

Package ‘alternativeROC’

June 29, 2025

Type Package

Title Alternative and Fast ROC Analysis

Version 1.0.0

Date 2025-06-27

Description Alternative and fast algorithms for the analysis of receiver operating characteristics curves (ROC curves) as described in Thomas et al. (2017) [<doi:10.1186/s41512-017-0017-y>](https://doi.org/10.1186/s41512-017-0017-y) and Thomas et al. (2023) [<doi:10.1016/j.ajogmf.2023.101110>](https://doi.org/10.1016/j.ajogmf.2023.101110).

Depends R (>= 4.0.0), stats,utils, graphics, pROC, plyr, sn, Hmisc

Suggests roxygen2 (>= 3.1.0), testthat

Imports Rcpp (>= 1.0.9)

LinkingTo Rcpp

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

URL <https://bitbucket.org/SQ4/alternativeROC>

ByteCompile true

Repository CRAN

BugReports <https://bitbucket.org/SQ4/alternativeROC/issues>

NeedsCompilation yes

Author Gregoire Thomas [aut, cre] (ORCID: [<https://orcid.org/0000-0002-6247-9438>](https://orcid.org/0000-0002-6247-9438)),
Robin Tuytten [ctb] (ORCID: [<https://orcid.org/0000-0002-8734-7335>](https://orcid.org/0000-0002-8734-7335)),
Jef Moerman [ctb],
Xavier Robin [cph] (Co-author of delong.cpp from package pROC),
Stefan Siegert [cph] (Co-author of delong.cpp from package pROC)

Maintainer Gregoire Thomas [<gregoire.thomas@SQU4RE.com>](mailto:gregoire.thomas@SQU4RE.com)

Date/Publication 2025-06-29 06:50:02 UTC

Contents

alternativeROC	2
alternativeROC-common-args	2
cinpv	3
cippv	4
npv	5
npv.seatssp	5
npv.spatse	6
p.auc	6
plotROC	7
ppv	9
ppv.seatssp	9
ppv.spatse	10
pvs	10
rocrperf	11
rocthreshold	14
Index	15

alternativeROC	<i>alternativeROC</i>
----------------	-----------------------

Description

Alternative analysis methods for the analysis of receiver operating characteristics curve (ROC curve) as described in Thomas et al. (2017) <doi:10.1186/s41512-017-0017-y>.

Author(s)

Gregoire Thomas <gregoire.thomas@SQU4RE.com>

alternativeROC-common-args	<i>Arguments used across the functions of the alternativeROC package.</i>
----------------------------	---

Description

Arguments used across the functions of the alternativeROC package.

Arguments

roc	Object of class pROC.
se	Sensitivity.
sp	Specificity.
ppv	Positive predictive value.
npv	Negative predictive value.
prevalence	Prevalence of the endpoint in the study population.
boot.n	Number of bootstrap replicates. Default: 2000.
quantiles	Quantiles. Default: c(0.5,.025,.975).
conf.level	Width of the confidence interval. Default: 0.95 (i.e., 95% CI).

Value

No return value, used for the documentation of the functions of the package.

cinpv	<i>Diagnostic performance: Confidence interval for negative predictive value</i>
-------	--

Description

This function computes a confidence interval for the negative predictive value from a ROC curve, given the prevalence of the negative outcome.

Usage

```
cinpv(roc, npv, prevalence, boot.n, quantiles = c(0.5, 0.025, 0.975), ...)
```

Arguments

roc	Object of class pROC.
npv	Negative predictive value.
prevalence	Prevalence of the endpoint in the study population.
boot.n	Number of bootstrap replicates. Default: 2000.
quantiles	Quantiles. Default: c(0.5,.025,.975).
...	Not used.

Details

This function computes a confidence interval for the negative predictive value from a ROC curve, given the prevalence of the negative outcome. The confidence interval is computed using bootstrap resampling.

Value

A numeric vector of length 3 containing the median, lower bound, and upper bound of the confidence interval.

cippv	<i>Diagnostic performance: Confidence interval for positive predictive value</i>
-------	--

Description

This function computes a confidence interval for the positive predictive value from a ROC curve, given the prevalence of the positive outcome.

Usage

```
cippv(
  roc,
  ppv,
  prevalence,
  boot.n = 2000,
  quantiles = c(0.5, 0.025, 0.975),
  ...
)
```

Arguments

roc	Object of class pROC.
ppv	Positive predictive value.
prevalence	Prevalence of the endpoint in the study population.
boot.n	Number of bootstrap replicates. Default: 2000.
quantiles	Quantiles. Default: c(0.5,.025,.975).
...	Not used.

Details

This function computes a confidence interval for the positive predictive value from a ROC curve, given the prevalence of the positive outcome. The confidence interval is computed using bootstrap resampling.

Value

A numeric vector of length 3 containing the median, lower bound, and upper bound of the confidence interval.

npv	<i>Diagnostic performance: Negative predictive values from a ROC curve</i>
-----	--

Description

Diagnostic performance: Negative predictive values (NPV) from a ROC curve

Usage

```
npv(roc, prevalence)
```

Arguments

roc	Object of class pROC.
prevalence	Prevalence of the endpoint in the study population.

Value

A matrix with the following columns:

- **threshold**: The thresholds used to compute the sensitivity and specificity.
- **sensitivity**: The sensitivities at the threshold.
- **specificity**: The specificities at the threshold.
- **npv**: The negative predictive values at the threshold.
- **prevalence**: The prevalence of the endpoint in the study population, as provided in the input.

npv.seatasp	<i>Diagnostic performance: Sensitivity from specificity, NPV and prevalence</i>
-------------	---

Description

Diagnostic performance: Sensitivity from specificity, negative predictive value (NPV) and prevalence of the endpoint

Usage

```
npv.seatasp(sp, npv, prevalence)
```

Arguments

sp	Specificity.
npv	Negative predictive value.
prevalence	Prevalence of the endpoint in the study population.

Value

Sensitivity from specificity, NPV and prevalence.

npv.spatse	<i>Diagnostic performance: Specificity from sensitivity, NPV and prevalence</i>
------------	---

Description

Diagnostic performance: Specificity from sensitivity, negative predictive value (NPV) and prevalence of the endpoint

Usage

```
npv.spatse(se, npv, prevalence)
```

Arguments

se	Sensitivity.
npv	Negative predictive value.
prevalence	Prevalence of the endpoint in the study population.

Value

Specificity from sensitivity, NPV and prevalence.

p.auc	$p[AUC=0.5]$
-------	--------------

Description

$p[AUC=0.5]$ using DeLong's methods (DeLong et al. 1988).

Usage

```
p.auc(roc, ref = 0.5)
```

Arguments

roc	Object of class pROC.
ref	Reference AUC. Default 0.5.

Value

p value.

plotROC

*Diagnostic performance: Plot ROC curve***Description**

Diagnostic performance: Plot ROC curve

Usage

```
plotROC(
  x,
  annotate = FALSE,
  col.diagonal = "#00000080",
  lty.diagonal = "solid",
  lwd.diagonal = 1,
  col = "#303030",
  lwd = 2,
  cex = 2,
  ppv = NULL,
  npv = NULL,
  prevalence = NULL,
  col.pvs = "#888888",
  col.ci = "#dd0000",
  lwd.ci = lwd,
  len.ci = 0.1,
  boot.n = 1000,
  conf.level = 0.95,
  ...
)
```

Arguments

x	Object of class roc.
annotate	Annotate plot.
col.diagonal	Color of the diagonal.
lty.diagonal	Line type of the diagonal.
lwd.diagonal	Line width of the diagonal.
col	Color.
lwd	Line width.
cex	Size of the symbols.
ppv	Positive predictive value cutoff
npv	Negative predictive value cutoff
prevalence	Prevalence of the positive outcome
col.pvs	Color of the predictive value triangles

<code>col.ci</code>	Color of the positive and negative predictive values.
<code>lwd.ci</code>	Line width for the positive and negative predictive values.
<code>len.ci</code>	Length of the end segment for positive and negative predictive values (see arrows).
<code>boot.n</code>	Number of bootstrap replicates for the computation of the confidence interval of the specificity at NPV and of the sensitivity at PPV.
<code>conf.level</code>	Width of the confidence interval of the specificity at NPV and of the sensitivity at PPV.
<code>...</code>	parameters to be passed to plot.

Value

A list with the following elements:

- AUC: A numeric vector of length 3 containing the median, lower bound, and upper bound of the AUC.
- PPV: The positive predictive value cutoff.
- sensitivity@PPV: A numeric vector of length 3 containing the median, lower bound, and upper bound of the sensitivity at the specified PPV.
- NPV: The negative predictive value cutoff.
- specificity@NPV: A numeric vector of length 3 containing the median, lower bound, and upper bound of the specificity at the specified NPV.

Examples

```
set.seed(0)
count.neg <- count.pos <- 200
sd.neg <- sd.pos <- 1
mean.neg <- .5
mean.pos <- 1
skew.neg <- -8
skew.pos <- 0
x <- c(sn::rsn(count.neg, mean.neg, sd.neg, skew.neg),
      sn::rsn(count.pos, mean.pos, sd.pos, skew.pos))
y <- ordered(c(rep("neg", count.neg), rep("pos", count.pos)), levels=c("neg", "pos"))
r1 <- roc(y, x)
plotROC(r1, ppv=0.2, prevalence=0.05, boot.n=1e2)
skew.neg <- 0
skew.pos <- 8
x <- c(sn::rsn(count.neg, mean.neg, sd.neg, skew.neg),
      sn::rsn(count.pos, mean.pos, sd.pos, skew.pos))
y <- ordered(c(rep("neg", count.neg), rep("pos", count.pos)), levels=c("neg", "pos"))
r2 <- roc(y, x)
plotROC(r2, npv=0.995, prevalence=0.05, boot.n=1e2)
```

ppv	<i>Diagnostic performance: Positive predictive values from a ROC curve</i>
-----	--

Description

Diagnostic performance: Positive predictive values (PPV) from a ROC curve

Usage

```
ppv(roc, prevalence)
```

Arguments

roc	Object of class pROC.
prevalence	Prevalence of the endpoint in the study population.

Value

A matrix with the following columns:

- **threshold**: The thresholds used to compute the sensitivity and specificity.
- **sensitivity**: The sensitivities at the threshold.
- **specificity**: The specificities at the threshold.
- **ppv**: The positive predictive values at the threshold.
- **prevalence**: The prevalence of the endpoint in the study population, as provided in the input.

ppv.seatasp	<i>Diagnostic performance: Sensitivity from specificity, PPV and prevalence</i>
-------------	---

Description

Diagnostic performance: Sensitivity from specificity, positive predictive value (PPV) and prevalence of the endpoint

Usage

```
ppv.seatasp(sp, ppv, prevalence)
```

Arguments

sp	Specificity.
ppv	Positive predictive value.
prevalence	Prevalence of the endpoint in the study population.

Value

Sensitivity from specificity, PPV and prevalence.

ppv.spatse	<i>Diagnostic performance: Specificity from sensitivity, PPV and prevalence</i>
------------	---

Description

Diagnostic performance: Specificity from sensitivity, positive predictive value (PPV) and prevalence of the endpoint

Usage

```
ppv.spatse(se, ppv, prevalence)
```

Arguments

se	Sensitivity.
ppv	Positive predictive value.
prevalence	Prevalence of the endpoint in the study population.

Value

Specificity from sensitivity, PPV and prevalence.

pvs	<i>Diagnostic performance: Predictive values from a ROC curve</i>
-----	---

Description

Diagnostic performance: Predictive values from a ROC curve

Usage

```
pvs(
  roc = NULL,
  prevalence,
  thresholds = roc$thresholds,
  sensitivities = roc$sensitivities,
  specificities = roc$specificities
)
```

Arguments

roc	Object of class pROC.
prevalence	Prevalence of the endpoint in the study population.
thresholds	Thresholds of the ROC curve, default is roc\$thresholds.
sensitivities	Sensitivity values of the ROC curve, default is roc\$sensitivities.
specificities	Specificity values of the ROC curve, default is roc\$specificities.

Details

This function computes the positive and negative predictive values from a ROC curve, given the prevalence of the positive outcome.

Value

A data frame with the following columns:

- threshold: Thresholds of the ROC curve.
- sensitivity: Sensitivity values of the ROC curve.
- specificity: Specificity values of the ROC curve.
- prevalence: Prevalence of the positive outcome.
- ppv: Positive predictive value.
- npv: Negative predictive value.

rocperf

ROC curve performances

Description

Range of statistics associated with a ROC curve with confidence interval where applicable. This function is faster than the alternatives provided by the package pROC.

Usage

```
rocperf(
  x,
  y,
  sensitivities = NULL,
  specificities = NULL,
  conf.level = 0.95,
  fun = NULL,
  seed = 1,
  boot.n = 2000,
  median = FALSE,
  attr = FALSE,
```

```

parallel = FALSE,
simplify = TRUE,
...
)

```

Arguments

<code>x</code>	Numeric vector containing the predicted value for each observation.
<code>y</code>	Factor, numeric, logical or character vector encoding the response.
<code>sensitivities</code>	Vector of sensitivity thresholds. Default NULL.
<code>specificities</code>	Vector of specificity thresholds. Default NULL.
<code>conf.level</code>	Width of the confidence interval. Default: 0.95 (i.e., 95% CI).
<code>fun</code>	Function to compute additional statistics. Default NULL.
<code>seed</code>	Random seed for bootstrapping. Default 1.
<code>boot.n</code>	Number of bootstrap samples. Default 2e3.
<code>median</code>	If TRUE, return median bootstrap sensitivities and specificities, otherwise return observed values, otherwise the observe value is provided. Default FALSE.
<code>attr</code>	Return bootstrap results and ROC curve as attributes. Default FALSE.
<code>parallel</code>	Parallelise bootstrap. Default FALSE.
<code>simplify</code>	If TRUE, return only median for results of the function <code>fun</code> having one value across bootstraps. Default TRUE.
<code>...</code>	Additional arguments passed to <code>fun</code> if not NULL.

Details

This function computes the area under the ROC curve (AUC) and its confidence interval, the Mann-Whitney U test p-value, and the p-value for the null hypothesis that the AUC is equal to 0.5 (DeLong et al. 1988).

The function also computes the sensitivity at specified specificities and the specificity at specified sensitivities, with confidence intervals and interquartile ranges if bootstrapping is performed.

The function uses the pROC package to compute the ROC curve and confidence intervals, and it can handle parallel processing for bootstrapping.

The function returns a data frame with the computed statistics, including:

- Number of control and case patients
- Mann Whitney U test p-value
- AUC and its confidence intervals
- Sensitivity at specified specificities and their confidence intervals
- Specificity at specified sensitivities and their confidence intervals

The function `fun` must take the following arguments:

- `controls`: vector of control values
- `cases`: vector of case values

- thresholds: vector of thresholds used for the ROC curve
- sensitivities: vector of sensitivities
- specificities: vector of specificities
- ...: additional arguments

and return a named vector of values.

Value

A data frame with the following columns:

- n.control: Number of control patients
- n.case: Number of case patients
- MannWhitney.pvalue: Mann Whitney U test p-value
- AUC.pvalue: p-value for the null hypothesis that AUC=0.5
- AUC: Area under the ROC curve (point estimate)
- AUC.lCI: Lower limit of 95% confidence interval for AUC
- AUC.uCI: Upper limit of 95% confidence interval for AUC
- AUC.lQuart: Lower limit of 50% confidence interval for AUC
- AUC.uQuart: Upper limit of 50% confidence interval for AUC
- Se@SpX: Sensitivity at X% specificity
- Se@SpX.lCI: Lower limit of 95% confidence interval for sensitivity at X% specificity
- Se@SpX.uCI: Upper limit of 95% confidence interval for sensitivity at X% specificity
- Se@SpX.lQuart: Lower limit of 95% confidence interval for sensitivity at X% specificity
- Se@SpX.uQuart: Upper limit of 95% confidence interval for sensitivity at X% specificity
- Sp@SeX: Specificity at X% sensitivity
- Sp@SeX.lCI: Lower limit of 95% confidence interval for specificity at X% sensitivity
- Sp@SeX.uCI: Upper limit of 95% confidence interval for specificity at X% sensitivity
- Sp@SeX.lQuart: Lower limit of 50% confidence interval for specificity at X% sensitivity
- Sp@SeX.uQuart: Upper limit of 50% confidence interval for specificity at X% sensitivity
- Additional columns for statistics computed by the function fun if provided

data.frame with one row with computed statistics in columns.

Examples

```
fu <- function(controls,cases,thr,se,sp,...) {
  r <- pROC::roc(c(rep(0,length(controls)),
                  rep(1,length(cases))),
                c(controls,cases),
                quiet=TRUE)
  c(AUC_fun=r$auc)
}
set.seed(1)
```

```

n <- 123
y <- runif(n)<.5
x <- rnorm(n)+y*1
ans <- rocperfc(x,y,fun=fu)
ans <- rocperfc(x,y,fun=fu,
                 sensitivities=c(.5,.75,.9),
                 specificities=c(.5,.75,.9))

```

rocthreshold

rocthreshold

Description

Compute ROC sensitivity, specificity and threshold for a given cutoff.

Usage

```
rocthreshold(x, y, cut, input, prevalence = NULL, annotate = FALSE)
```

Arguments

x	either a ROC object or the predictor to build the ROC curve
y	if x is not a ROC object, the binary outcome of the ROC curve
cut	cutoff value
input	cutoff type, either sensitivity, specificity, threshold, PPV or NPV
prevalence	Prevalence of the endpoint in the study population.
annotate	if TRUE, the full ROC curve is returned as an attribute

Details

This function computes the sensitivity, specificity, threshold, PPV and NPV at a given cutoff value from a ROC curve.

NPV and PPV are computed only if a prevalence is provided as input.

If the ROC curve is not provided, it will be computed from the predictor and outcome.

Value

a vector with sensitivity, specificity, threshold, PPV and NPV at the given cutoff.

Index

`alternativeROC`, [2](#)
`alternativeROC-common-args`, [2](#)

`cinpv`, [3](#)
`cippv`, [4](#)

`npv`, [5](#)
`npv.seatse`, [5](#)
`npv.spatse`, [6](#)

`p.auc`, [6](#)
`plotROC`, [7](#)
`ppv`, [9](#)
`ppv.seatse`, [9](#)
`ppv.spatse`, [10](#)
`pvs`, [10](#)

`rocperf`, [11](#)
`rocthreshold`, [14](#)