Package 'risksetROC'

July 23, 2025

Version 1.0.4.1

Date 2012-04-13

Title Riskset ROC Curve Estimation from Censored Survival Data

Author Patrick J. Heagerty <heagerty@u.washington.edu>, packaging by
Paramita Saha-Chaudhuri <paramita.sahachaudhuri.work@gmail.com>

Maintainer Paramita Saha-Chaudhuri

<paramita.sahachaudhuri.work@gmail.com>

Depends R (>= 2.10), survival, MASS

Description Compute time-dependent Incident/dynamic accuracy measures
(ROC curve, AUC, integrated AUC)from censored survival data
under proportional or non-proportional hazard assumption of
Heagerty & Zheng (Biometrics, Vol 61 No 1, 2005, PP 92-105).

License GPL (>= 2)

Repository CRAN

Date/Publication 2022-06-13 08:14:56 UTC

NeedsCompilation no

Contents

CoxWeights *Incident/Dynamic (I/D) ROC curve, AUC and integrated AUC (iAUC) estimation of censored survival data*

Description

This function estimates of TP and FP based on a Cox model as discussed in Heagerty and Zheng, 2005, for incident/dynamic ROC curve. TP is estimated as Equation (1) and FP is estimated as Equation (2) of the paper.

Usage

```
CoxWeights(marker, Stime, status, predict.time, entry)
```

Arguments

marker	estimated linear predictor from a set of covariates. Note that this linear predictor can be obtained from any model.
Stime	For right censored data, this is the follow up time. For left truncated data, this is the ending time for the interval.
status	Indicator of status, 1 if death or event, 0 otherwise.
predict.time	Time point of the ROC curve.
entry	For left truncated data, this is the entry time of the interval. The default is set to NULL for right censored data.

Details

Suppose we have censored survival data (right censored or both left-truncated and right censored data) along with a marker value and we want to see how well the marker predicts the survival time for the subjects in the dataset using Incident/dynamic definition of ROC curve. In particular, suppose we have survival times in days and we want to see how well the marker predicts the one-year survival (predict.time=365 days). This function CoxWeights(), returns the unique marker values, TP (True Positive), FP (False Positive) and AUC (Area under (ROC) curve) corresponding to the time point of interest (predict.time). Note that the linear predictor *marker* can be obtained from any model, specifically, the survival model may be based on either a PH or a time-varying Cox model.

Value

Returns a list of the following items:

eta	unique marker values for calculation of TP and FP
TP	True Positive values corresponding to unique marker values
FP	False Positive values corresponding to unique marker values
AUC	Area Under (ROC) Curve at time predict.time

Author(s)

Patrick J. Heagerty

References

Heagerty, P.J., Zheng Y. (2005) Survival Model Predictive Accuracy and ROC curves *Biometrics*, **61**, 92 – 105

Examples

```
library(MASS)
data(VA)
survival.time <- VA$stime
survival.status <- VA$status
score <- VA$Karn
cell.type <- factor(VA$cell )
tx <- as.integer( VA$treat==1 )
age <- VA$age
survival.status[VA$stime > 500 ] <- 0
survival.time[VA$stime > 500 ] <- 500
library(survival)
fit0 <- coxph( Surv(survival.time,survival.status)
        ~ score + cell.type + tx + age, na.action=na.omit )
summary(fit0)
eta <- fit0$linear.predictor
AUC <- NULL
out <- CoxWeights(marker=eta, Stime=survival.time, status=survival.status,
predict.time=30)
## to see how well the marker predicts one-month survival
AUC <- out$AUC
```

`IntegrateAUC` *Incident/Dynamic (I/D) ROC curve, AUC and integrated AUC (iAUC) estimation of censored survival data*

Description

This function integrates AUC using weights $w(t) = 2 * f(t) * S(t)$ as discussed in Heagerty and Zheng, 2005.

Usage

```
IntegrateAUC(AUC, utimes, St, tmax, weight="rescale")
```

Arguments

AUC	Area under ROC curve at utimes
utimes	Unique event times for subjects
St	Estimated survival probability at utimes
tmax	Maximum time length to be considered
weight	Either of "rescale" or "conditional"

Details

This function estimates time-dependent concordance measure

$$P(M_i > M_j | T_i < t, T_i < tmax, T_j > t)$$

as discussed in the paper from AUC and weights derived from the survival time distribution. The concordance measure is estimated under the assumption that smaller of the two event times happened before time *tmax*. The resulting measure is an weighted sum of estimated AUC at each unique failure time where weights are proportional to $2 * f(t) * S(t)$, and T is failure time of interest. If weight="rescale", then the weights are rescaled so that the sum of the weights is one. If weight="conditional", it is assumed that both the events happened before *tmax*.

Value

Returns the following item:

iAUC	Integrated AUC using w(t) as above as weights

Author(s)

Patrick J. Heagerty

References

Heagerty, P.J., Zheng Y. (2005) Survival Model Predictive Accuracy and ROC curves *Biometrics*, **61**, 92 – 105

Examples

```
library(MASS)
data(VA)
survival.time <- VA$stime
survival.status <- VA$status
score <- VA$Karn
cell.type <- factor(VA$cell )
tx <- as.integer( VA$treat==1 )
age <- VA$age
survival.status[VA$stime > 500 ] <- 0
survival.time[VA$stime > 500 ] <- 500
library(survival)
## first find the estimated survival probabilities at unique failure times
surv.prob <- unique(survfit(Surv(survival.time,survival.status)~1)$surv)
```

```
fit0 <- coxph( Surv(survival.time,survival.status)
        ~ score + cell.type + tx + age, na.action=na.omit )
eta <- fit0$linear.predictor
model.score <- eta

utimes <- unique( survival.time[ survival.status == 1 ] )
utimes <- utimes[ order(utimes) ]

## find AUC at unique failure times
AUC <- rep( NA, length(utimes) )
for( j in 1:length(utimes) )
{
out <- CoxWeights( eta, survival.time, survival.status,utimes[j])
AUC[j] <- out$AUC
}
## integrated AUC to get concordance measure
iAUC <- IntegrateAUC( AUC, utimes, surv.prob, tmax=365 )
```

KM.plot *Incident/Dynamic (I/D) ROC curve, AUC and integrated AUC (iAUC) estimation of censored survival data*

Description

This function creates Kaplan-Meier plot.

Usage

```
KM.plot(Stime, survival, max.T=NULL, lty=NULL, all=TRUE, ...)
```

Arguments

Stime	unique ordered event times
survival	estimates of survival probabilities at *Stime*
max.T	maximum time to be considered for plotting, default is NULL which plots survival till max(Stime)+1 units
lty	line type
all	TRUE or FALSE, default is TRUE
...	additional plot arguments

Details

This function creates Kaplan-Meier plot. If *all=TRUE*, then this creates a new plot. If *all=FALSE*, it adds line to an existing plot and hence must be called after a plot() or similar call.

Author(s)

Patrick J. Heagerty

References

Heagerty, P.J., Zheng Y. (2005) Survival Model Predictive Accuracy and ROC curves *Biometrics*, **61**, 92 – 105

Examples

```
data(pbc)
## considering only randomized patients
pbc1 <- pbc[1:312,]
## create new censoring variable combine 0,1 as 0, 2 as 1
survival.status <- ifelse( pbc1$status==2, 1, 0)
survival.time <- pbc1$fudays
kout <- weightedKM(Stime=survival.time, status=survival.status)
KM.plot(kout$time,kout$survival)
```

`llCoxReg` *Incident/Dynamic (I/D) ROC curve, AUC and integrated AUC (iAUC) estimation of censored survival data*

Description

This function estimates the time-varying parameter estimate $\beta(t)$ of non-proportional hazard model using local-linear Cox regression as discussed in Heagerty and Zheng, 2005.

Usage

```
llCoxReg(Stime, entry=NULL, status, marker, span=0.40, p=1, window="asymmetric")
```

Arguments

`Stime`	For right censored data, this is the follow up time. For left truncated data, this is the ending time for the interval.
`entry`	For left truncated data, this is the entry time of the interval. The default is set to NULL for right censored data.
`status`	Survival status.
`marker`	Marker value.
`span`	bandwidth parameter that controls the size of a local neighborhood.
`p`	1 if only the time-varying coefficient is of interest and 2 if the derivative of time-varying coefficient is also of interest, default is 1
`window`	Either of "asymmetric" or "symmetric", default is asymmetric.

Details

This function calculates the parameter estimate $\beta(t)$ of non-proportional hazard model using local-linear Cox regression as discussed in Heagerty and Zheng, 2005. This estimation is based on a time-dependent Cox model (Cai and Sun, 2003). For *p=1*, the return item *beta* has two columns, the first column is the time-varying parameter estimate, while the second column is the derivative. However, if the derivative of the time-varying parameter is of interest, then we suggest to use *p=2*. In this case, *beta* has four columns, the first two columns are the same when *p=1*, while the last two columns estimates the coefficients of squared marker value and its derivative.

Value

Returns a list of following items:

`time`	unique failure times
`beta`	estimate of time-varying parameter $\beta(t)$ at each unique failure time.

Author(s)

Patrick J. Heagerty

References

Heagerty, P.J., Zheng Y. (2005) Survival Model Predictive Accuracy and ROC curves *Biometrics*, **61**, 92 – 105

Examples

```
data(pbc)
## considering only randomized patients
pbc1 <- pbc[1:312,]
## create new censoring variable combine 0,1 as 0, 2 as 1
survival.status <- ifelse( pbc1$status==2, 1, 0)
survival.time <- pbc1$fudays
pbc1$status1 <- survival.status
fit <- coxph( Surv(fudays,status1) ~ log(bili) +
                                     log(protime) +
                                     edema +
                                     albumin +
                                     age,
              data=pbc1 )
eta5 <- fit$linear.predictors
x <- eta5
nobs <- length(survival.time[survival.status==1])
span <- 1.0*(nobs^(-0.2))

## Not run:
bfnx1 <- llCoxReg(Stime=survival.time, status=survival.status, marker=x,
                  span=span, p=1)
plot(bfnx1$time, bfnx1$beta[,1], type="l", xlab="Time", ylab="beta(t)")

## End(Not run)
```

pbc *Incident/Dynamic (I/D) ROC curve, AUC and integrated AUC (iAUC) estimation of censored survival data*

Description

This is Mayo PBC data as obtained from the website: http://lib.stat.cmu.edu/datasets/pbc

Format

A data frame with 418 observations and 20 variables: id (patient id), fudays (follow-up days, number of days between registration and the earlier of death, transplantation, or study analysis time in July, 1986), status (survival status), drug (1 = D-penicillamine, 2 = placebo) age (age in days), sex (0 = male, 1 = female), ascites (presence of asictes: 0=no 1=yes), hepatom (presence of hepatomegaly: 0=no 1=yes), spiders (presence of spiders: 0=no 1=yes), edema (presence of edema: 0=no edema and no diuretic therapy for edema; .5 = edema present without diuretics, or edema resolved by diuretics; 1 = edema despite diuretic therapy), bili (serum bilirubin in mg/dl), chol (serum cholesterol in mg/dl), albumin (albumin in gm/dl), copper (urine copper in ug/day), alkphos (alkaline phosphatase in U/liter), sgot (SGOT in U/ml), trig (triglicerides in mg/dl), platelet (platelets per cubic ml / 1000), protime (prothrombin time in seconds), stage (histologic stage of disease)

Author(s)

Patrick J. Heagerty

References

Heagerty, P.J., Zheng Y. (2005) Survival Model Predictive Accuracy and ROC curves *Biometrics*, **61**, 92 – 105

Examples

```
library(MASS)
data(VA)
## need to order the data in ascending order of survival time
new.VA=VA[order(VA$stime),]
risket.VA=riskset(new.VA)
```

riskset *Incident/Dynamic (I/D) ROC curve, AUC and integrated AUC (iAUC) estimation of censored survival data*

Description

This function creates risk set at each unique failure time from a survival data set.

Usage

```
riskset(dat, entry=FALSE)
```

Arguments

dat	survival dataset with at least three variables: survival.times, survival.status and marker, in that order. The survival data set may have additional variables. In case of interval censored data, the first four columns are: entry time, exit time, status at exit and marker
entry	default is FALSE indicating right censored data. TRUE if left truncated data

Details

This function creates risk set at each unique failure time from a survival data set and is needed for llCoxReg(). The function can handle both right censored and interval censored data.

Value

Returns a new data set with columns as follows: start, finish, newStatus and other variables from the original dataset except survival time and status. The first two columns correspond to the start and end of time intervals considered and the newStatus corresponds to the survival status of the patient corresponding to this interval, i.e. the status is 1 if the patient had event during this interval (start, finish] and 0 otherwise. Note that the survival time need to be in ascending order.

Author(s)

Patrick J. Heagerty

References

Heagerty, P.J., Zheng Y. (2005) Survival Model Predictive Accuracy and ROC curves *Biometrics*, **61**, 92 – 105

Examples

```
library(MASS)
data(VA)
## need to order the data in ascending order of survival time
new.VA=VA[order(VA$stime),]
risket.VA=riskset(new.VA)
```

`risksetAUC` *Incident/Dynamic (I/D) ROC curve, AUC and integrated AUC (iAUC) estimation of censored survival data*

Description

This function creates risksetAUC from a survival data set

Usage

```
risksetAUC(Stime, entry=NULL, status, marker, method="Cox",
                        span=NULL, order=1, window="asymmetric",
                        tmax, weight="rescale", plot=TRUE, type="l",
                        xlab="Time", ylab="AUC", ...)
```

Arguments

`Stime`	For right censored data, this is the follow up time. For left truncated data, this is the ending time for the interval.
`entry`	For left truncated data, this is the entry time of the interval. The default is set to NULL for right censored data.
`status`	survival status, 1 if had an event and 0 otherwise
`marker`	marker
`method`	either of "Cox", "LocalCox" and "Schoenfeld", default is "Cox"
`span`	bandwidth parameter that controls the size of a local neighborhood, needed for *method="LocalCox"* or *method="Schoenfeld"*
`order`	0 or 1, locally mean if 0 and local linear if 1, needed for method="Schoenfeld", default is 1
`window`	either of "asymmetric" or "symmetric", default is asymmetric, needed for method="LocalCox"
`tmax`	maximum time to be considered for calculation of AUC
`weight`	either of "rescale" or "conditional". If *weight="rescale"*, then weights are rescaled so that the sum is unity. If *weight="conditional"* both the event times are assumed to be less than *tmax*
`plot`	TRUE or FALSE, default is TRUE
`type`	default is "l", can be either of "p" for points, "l" for line, "b" for both
`xlab`	label for x-axis
`ylab`	label for y-axis
`...`	additional plot arguments

Details

This function creates and plots AUC based on incident/dynamic definition of Heagerty, et. al. based on a survival data and marker values. If proportional hazard is assumed then method="Cox" can be used. In case of non-proportional hazard, either of "LocalCox" or "Schoenfeld" can be used. These two methods differ in how the smoothing is done. If *plot="TRUE"* then the AUC curve is plotted against time (till *tmax+1*). Additional plot arguments can be supplied.

Value

Returns a list of the following items:

`utimes`	ordered unique failure times
`St`	estimated survival probability at utimes
`AUC`	Area under ROC curve at utimes
`Cindex`	Cindex

Author(s)

Paramita Saha

References

Heagerty, P.J., Zheng Y. (2005) Survival Model Predictive Accuracy and ROC curves *Biometrics*, **61**, 92 – 105

See Also

IntegrateAUC(), weightedKM(), llCoxReg(), SchoenSmooth(), CoxWeights()

Examples

```
library(MASS)
data(VA)
survival.time=VA$stime
survival.status=VA$status
score <- VA$Karn
cell.type <- factor(VA$cell)
tx <- as.integer( VA$treat==1 )
age <- VA$age
survival.status[survival.time>500 ] <- 0
survival.time[survival.time>500 ] <- 500
fit0 <- coxph( Surv(survival.time,survival.status)
        ~ score + cell.type + tx + age, na.action=na.omit )
eta <- fit0$linear.predictor
tmax=365
AUC.CC=risksetAUC(Stime=survival.time,
       status=survival.status, marker=eta, method="Cox", tmax=tmax);
```

`risksetROC` *Incident/Dynamic (I/D) ROC curve, AUC and integrated AUC (iAUC) estimation of censored survival data*

Description

This function creates risksetROC from a survival data set

Usage

```
risksetROC(Stime, entry=NULL, status, marker, predict.time, method="Cox",
                        span=NULL, order=1, window="asymmetric", prop=0.5,
                        plot=TRUE, type="l", xlab="FP", ylab="TP",
                        ...)
```

Arguments

`Stime`	For right censored data, this is the follow up time. For left truncated data, this is the ending time for the interval.
`entry`	For left truncated data, this is the entry time of the interval. The default is set to NULL for right censored data.
`status`	survival status, 1 if had an event and 0 otherwise
`marker`	marker
`predict.time`	time point of interest
`method`	either of "Cox", "LocalCox" and "Schoenfeld", default is "Cox"
`span`	bandwidth parameter that controls the size of a local neighborhood, needed for *method="LocalCox"* or *method="Schoenfeld"*
`order`	0 or 1, locally mean if 0 and local linear if 1, needed for method="Schoenfeld", default is 1
`window`	either of "asymmetric" or "symmetric", default is asymmetric, needed for method="LocalCox"
`prop`	what proportion of the time-interval to consider when doing a local Cox fitting at *predict.time*, needed for *method="LocalCox"*, default is 0.5.
`plot`	TRUE or FALSE, default is TRUE
`type`	default is "l", can be either of "p" for points, "l" for line, "b" for both
`xlab`	label for x-axis
`ylab`	label for y-axis
`...`	additional plot arguments

Details

This function creates and plots ROC based on incident/dynamic definition of Heagerty, et. al. based on a survival data and marker values. If proportional hazard is assumed then method="Cox" can be used. In case of non-proportional hazard, either of "LocalCox" or "Schoenfeld" can be used. These two methods differ in how the smoothing is done. If *plot="TRUE"* then the ROC curve is plotted with the diagonal line. Additional plot arguments can be supplied.

Value

Returns a list of the following items:

eta	unique marker values for calculation of TP and FP
TP	True Positive values corresponding to unique marker values
FP	False Positive values corresponding to unique marker values
AUC	Area Under (ROC) Curve at time predict.time

Author(s)

Paramita Saha

References

Heagerty, P.J., Zheng Y. (2005) Survival Model Predictive Accuracy and ROC curves *Biometrics*, **61**, 92 – 105

See Also

llCoxReg(), SchoenSmooth(), CoxWeights()

Examples

```
library(MASS)
data(VA)
survival.time=VA$stime
survival.status=VA$status
score <- VA$Karn
cell.type <- factor(VA$cell)
tx <- as.integer( VA$treat==1 )
age <- VA$age
survival.status[survival.time>500 ] <- 0
survival.time[survival.time>500 ] <- 500
fit0 <- coxph( Surv(survival.time,survival.status)
        ~ score + cell.type + tx + age, na.action=na.omit )
eta <- fit0$linear.predictor

ROC.CC30=risksetROC(Stime=survival.time, status=survival.status,
                    marker=eta, predict.time=30, method="Cox",
                    main="ROC Curve", lty=2, col="red")
```

SchoenSmooth *Incident/Dynamic (I/D) ROC curve, AUC and integrated AUC (iAUC) estimation of censored survival data*

Description

This function smooths the Schoenfeld residuals using Epanechnikov's optimal kernel.

Usage

```
SchoenSmooth(fit, Stime, status, span=0.40, order=0, entry=NULL)
```

Arguments

fit	the result of fitting a Cox regression model, using the coxph function
Stime	Survival times in case of right censored data and exit time for left truncated data
status	Survival status
span	bandwidth parameter that controls the size of a local neighborhood
order	0 or 1, locally mean if 0 and local linear if 1
entry	entry time when left censored data is considered, default is NULL for only right censored data

Details

This function smooths the Schoenfeld residuals to get an estimate of time-varying effect of the marker using Epanechnikov's optimal kernel using either local mean or local linear smoother.

Value

Returns a list of following items:

time	failure times
beta	estimate of time-varying parameter $\beta(t)$ at each unique failure time.

Author(s)

Patrick J. Heagerty

References

Heagerty, P.J., Zheng Y. (2005) Survival Model Predictive Accuracy and ROC curves *Biometrics*, **61**, 92 – 105

Examples

```
data(pbc)
## considering only randomized patients
pbc1 <- pbc[1:312,]
## create new censoring variable combine 0,1 as 0, 2 as 1
survival.status <- ifelse( pbc1$status==2, 1, 0)
survival.time <- pbc1$fudays
pbc1$status1 <- survival.status
fit <- coxph( Surv(fudays,status1) ~ log(bili) +
                                     log(protime) +
                                     edema +
                                     albumin +
                                     age,
              data=pbc1 )
eta5 <- fit$linear.predictors
x <- eta5
nobs <- length(survival.time[survival.status==1])
span <- 1.5*(nobs^(-0.2))
fitCox5 <- coxph( Surv(survival.time,survival.status) ~ x )
bfnx1.1 <- SchoenSmooth( fit=fitCox5, Stime=survival.time, status=survival.status,
                         span=span, order=1)
bfnx1.0 <- SchoenSmooth( fit=fitCox5, Stime=survival.time, status=survival.status,
                         span=span, order=0)
plot(bfnx1.1$time, bfnx1.1$beta, type="l", xlab="Time", ylab="beta(t)")
lines(bfnx1.0$time, bfnx1.0$beta, lty=3)
```

---

weightedKM *Incident/Dynamic (I/D) ROC curve, AUC and integrated AUC (iAUC) estimation of censored survival data*

Description

This function estimates S(t) where sampling weights are permitted.

Usage

```
weightedKM(Stime, status, wt=NULL, entry=NULL)
```

Arguments

Stime	Survival times when right censored data is considered. In case of interval censored data this is the end point for the time interval.
status	Survival status
wt	weight, default is unweighted
entry	entry times in case of interval censored data, default is *NULL* when right censored data is considered

Details

This function obtains survival function estimate where sampling weights are permitted.

Value

Returns a list of following items:

`time`	ordered unique failure times
`survival`	survival estimate at the unique failure times

Author(s)

Patrick J. Heagerty

References

Heagerty, P.J., Zheng Y. (2005) Survival Model Predictive Accuracy and ROC curves *Biometrics*, **61**, 92 – 105

Examples

```
data(pbc)
## considering only randomized patients
pbc1 <- pbc[1:312,]
## create new censoring variable combine 0,1 as 0, 2 as 1
survival.status <- ifelse( pbc1$status==2, 1, 0)
survival.time <- pbc1$fudays
kout <- weightedKM(Stime=survival.time, status=survival.status)
KM.plot(kout$time,kout$survival)
```

Index