

Package ‘ThresholdROCsurvival’

November 8, 2023

Type Package

Title Diagnostic Ability Assessment with Right-Censored Data at a Fixed Time t

Version 1.2.0

Date 2023-11-08

Description We focus on the diagnostic ability assessment of medical tests when the outcome of interest is the status (alive or dead) of the subjects at a certain time-point t. This binary status is determined by right-censored times to event and it is unknown for those subjects censored before t. Here we provide three methods (unknown status exclusion, imputation of censored times and using time-dependent ROC curves) to evaluate the diagnostic ability of binary and continuous tests in this context. Two references for the methods used here are Skaltsa et al. (2010) <[doi:10.1002/bimj.200900294](https://doi.org/10.1002/bimj.200900294)> and Heagerty et al. (2000) <[doi:10.1111/j.0006-341x.2000.00337.x](https://doi.org/10.1111/j.0006-341x.2000.00337.x)>.

License GPL (>= 2)

Depends R (>= 4.0.0)

Imports boot, InformativeCensoring, pROC, psych, survival, ThresholdROC

LazyData TRUE

NeedsCompilation no

Author Sara Perez-Jaume [aut, cre],
Josep L Carrasco [aut]

Maintainer Sara Perez-Jaume <spjaume@gmail.com>

Repository CRAN

Date/Publication 2023-11-08 18:20:06 UTC

R topics documented:

ThresholdROCsurvival-package	2
diagnostic_assessment_binary	3
diagnostic_assessment_continuous	5
NSCLC	9

Index	10
--------------	-----------

ThresholdROCsurvival-package

Diagnostic Ability Assessment with Right-Censored Data at a Fixed Time t

Description

We focus on the diagnostic ability assessment of medical tests when the outcome of interest is the status (alive or dead) of the subjects at a certain time-point t . This binary status is determined by right-censored times to event and it is unknown for those subjects censored before t . Here we provide three methods (unknown status exclusion, imputation of censored times and using time-dependent ROC curves) to evaluate the diagnostic ability of binary and continuous tests in this context. Two references for the methods used here are Skaltsa et al. (2010) <doi:10.1002/bimj.200900294> and Heagerty et al. (2000) <doi:10.1111/j.0006-341x.2000.00337.x>.

Details

The functions in this package are `diagnostic_assessment_binary()` (for binary medical tests) and `diagnostic_assessment_continuous()` (for continuous medical tests).

Author(s)

Sara Perez-Jaume and Josep L Carrasco

Maintainer: Sara Perez-Jaume

References

Heagerty PJ, Lumley T, Pepe MS. Time-Dependent ROC Curves for Censored Survival Data and a Diagnostic Marker. *Biometrics* 2000; 56(2): 337-344. doi: 10.1111/j.0006-341X.2000.00337.x

Hsu CH, Taylor JMG, Murray S, Commenges D. Survival analysis using auxiliary variables via non-parametric multiple imputation. *Statistics in Medicine* 2006; 25(20): 3503-3517. doi: <https://doi.org/10.1002/sim.2452>

Perez-Jaume S, Skaltsa K, Pallares N, Carrasco JL. ThresholdROC: Optimum Threshold Estimation Tools for Continuous Diagnostic Tests in R. *Journal of Statistical Software* 2017; 82(4): 1-21. doi: 10.18637/jss.v082.i04

Skaltsa K, Jover L, Carrasco JL. Estimation of the diagnostic threshold accounting for decision costs and sampling uncertainty. *Biometrical Journal* 2010; 52(5): 676-697. doi: 10.1002/bimj.200900294

diagnostic_assessment_binary

Diagnostic ability assessment for binary diagnostic tests

Description

This function estimates sensitivity and specificity at a fixed time-point t for binary diagnostic tests with survival data by using two methods: 1) unknown status exclusion (USE), which excludes subjects with unknown status at t ; and 2) imputation of censored times (ICT), a method based on multiple imputation. The status of the subjects at a certain time-point of interest t (the event occurred before or at t or not) is defined by the time-to-event variable.

Usage

```
diagnostic_assessment_binary(binary.var, time, status, predict.time,
                             method=c("USE", "ICT"), index=c("all", "sens", "spec"),
                             m=10, ci=TRUE, alpha=0.05, range=3)
```

Arguments

<code>binary.var</code>	binary variable to be used as predictor of the status. It should be a factor which two levels: - (negative, which indicates absence of the event) and + (positive, which indicates presence of the event)
<code>time</code>	survival time
<code>status</code>	censoring status codified as 0=censored, 1=event
<code>predict.time</code>	time-point of interest to define the subjects' status as event present or absent
<code>method</code>	method to be used in the estimation process. The user can choose between USE (unknown status exclusion) or ICT (imputation of censored times). Default, USE
<code>index</code>	indices to be estimated. The user can choose one or more of the following: sens and spec. The option all (default) estimates all two indices
<code>m</code>	the number of data sets to impute. Default, 10
<code>ci</code>	Should a confidence interval be calculated? Default, TRUE
<code>alpha</code>	significance level for the confidence interval. Default, 0.05
<code>range</code>	this value, which is passed to <code>boxplot</code> function from <code>graphics</code> package, determines the data points that are considered to be extreme in the estimates and standard errors from the multiple imputation process. We consider extreme observations those that exceed <code>range</code> times the interquartile range. If extreme observations are found in the estimates or standard errors from the multiple imputation process, Winsorized estimators (Wilcox, 2012) are used for the point estimate and the between and within variances. Default, 3

Details

When method is USE: First, the algorithm determines the status of the subjects at time `predict.time`. Those censored subjects whose status could not be determined are excluded from the analysis. Then, diagnostic ability is assessed with standard methods in the binary setting.

When method is ICT: First, the algorithm determines the status of the subjects at time `predict.time`. For those subjects whose status could not be determined because their censored time is lower than `t`, we impute survival times using the method of Hsu et al (2006), that is implemented in the package `InformativeCensoring` (Ruau et al, 2020). The status of the subjects is then determined by these imputed times and is used to estimate the indices in `index`. Confidence intervals are calculated using the standard error proposed by Rubin (1987).

Value

An object of class `diagnostic_assessment`, which is a list with the following components:

<code>sens</code>	Sensitivity estimate and its corresponding confidence interval (if <code>ci=TRUE</code>), only if sensitivity has been included in <code>index</code>
<code>spec</code>	Specificity estimate and its corresponding confidence interval (if <code>ci=TRUE</code>), only if specificity has been included in <code>index</code>
<code>method</code>	method used in the estimation
<code>alpha</code>	significance level provided by the user
<code>data</code>	A <code>data.frame</code> containing the following columns previously provided by the user: <code>cont.var</code> , <code>time</code> and <code>status</code> , and a new column <code>statusNA</code> , which contains the status of the subjects at time <code>predict.time</code> (0=no event, 1=event, NA=unknown)

References

- Heagerty PJ, Lumley T, Pepe MS. Time-Dependent ROC Curves for Censored Survival Data and a Diagnostic Marker. *Biometrics* 2000; 56(2): 337-344. doi: 10.1111/j.0006-341X.2000.00337.x
- Heagerty PJ, Saha-Chaudhuri P (2013). `survivalROC`: Time-dependent ROC curve estimation from censored survival data. R package version 1.0.3. <https://CRAN.R-project.org/package=survivalROC>
- Hsu CH, Taylor JMG, Murray S, Commenges D. Survival analysis using auxiliary variables via non-parametric multiple imputation. *Statistics in Medicine* 2006; 25(20): 3503-3517. doi: <https://doi.org/10.1002/sim.2452>
- Ruau D, Burkoff N, Bartlett J, Jackson D, Jones E, Law M and Metcalfe P (2020). `InformativeCensoring`: Multiple Imputation for Informative Censoring. R package version 0.3.5. <https://CRAN.R-project.org/package=InformativeCensoring>
- Rubin DB. *Multiple Imputation for Nonresponse in Surveys*. Wiley Series in Probability and Statistics. John Wiley & Sons (1987)
- Wilcox, R. *Introduction to Robust Estimation and Hypothesis Testing*. 3rd Edition. Elsevier, Amsterdam (2012)
- Zhou XH, Obuchowski NA and McClish DK. *Statistical methods in diagnostic medicine*. John Wiley and sons (2002)

See Also[diagnostic_assessment_continuous](#)**Examples**

```

data(NSCLC)
NSCLC$COL_cat <- factor(ifelse(NSCLC$COL>=10, "+", "-"))
set.seed(2020)
with(NSCLC, diagnostic_assessment_binary(COL_cat, OS, ST,
    1095, method="ICT", m=10, ci=TRUE))

```

diagnostic_assessment_continuous

Diagnostic ability assessment for continuous diagnostic tests

Description

This function estimates the AUC, optimal threshold, sensitivity and specificity at a fixed time-point t for continuous diagnostic tests with survival data by using three methods: 1) unknown status exclusion (USE), which excludes subjects with missing status at t ; 2) imputation of censored times (ICT), a method based on multiple imputation; and 3) survivalROC, which uses a method based on time-dependent ROC curves. The status of the subjects at a certain time-point of interest t (the event occurred before or at t or not) is defined by the time-to-event variable.

Usage

```

diagnostic_assessment_continuous(cont.var, time, status, predict.time,
    method=c("USE", "ICT", "survivalROC"),
    index=c("all", "AUC", "threshold", "sens", "spec"),
    costs=NULL, R=NULL,
    method.thres=c("normal", "empirical"),
    var.equal=FALSE, lambda=0.05, m=10,
    ci=TRUE, plot=FALSE, alpha=0.05,
    B=1000, range=3, ...)

```

Arguments

<code>cont.var</code>	continuous variable or biomarker to be used as predictor of the status
<code>time</code>	survival time
<code>status</code>	censoring status codified as 0=censored, 1=event
<code>predict.time</code>	time-point of interest to define the subjects' status as event present or absent
<code>method</code>	method to be used in the estimation process. The user can choose between USE (unknown status exclusion), ICT (imputation of censored times) or survivalROC (time-dependent ROC curves). Default, ME

index	indices to be estimated. The user can choose one or more of the following: AUC, threshold, sens (sensitivity achieved by the optimal threshold), spec (specificity achieved by the optimal threshold). The option all (default) estimates all four indices
costs	cost matrix. Costs should be entered as a 2x2 matrix, where the first row corresponds to the true positive and true negative costs and the second row to the false positive and false negative costs. Default cost values (costs=NULL, when also R=NULL) are a combination of costs that yields R=1, which is equivalent to the Youden index method (for details about this concept, see Details and References)
R	if the cost matrix costs is not set (that is, costs=NULL), R desired (the algorithm will choose a suitable combination of costs that leads to R). Default, NULL (which leads to R=1 using the default costs). For details about this concept, see Details and References
method.thres	method used in the estimation: "normal" (default) or "empirical". The user can specify just the initial letters. See Details for more information about the methods available
var.equal	when method="normal", assume equal variances? Default, FALSE. When method="empirical", var.equal is ignored
lambda	smoothing parameter for the NNE algorithm used in survivalROC() function. Default, 0.05
m	the number of data sets to impute. Default, 10
ci	Should a confidence interval be calculated? Default, TRUE
plot	Should some graphs about the estimation be plotted? Default, FALSE
alpha	significance level for the confidence interval. Default, 0.05
B	number of bootstrap resamples for the confidence interval. Only used when method is survivalROC. Otherwise, this argument is ignored. Default, 1000
range	this value, which is passed to boxplot function from graphics package, determines the data points that are considered to be extreme in the estimates and standard errors from the multiple imputation process. We consider extreme observations those that exceed range times the interquartile range. If extreme observations are found in the estimates or standard errors from the multiple imputation process, Winsorized estimators (Wilcox, 2012) are used for the point estimate and the between and within variances. Default, 3
...	extra arguments to be passed to plot()

Details

When method is USE: First, the algorithm determines the status of the subjects at time `predict.time`. Those censored subjects whose status could not be determined are excluded from the analysis. Then, diagnostic ability is assessed with standard methods in the binary setting.

When method is ICT: First, the algorithm determines the status of the subjects at time `predict.time`. For those subjects whose status could not be determined because their censored time is lower than `t`, we impute survival times using the method of Hsu et al (2006), that is implemented in the package `InformativeCensoring` (Ruau et al, 2020). The status of the subjects is then determined by these

imputed times and is used to estimate the indices in `index`. Confidence intervals are calculated using the standard error proposed by Rubin (1987).

When `method` is `survivalROC`: Diagnostic ability is assessed by constructing the ROC curve at time `t` through time-dependent ROC curves (Heagerty et al, 2000). Confidence intervals are obtained using normal and percentile bootstrap. In normal bootstrap, the bootstrap is used to obtain an estimate of the standard error of the threshold estimate, and then the standard normal distribution is used for the confidence interval calculation. In percentile bootstrap, `B` bootstrap resamples are generated and the threshold is estimated in all of them. Then, the confidence interval is calculated as the empirical $1-\alpha/2$ and $1+\alpha/2$ percentiles of the `B` bootstrap estimates.

For parameter `method.thres`, the method used in the estimation of the optimal threshold, the user can choose between "normal" (assumes binormality) or "empirical" (leaves out any distributional assumption). When `method="normal"`, the user can specify if the algorithm should assume equal or different variances using the parameter `var.equal`. For further details see the `thres2` function in the `ThresholdROC` package.

`R`, mentioned in parameters `costs` and `R`, is the product of the non-disease odds and the cost ratio:

$$R = ((1 - p)/p)((C_{TN} - C_{FP})/(C_{TP} - C_{FN})),$$

where p is the disease prevalence (estimated using Kaplan-Meier) and C_i are the classification costs.

Value

An object of class `diagnostic_assessment`, which is a list with the following components:

<code>AUC</code>	AUC estimate and its corresponding confidence interval (if <code>ci=TRUE</code>), only if AUC has been included in <code>index</code>
<code>threshold</code>	threshold estimate and its corresponding confidence interval (if <code>ci=TRUE</code>), only if threshold has been included in <code>index</code>
<code>sens</code>	Sensitivity estimate (achieved by the optimal threshold) and its corresponding confidence interval (if <code>ci=TRUE</code>), only if sensitivity has been included in <code>index</code>
<code>spec</code>	Specificity estimate (achieved by the optimal threshold) and its corresponding confidence interval (if <code>ci=TRUE</code>), only if specificity has been included in <code>index</code>
<code>method</code>	method used in the estimation
<code>alpha</code>	significance level provided by the user
<code>data</code>	A <code>data.frame</code> containing the following columns previously provided by the user: <code>cont.var</code> , <code>time</code> and <code>status</code> , and a new column <code>statusNA</code> , which contains the status of the subjects at time <code>predict.time</code> (0=no event, 1=event, NA=unknown)

References

Heagerty PJ, Lumley T, Pepe MS. Time-Dependent ROC Curves for Censored Survival Data and a Diagnostic Marker. *Biometrics* 2000; 56(2): 337-344. doi: 10.1111/j.0006-341X.2000.00337.x

Heagerty PJ, Saha-Chaudhuri P (2022). `survivalROC`: Time-dependent ROC curve estimation from censored survival data. R package version 1.0.3.1. <https://CRAN.R-project.org/package=survivalROC>

- Hsu CH, Taylor JMG, Murray S, Commenges D. Survival analysis using auxiliary variables via non-parametric multiple imputation. *Statistics in Medicine* 2006; 25(20): 3503-3517. doi: <https://doi.org/10.1002/sim.2452>
- Kottas M, Kuss O, Zapf A. A modified Wald interval for the area under the ROC curve (AUC) in diagnostic case-control studies. *BMC Medical Research Methodology* 2014; 14(26). doi:10.1186/1471-2288-14-26
- Perez-Jaume S, Skaltsa K, Pallares N, Carrasco JL. ThresholdROC: Optimum Threshold Estimation Tools for Continuous Diagnostic Tests in R. *Journal of Statistical Software* 2017; 82(4): 1-21. doi: 10.18637/jss.v082.i04
- Ruau D, Burkoff N, Bartlett J, Jackson D, Jones E, Law M and Metcalfe P (2020). InformativeCensoring: Multiple Imputation for Informative Censoring. R package version 0.3.5. <https://CRAN.R-project.org/package=InformativeCensoring>
- Robin X, Turck N, Hainard A, Tiberti N, Lisacek F, Sanchez JC, Muller M. pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics* 2011; 12. doi:10.1186/1471-2105-12-77
- Rubin DB. *Multiple Imputation for Nonresponse in Surveys*. Wiley Series in Probability and Statistics. John Wiley & Sons (1987)
- Skaltsa K, Jover L, Carrasco JL. Estimation of the diagnostic threshold accounting for decision costs and sampling uncertainty. *Biometrical Journal* 2010; 52(5): 676-697. doi: 10.1002/bimj.200900294
- Wilcox, R. *Introduction to Robust Estimation and Hypothesis Testing*. 3rd Edition. Elsevier, Amsterdam (2012)

See Also

[diagnostic_assessment_binary](#)

Examples

```
library(ThresholdROCsurvival)
data(NSCLC)
# unknown status exclusion (Youden index maximization, R=1)
with(NSCLC, diagnostic_assessment_continuous(log(COL), OS,
                                           ST, 1095, method="USE", method.thres="normal",
                                           var.equal=FALSE, ci=TRUE))

# multiple imputation (Youden index maximization, R=1)
set.seed(2020)
with(NSCLC, diagnostic_assessment_continuous(log(COL), OS,
                                           ST, 1095, method="ICT", method.thres="normal",
                                           var.equal=FALSE, m=50, ci=TRUE))

# missing exclusion (R=1.1)
with(NSCLC, diagnostic_assessment_continuous(log(COL), OS,
                                           ST, 1095, method="USE", method.thres="normal",
                                           var.equal=FALSE, ci=TRUE, R=1.1))

# unknown status exclusion (R=1.1)
set.seed(2020)
```



```
with(NSCLC, diagnostic_assessment_continuous(log(COL), OS,  
ST, 1095, method="ICT", method.thres="normal",  
var.equal=FALSE, m=50, ci=TRUE, R=1.1))
```

NSCLC

Non-small cell lung cancer (NSCLC) data

Description

Non-small cell lung cancer (NSCLC) is the most common lung cancer and comprises several subtypes of lung cancers. These data come from a study by Alcaraz *et al.*, 2019, in which the authors investigated the prognostic value of some activation markers in NSCLC.

Usage

```
data("NSCLC")
```

Format

A data frame with 203 observations on the following 4 variables.

ID subject's identifier

OS overall survival, that is, the time from surgery until death or last follow-up, in days

ST censoring status (0=censored, 1=dead)

COL percentage of collagen quantified using an imaging technique from tumour samples

Source

Alcaraz J, Carrasco JL, Millares L, et al. Stromal markers of activated tumor associated fibroblasts predict poor survival and are associated with necrosis in non-small cell lung cancer. *Lung Cancer* 2019; 135: 151 - 160. doi: 10.1016/j.lungcan.2019.07.020

Examples

```
data(NSCLC)  
summary(NSCLC)
```

Index

* **datasets**

NSCLC, [9](#)

* **package**

ThresholdROCsurvival-package, [2](#)

diagnostic_assessment_binary, [3](#), [8](#)

diagnostic_assessment_continuous, [5](#), [5](#)

NSCLC, [9](#)

ThresholdROCsurvival-package, [2](#)