

# Package ‘tdROC’

March 30, 2016

**Type** Package

**Version** 1.0

**Date** 2016-03-26

**Title** Nonparametric Estimation of Time-Dependent ROC Curve from Right Censored Survival Data

**Depends** R(>= 3.1.1), survival

**Author** Liang Li <LLi15@mdanderson.org>, Cai Wu <cwu13@mdanderson.org>  
Department of Biostatistics, The University of Texas MD Anderson Cancer Center

**Maintainer** Cai Wu <cwu13@mdanderson.org>

**Description** Compute time-dependent ROC curve from censored survival data using nonparametric weight adjustments.

**License** GPL (>= 2)

**LazyData** Yes

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-03-30 00:59:03

## R topics documented:

calc.AUC . . . . .	2
is.monotone . . . . .	2
mayo . . . . .	3
plot.tdROC . . . . .	4
tdROC . . . . .	5
<b>Index</b>	<b>7</b>

---

`calc.AUC`*Calculate the area under a ROC curve (AUC)*

---

**Description**

This function reads in a vector of sensitivity and a vector of specificity and calculates the area under the curve (AUC) by trapezoidal integration.

**Usage**

```
calc.AUC(sens, spec)
```

**Arguments**

`sens` a numerical vector of sensitivity values within the range of (0, 1).  
`spec` a numerical vector of specificity values within the range of (0, 1).

**Value**

AUC as a numerical scalar.

**Note**

This function sorts `sens` and `1-spec` in an increasing order. A 0 and 1 will be added to the two ends of the sorted vectors. The Area Under the Curve (AUC) is obtained by trapezoidal integration of the area under the piecewise linear curve obtained by connecting points in `sens` and `1-spec`.

**Examples**

```
sens <- c(0.99, 0.97, 0.83, 0.60, 0.40, 0.20) ;  
spec <- c(0.50, 0.61, 0.80, 0.90, 0.95, 0.98) ;  
plot( 1-spec, sens, type = "l" ) ;  
points(1-spec, sens) ;  
calc.AUC( sens, spec ) ;
```

---

`is.monotone`*Check monotonicity*

---

**Description**

This internal function determines if a numeric vector `X` is monotone (increasing or decreasing) or not. Ties in the sequence are considered as monotone.

**Usage**

```
is.monotone(X)
```

**Arguments**

X a numerical vector to be tested.

**Value**

Returns True or False.

**Examples**

```
a <- 1:10 ;
b <- c( 1:5, 4:2 ) ;
c <- c( 1:4, 4, 4, 4, 5:10 ) ;
d <- c( 10:6, 3:-2 ) ;

is.monotone( a )
is.monotone( b )
is.monotone( c )
is.monotone( d )
```

---

mayo

*example data: Mayo Marker Data*

---

**Description**

Two marker values with event time and censoring status for the subjects in Mayo PBC data

**Usage**

```
data(mayo)
```

**Format**

A data frame with 312 observations and 4 variables: time (event time/censoring time), censor (censoring indicator), mayoscore4, mayoscore5. The two scores are derived from 4 and 5 covariates respectively.

**References**

Heagerty, P. J., & Zheng, Y. (2005). Survival model predictive accuracy and ROC curves. *Biometrics*, 61(1), 92-105.

**See Also**

[mayo](#)

---

plot.tdROC

*Plot the time-dependent ROC curve*


---

### Description

Plot the ROC curve estimated by `tdROC()`.

### Usage

```
## S3 method for class 'tdROC'
plot(x, lwd = 2, xlab = "1-specificity",
     ylab = "sensitivity", xlim = c(0, 1), ylim = c(0, 1), col = "black",
     main = "ROC curve", abline = T, ...)
```

### Arguments

<code>x</code>	the object returned by <code>tdROC()</code> .
<code>lwd</code>	user-specified line width. Default is 2.
<code>xlab</code>	user-specified label for x-axis. Default is "1-specificity".
<code>ylab</code>	user-specified label for y-axis. Default is "sensitivity".
<code>xlim</code>	user-specified limit for x axis. Default is <code>c(0,1)</code> .
<code>ylim</code>	user-specified limit for y axis. Default is <code>c(0,1)</code> .
<code>col</code>	user-specified color for ROC curve. Default is "black".
<code>main</code>	user-specified title for the plot. Default is "ROC curve"
<code>abline</code>	user-specified reference diagonal line. Default is True.
<code>...</code>	for future methods

### Value

Returns a plot of ROC curve.

### Examples

```
library( survival )
data( mayo );
dat <- mayo[ , c( "time", "censor", "mayoscore5" ) ];
fm <- tdROC( X = dat$mayoscore5, Y = dat$time, delta = dat$censor,
            tau = 365*6, span = 0.1, nboot = 0, alpha = 0.05, n.grid = 1000, cut.off = 5:9 )
# plot the object "fm" from tdROC()
plot.tdROC( fm );
```

---

tdROC	<i>Calculate Time-dependent ROC and AUC</i>
-------	---

---

### Description

This is the main function of this package. It calculates the time-dependent sensitivity and specificity and area under the curve (AUC) using a nonparametric weighting adjustment. It also provides variance estimation through bootstrap.

### Usage

```
tdROC(X, Y, delta, tau, span = 0.1, h = NULL, type = "uniform",
      cut.off = NULL, nboot = 0, alpha = 0.05, n.grid = 1000,
      X.min = NULL, X.max = NULL)
```

### Arguments

X	a numeric vector of biomarker values. Same length with Y and delta.
Y	a numeric vector of time to event. Same length with X and delta.
delta	a vector of binary indicator of event (1) or censoring (0). Same length with X and Y.
tau	a scalar, the prediction horizon at which the prediction is evaluated.
span	a numeric value, the proportion of neighbour observations used in nearest neighbor method, default is 0.1.
h	a numeric value, the bandwidth of kernel weights, default is NULL. If not specified, the function will use the value of span to calculate kernel weights. In case both span and h are specified, the function will use h.
type	a character value, indicating the type of kernel function used to calculate kernel weights. Default is "uniform" kernel. Other options are "Epanechnikov" and "normal". It will only be used when the bandwidth h is specified.
cut.off	a vector of biomarker cut-off values at which sensitivity and specificity will be calculated. When bootstrap is requested, the corresponding confidence intervals will also be provided.
nboot	the number of bootstrap replications to be used for variance estimation; default is nboot = 0, corresponding to no variance estimation.
alpha	1-level of confidence interval, default is 0.05. It is used only when nboot > 0.
n.grid	an positive integer, the number of grid points used when calculating the ROC curve. The default is 1000.
X.min	the lower boundary of grid cut-off points for biomarker X. If not specified, default will be the minimum of biomarker X.
X.max	the upper boundary of grid cut-off points for biomarker X. If not specified, default will be the maximum of biomarker X. X.min and X.max are not needed for point estimate but are needed for bootstrap variance estimation.

## Details

This function reads in the prognostic biomarker value  $X$ , the time-to-event data  $Y$  and censoring indicator  $\delta$  to calculate the sensitivity and specificity at the prediction horizon  $\tau$  for a series of specified grid points. It uses simple nonparametric weight adjustments for right censored data (Li *et al.*, 2015).

## Value

Returns a list of the following items:

ROC: a data frame of dimension  $(2+n.grid) \times 3$ , the three columns are: grid, sens, and spec.

AUC: a data frame of one row and four columns: AUC, standard error of AUC, the lower and upper limits of bootstrap CI. AUC is calculated by integrating the area under ROC curve with trapezoidal method.

AUC2: a data frame of one row and four columns: AUC2, standard error of AUC2, the lower and upper limits of bootstrap CI. AUC2 is the AUC calculated by the concordance based formula (Li *et al.*, 2015).

prob: a data frame of three columns if `nboot=0`: cut.off, sens, and spec. If `nboot>0`, another six columns of standard error, lower and upper limits of both sens and spec will be added. The number of rows equals length of cut.off. A series of sensitivity and specificity are calculated at requested cut.off points.

## Author(s)

Liang Li, Cai Wu

## References

Li, Liang, Bo Hu, and Tom Greene. "A Simple Method to Estimate the Time-dependent ROC Curve Under Right Censoring." (2015). <http://biostats.bepress.com/cobra/art114/>

## See Also

[survfit](#), [mayo](#)

## Examples

```
library( survival ) ;
data( mayo ) ;
dat <- mayo[ ,c( "time","censor","mayoscore5" )] ;

fm <- tdROC( X = dat$mayoscore5, Y = dat$time, delta = dat$censor,
            tau = 365*6, span = 0.1, nboot = 0, alpha = 0.05,
            n.grid = 1000, cut.off = 5:9 ) ;
```

# Index

\*Topic **datasets**

    mayo, [3](#)

calc.AUC, [2](#)

is.monotone, [2](#)

mayo, [3](#), [3](#), [6](#)

plot.tdROC, [4](#)

survfit, [6](#)

tdROC, [5](#)