Package: cmprsk (via r-universe)

September 17, 2024

Version 2.2-12

Date 2024-05-14

Title Subdistribution Analysis of Competing Risks

Author Bob Gray <gray@jimmy.harvard.edu>

Maintainer Bob Gray <gray@jimmy.harvard.edu>

Depends R (>= 3.0.0), survival

Description Estimation, testing and regression modeling of subdistribution functions in competing risks, as described in Gray (1988), A class of K-sample tests for comparing the cumulative incidence of a competing risk, Ann. Stat. 16:1141-1154 <DOI:10.1214/aos/1176350951>, and Fine JP and Gray RJ (1999), A proportional hazards model for the subdistribution of a competing risk, JASA, 94:496-509, <DOI:10.1080/01621459.1999.10474144>.

URL https://www.R-project.org

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-05-19 14:50:02 UTC

Contents

crr 2
cuminc 4
plot.cuminc 6
plot.predict.crr 7
predict.crr 8
print.crr 9
print.cuminc 9
summary.crr 10
timepoints 11
[.cuminc 12

Index 13

crr *Competing Risks Regression*

Description

regression modeling of subdistribution functions in competing risks

Usage

```
crr(ftime, fstatus, cov1, cov2, tf, cengroup, failcode=1, cencode=0,
subset, na.action=na.omit, gtol=1e-06, maxiter=10, init, variance=TRUE)
```

Arguments

`ftime`	vector of failure/censoring times
`fstatus`	vector with a unique code for each failure type and a separate code for censored observations
`cov1`	matrix (nobs x ncovs) of fixed covariates (either cov1, cov2, or both are required)
`cov2`	matrix of covariates that will be multiplied by functions of time; if used, often these covariates would also appear in cov1 to give a prop hazards effect plus a time interaction
`tf`	functions of time. A function that takes a vector of times as an argument and returns a matrix whose jth column is the value of the time function corresponding to the jth column of cov2 evaluated at the input time vector. At time `tk`, the model includes the term `cov2[,j]*tf(tk)[,j]` as a covariate.
`cengroup`	vector with different values for each group with a distinct censoring distribution (the censoring distribution is estimated separately within these groups). All data in one group, if missing.
`failcode`	code of fstatus that denotes the failure type of interest
`cencode`	code of fstatus that denotes censored observations
`subset`	a logical vector specifying a subset of cases to include in the analysis
`na.action`	a function specifying the action to take for any cases missing any of ftime, fstatus, cov1, cov2, cengroup, or subset.
`gtol`	iteration stops when a function of the gradient is < `gtol`
`maxiter`	maximum number of iterations in Newton algorithm (0 computes scores and var at `init`, but performs no iterations)
`init`	initial values of regression parameters (default=all 0)
`variance`	If `FALSE`, then suppresses computation of the variance estimate and residuals

Details

Fits the 'proportional subdistribution hazards' regression model described in Fine and Gray (1999). This model directly assesses the effect of covariates on the subdistribution of a particular type of failure in a competing risks setting. The method implemented here is described in the paper as the weighted estimating equation.

While the use of model formulas is not supported, the `model.matrix` function can be used to generate suitable matrices of covariates from factors, eg `model.matrix(~factor1+factor2)[,-1]` will generate the variables for the factor coding of the factors `factor1` and `factor2`. The final `[,-1]` removes the constant term from the output of `model.matrix`.

The basic model assumes the subdistribution with covariates z is a constant shift on the complementary log log scale from a baseline subdistribution function. This can be generalized by including interactions of z with functions of time to allow the magnitude of the shift to change with follow-up time, through the cov2 and tfs arguments. For example, if z is a vector of covariate values, and uft is a vector containing the unique failure times for failures of the type of interest (sorted in ascending order), then the coefficients a, b and c in the quadratic (in time) model $az + bzt + zt^2$ can be fit by specifying `cov1=z`, `cov2=cbind(z,z)`, `tf=function(uft) cbind(uft,uft*uft)`.

This function uses an estimate of the survivor function of the censoring distribution to reweight contributions to the risk sets for failures from competing causes. In a generalization of the methodology in the paper, the censoring distribution can be estimated separately within strata defined by the cengroup argument. If the censoring distribution is different within groups defined by covariates in the model, then validity of the method requires using separate estimates of the censoring distribution within those groups.

The residuals returned are analogous to the Schoenfeld residuals in ordinary survival models. Plotting the jth column of res against the vector of unique failure times checks for lack of fit over time in the corresponding covariate (column of cov1).

If `variance=FALSE`, then some of the functionality in `summary.crr` and `print.crr` will be lost. This option can be useful in situations where crr is called repeatedly for point estimates, but standard errors are not required, such as in some approaches to stepwise model selection.

Value

Returns a list of class crr, with components

`$coef`	the estimated regression coefficients
`$loglik`	log pseudo-liklihood evaluated at `coef`
`$score`	derivitives of the log pseudo-likelihood evaluated at `coef`
`$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
$call	The call to crr
$n	The number of observations used in fitting the model
$n.missing	The number of observations removed from the input data due to missing values
$loglik.null	The value of the log pseudo-likelihood when all the coefficients are 0
$invinf	- inverse of second derivative matrix of the log pseudo-likelihood

References

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

See Also

predict.crr print.crr plot.predict.crr summary.crr

Examples

```
# simulated data to test
set.seed(10)
ftime <- rexp(200)
fstatus <- sample(0:2,200,replace=TRUE)
cov <- matrix(runif(600),nrow=200)
dimnames(cov)[[2]] <- c('x1','x2','x3')
print(z <- crr(ftime,fstatus,cov))
summary(z)
z.p <- predict(z,rbind(c(.1,.5,.8),c(.1,.5,.2)))
plot(z.p,lty=1,color=2:3)
crr(ftime,fstatus,cov,failcode=2)
# quadratic in time for first cov
crr(ftime,fstatus,cov,cbind(cov[,1],cov[,1]),function(Uft) cbind(Uft,Uft^2))
#additional examples in test.R
```

---

cuminc *Cumulative Incidence Analysis*

Description

Estimate cumulative incidence functions from competing risks data and test equality across groups

Usage

```
cuminc(ftime, fstatus, group, strata, rho=0, cencode=0,
subset, na.action=na.omit)
```

Arguments

`ftime`	failure time variable
`fstatus`	variable with distinct codes for different causes of failure and also a distinct code for censored observations
`group`	estimates will calculated within groups given by distinct values of this variable. Tests will compare these groups. If missing then treated as all one group (no test statistics)
`strata`	stratification variable. Has no effect on estimates. Tests will be stratified on this variable. (all data in 1 stratum, if missing)
`rho`	Power of the weight function used in the tests.
`cencode`	value of fstatus variable which indicates the failure time is censored.
`subset`	a logical vector specifying a subset of cases to include in the analysis
`na.action`	a function specifying the action to take for any cases missing any of ftime, fstatus, group, strata, or subset.

Value

A list with components giving the subdistribution estimates for each cause in each group, and a component `Tests` giving the test statistics and p-values for comparing the subdistribution for each cause across groups (if the number of groups is >1). The components giving the estimates have names that are a combination of the group name and the cause code. These components are also lists, with components

`time`	the times where the estimates are calculated
`est`	the estimated sub-distribution functions. These are step functions (all corners of the steps given), so they can be plotted using ordinary lines() commands. Estimates at particular times can be located using the timepoints() function.
`var`	the estimated variance of the estimates, which are estimates of the asymptotic variance of Aalen (1978).

Author(s)

Robert Gray

References

Gray RJ (1988) A class of K-sample tests for comparing the cumulative incidence of a competing risk, ANNALS OF STATISTICS, 16:1141-1154.

Kalbfleisch and Prentice (1980) THE ANALYSIS OF FAILURE TIME DATA, p 168-9.

Aalen, O. (1978) Nonparametric estimation of partial transition probabilities in multiple decrement models, ANNALS OF STATISTICS, 6:534-545.

See Also

`plot.cuminc` `timepoints` `print.cuminc`

Examples

```
set.seed(2)
ss <- rexp(100)
gg <- factor(sample(1:3,100,replace=TRUE),1:3,c('a','b','c'))
cc <- sample(0:2,100,replace=TRUE)
strt <- sample(1:2,100,replace=TRUE)
print(xx <- cuminc(ss,cc,gg,strt))
plot(xx,lty=1,color=1:6)
# see also test.R, test.out
```

---

plot.cuminc *Create Labeled Cumulative Incidence Plots*

Description

Plot method for cuminc. Creates labeled line plots from appropriate list input, for example, the output from `cuminc()`.

Usage

```
## S3 method for class 'cuminc'
plot(x, main=" ", curvlab, ylim=c(0, 1), xlim, wh=2,
xlab="Years",  ylab="Probability", lty=1:length(x), color=1, lwd=par('lwd'),
...)
```

Arguments

`x`	a list, with each component representing one curve in the plot. Each component of `x` is itself a list whose first component gives the x values and 2nd component the `y` values to be plotted. Although written for cumulative incidence curves, can in principle be used for any set of lines.
`main`	the main title for the plot.
`curvlab`	Curve labels for the plot. Default is `names(x)`, or if that is missing, `1:nc`, where `nc` is the number of curves in `x`.
`ylim`	yaxis limits for plot
`xlim`	xaxis limits for plot (default is 0 to the largest time in any of the curves)
`wh`	if a vector of length 2, then the upper right coordinates of the legend; otherwise the legend is placed in the upper right corner of the plot
`xlab`	X axis label
`ylab`	y axis label
`lty`	vector of line types. Default `1:nc` (`nc` is the number of curves in `x`). For color displays, `lty=1, color=1:nc`, might be more appropriate. If `length(lty)<nc`, then `lty[1]` is used for all.
`color`	vector of colors. If `length(color)<nc`, then the `color[1]` is used for all.
`lwd`	vector of line widths. If `length(lwd)<nc`, then `lwd[1]` is used for all.
`...`	additional arguments passed to the initial call of the plot function.

Value

No value is returned.

See Also

cuminc

---

plot.predict.crr — *Plot estimated subdistribution functions*

Description

plot method for `predict.crr`

Usage

```
## S3 method for class 'predict.crr'
plot(x, lty=1:(ncol(x)-1), color=1,
ylim=c(0, max(x[, -1])), xmin=0, xmax=max(x[, 1]), ...)
```

Arguments

`x`	Output from `predict.crr`
`lty`	vector of line types. If length is $<$ number of curves, then `lty[1]` is used for all.
`color`	vector of line colors. If length is $<$ number of curves, then `color[1]` is used for all.
`ylim`	range of y-axis (vector of length two)
`xmin`	lower limit of x-axis (often 0, the default)
`xmax`	upper limit of x-axis
`...`	Other arguments to plot

Side Effects

plots the subdistribution functions estimated by `predict.crr`, by default using a different line type for each curve

See Also

crr predict.crr

`predict.crr` *Estimate subdistribution functions from crr output*

Description

predict method for crr

Usage

```
## S3 method for class 'crr'
predict(object, cov1, cov2, ...)
```

Arguments

`object`	output from crr
`cov1, cov2`	each row of cov1 and cov2 is a set of covariate values where the subdistribution should be estimated. The columns of cov1 and cov2 must be in the same order as in the original call to crr. Each must be given if present in the original call to crr.
`...`	additional arguments are ignored (included for compatibility with generic).

Details

Computes $1-\exp(-B(t))$, where $B(t)$ is the estimated cumulative sub-distribution hazard obtained for the specified covariate values, obtained from the Breslow-type estimate of the underlying hazard and the estimated regression coefficients.

Value

Returns a matrix with the unique type 1 failure times in the first column, and the other columns giving the estimated subdistribution function corresponding to the covariate combinations in the rows of cov1 and cov2, at each failure time (the value that the estimate jumps to at that failure time).

See Also

`crr` `plot.predict.crr`

`print.crr` *prints summary of a crr object*

Description

print method for crr objects

Usage

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

Arguments

`x`	crr object (output from `crr()`)
`...`	additional arguments to `print()`

Details

prints the convergence status, the estimated coefficients, the estimated standard errors, and the two-sided p-values for the test of the individual coefficients equal to 0. (If convergence is false everything else may be meaningless.)

See Also

`crr`

`print.cuminc` *Print cuminc objects*

Description

A print method for objects of class cuminc (output from `cuminc()`).

Usage

```
## S3 method for class 'cuminc'
print(x, ntp=4, maxtime, ...)
```

Arguments

`x`	an object of class cuminc
`ntp`	number of timepoints where estimates are printed
`maxtime`	the maximum timepoint where values are printed. The default is the maximum time in the curves in `x`
`...`	additional arguments to `print()`

Details

Prints the test statistics and p-values (if present in x), and for each estimated cumulative incidence curve prints its value and estimated variance at a vector of times. The times are chosen between 0 and maxtime using the pretty() function.

Author(s)

Robert Gray

See Also

cuminc

---

summary.crr — *Summary method for crr*

Description

Generate and print summaries of crr output

Usage

```
## S3 method for class 'crr'
summary(object, conf.int = 0.95, digits =
max(options()$digits - 5, 2), ...)

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

Arguments

object	An object of class crr (output from the crr function)
conf.int	the level for a two-sided confidence interval on the coeficients. Default is 0.95.
digits	In summary.crr, digits determines the number of significant digits retained in the p-values. In print.summary.crr, digits sets the values of the digits option for printing the output.
...	Included for compatibility with the generic functions. Not currently used.
x	An object of class summary.crr (output from the summary method for crr)

Details

The summary method calculates the standard errors, subdistribution hazard ratios z-scores, p-values, and confidence intervals on the hazard ratios. The print method prints a fairly standard format tabular summary of the results.

The pseudo likelihood ratio test in the printed output is based on the difference in the objective function at the global null and at the final estimates. Since this objective function is not a true likelihood, this test statistic is not asymptotically chi-square.

Value

`summary.crr` returns a list of class summary.crr, which contains components

`call`	The call to crr
`converged`	TRUE if the iterative algorithm converged
`n`	The number of observations used in fitting the model
`n.missing`	The number of observations removed by `crr` from the input data due to missing values
`loglik`	The value of the negative of the objective function (the pseudo log likelihood at convergence
`coef`	A matrix giving the estimated coefficients, hazard ratios, standard errors, z-scores, and p-values
`conf.int`	A matrix giving the estimated hazard ratios, inverse hazard ratios and lower and upper confidence limits on the hazard ratios
`logtest`	Twice the difference in log pseudo likelihood values

Author(s)

The summary and print.summary methods were provided by Luca Scrucca

See Also

`crr`

Examples

```
## see examples in the crr help file
```

`timepoints` *Calculate Estimates at Specific Timepoints*

Description

Find values at specified timepoints from curves specified as all corners of step functions.

Usage

```
timepoints(w, times)
```

Arguments

`w`	a list containing the estimates, with points for all corners of the step function. (Usually created by cuminc.) Each component in the list contains the estimate for a different group. Each component has components giving times, function estimates, and variances (see cuminc)
`times`	vector of times where estimates are needed

Value

A list with components

`$est`	a matrix of estimates of the subdistributions with a row for each component in `w` and a column for each time
`$var`	a matrix giving the corresponding variances.

See Also

`cuminc`

`[.cuminc` *Subset method for lists of class cuminc*

Description

A subset method that preserves the class of objects of class cuminc, allowing a subset of the curves to be selected.

Usage

```
## S3 method for class 'cuminc'
x[i,...]
```

Arguments

`x`	object of class cuminc
`i`	elements to extract
`...`	not used

Value

A list with selected components of `x`, with the class set to cuminc so cuminc methods can be applied.

See Also

`cuminc` `plot.cuminc` `print.cuminc`

Index

∗ **hplot**
plot.cuminc, 6
∗ **survival**
[.cuminc, 12
crr, 2
cuminc, 4
plot.cuminc, 6
plot.predict.crr, 7
predict.crr, 8
print.crr, 9
print.cuminc, 9
summary.crr, 10
timepoints, 11
[.cuminc, 12

crr, 2, *7–9*, *11*
cuminc, 4, *7*, *10*, *12*

plot.cuminc, *5*, 6, *12*
plot.predict.crr, *4*, 7, *8*
predict.crr, *4*, *7*, 8
print.crr, *4*, 9
print.cuminc, *5*, 9, *12*
print.summary.crr *(*summary.crr*)*, 10

summary.crr, *4*, 10

timepoints, *5*, 11