

Value

out	a character vector that can be output as a formula by function cat
Rout	same as out except replacing "max","min","=" and "**" with "pmax","pmin", "==" and ^ respectively so that the formula can be pasted to R session and compute X beta directly without any further modification

Author(s)

changhong

See Also

sascode,Function, 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)
sas.cmprsk(prostate.crr, time = 60)
```

---

summary.cmprsk *summary for competing risks regression*

Description

generate summary information

Usage

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

Arguments

object	a crr model object from function crr.fit
...	other parameters

Details

summarize a crr regression model

Value

a matrix

Author(s)

changhong

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)
summary(prostate.crr)
```

---

tenf.crr *Ten fold cross validation for competing risks regression*

Description

Ten fold cross validation for crr endpoint

Usage

```
tenf.crr(fit, time = NA, lps = FALSE, fold = 10)
```

Arguments

fit	a competing risks regression model fittd by function crr.fit.
time	the expected time point.
lps	logical flag. If true, values of predicted X beta will be output instead of cumulative incidence
fold	number of fold. the default is 10 fold cross validation.

Details

Do cross validation on a competing risk regression model.

Value

A vector of predicted values of cumulative incidence or X beta for each observation.

Note

Before the function is called, packages 'Hmisc', 'rms' and 'cmprsk' should be loaded as the function will call some funcitons in these packages.

Author(s)

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

See Also

crr.fit, crr

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)

## ten fold cross validation
prostate.dat$preds.tenf<-
    tenf.crr(prostate.crr,time = 120)
```

Index