Package ‘ThresholdROC’

May 14, 2026

Type Package

Title Optimum Threshold Estimation

Version 2.9.6

Date 2026-05-14

Description Functions that provide point and interval estimations of optimum thresholds for continuous diagnostic tests. The methodology used is based on minimizing an overall cost function in the two- and three-state settings. We also provide functions for sample size determination and estimation of diagnostic accuracy measures. We also include graphical tools. The statistical methodology used here can be found in Perez-Jaume et al (2017) <doi:10.18637/jss.v082.i04> and in Skaltsa et al (2010, 2012) <doi:10.1002/bimj.200900294>, <doi:10.1

License GPL (>= 2)

Depends R (>= 3.1)

Imports MASS, numDeriv, pROC, ks

LazyData TRUE

NeedsCompilation no

Author Sara Perez-Jaume [aut, cre] (ORCID:
<`https://orcid.org/0000-0002-9805-4810`>),
Natalia Pallares [aut] (ORCID: <`https://orcid.org/0000-0002-1462-379X`>),
Konstantina Skaltsa [aut]

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

Repository CRAN

Date/Publication 2026-05-14 21:40:23 UTC

Contents

`AD` *Alzheimer's disease data*

Description

This `data.frame`, obtained by Konstantina Skaltsa from Kapaki *et al.* (2003), contains measurements from the tau protein located in the cerebrospinal fluid measured in pg/mL (`Tau` variable) of 49 control subjects (`Status==0`) and 49 patients with Alzheimer's disease (`Status==1`). A column indicating the identifier of the subject is also included (`id` variable).

Usage

```
data("AD")
```

References

Kapaki E, Paraskevas G, Zalonis I, Zournas C. (2003). CSF Tau Protein and beta-amyloid (1-42) in Alzheimer's Disease diagnosis: Discrimination from Normal Ageing and the Other Dementias in the Greek Population. European Journal of Neurology, 10, 119-128.

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

`chemo` *Response to chemotherapy data*

Description

This `data.frame` contains the data from a study on large primary breast cancer, in which the aim was to evaluate teh use of PET/CT imagining in predicting response to neoadjuvant chemotherapy. The study recruited 50 patients. The following variables have been collected:

- `id` Patient identifier

- `SUV` percent change in maximum standardised uptake value (SUV) (%) between the baseline evaluation (before treatment) and the evaluation after two cycles of neoadjuvant chemotherapy, quantified using PET/CT imaging. The following formula was used to calculate this variable:

  $$\frac{\mathrm{SUV}_b - \mathrm{SUV}_2}{\mathrm{SUV}_b} \cdot 100,$$

  where SUV_b is the maximum SUV in the baseline evaluation and SUV_2 is the maximum SUV after two cycles
- `resp` Response evaluated after the completion of the neoadjuvant chemotherapy, in accordance with the RECIST guidelines: complete response (lesions completely disappeared), partial response (lesions showed at least a 30% decrease in size), stable disease (otherwise)

Usage

```
data("chemo")
```

References

Duch J, Fuster D, Munoz M, Fernandez P, Paredes P, Fontanillas M, Guzman F, Rubi S, Lomena F, Pons F. (2009). 18 F-FDG PET/CT for Early Prediction of Response to Neoadjuvant Chemotherapy in Breast Cancer. European Journal of Nuclear Molecular Imaging, 36, 1551-1557.

Skaltsa K, Jover L, Fuster D, Carrasco JL. (2012). Optimum threshold estimation based on cost function in a multistate diagnostic setting. Statistics in Medicine, 31:1098-1109.

`diagnostic` *Diagnostic accuracy measures of binary diagnostic tests*

Description

This function estimates common accuracy measures for binary diagnostic tests involving 2x2 contingency tables of classification results (usually, test outcome versus status tables).

Usage

```
diagnostic(tab, method = c("par", "exact", "auto"),
           casecontrol = FALSE, p = NULL, conf.level = 0.95)
```

Arguments

`tab`	an object of class `table` or `matrix` in the following form: TP FP FN TN where TP=number of true positives, FP=number of false positives, FN=number of false negatives and TN=number of true negatives; that is, a table where the first row corresponds to the positive tests and the second row to the negative tests; the first column corresponds to the diseased subjects and the second one to the healthy individuals. `dim(tab)` should be `c(2, 2)`.

`method`	method for calculating the confidence intervals for sensitivity, specificity, predictive values, accuracy and error rate. The user can choose between `"par"` (parametric), `"exact"` (exact) and `"auto"`, which chooses automatically between the first two options. The eact method is recommended for small sample sizes. Default, `"par"`. The user can specify just the initial letters. For more information, see **Details**.
`casecontrol`	were data collected in a case-control study? Default, `FALSE`.
`p`	disease prevalence (only when `casecontrol = TRUE`; otherwise, this parameter is ignored).
`conf.level`	confidence level for the confidence intervals. Default, 0.95 (95%).

Details

For details about the expressions for the statistical measures calculated by this function, see **References**.

Since sensitivity, specificity, predictive values, accuracy and error rate are proportions, their confidence intervals are calculated using the functions `prop.test` (if `method = "par"`) and `binom.test` (if `method = "exact"`) from the `stats` package.

Confidence intervals for the likelihood ratios are calculated using the formulas proposed in Zhou et al (2002), Section 4.1.3. Furthermore, when likelihood ratios can not be calculated due to division by 0, the following correction is applied: 0.5 units are added to `tab`.

Confidence intervals for the odds ratio are calculated using the formulas proposed in Zhou et al (2002), Section 4.1.4. The same correction described before is applied when odds ratio can not be calculated due to division by 0.

Confidence intervals for the Youden index are calculated using the expression

$$\mathrm{CI}_{1-\alpha} = (Y - z_{1-\alpha/2} \cdot \mathrm{Var}(Y), Y + z_{1-\alpha/2} \cdot \mathrm{Var}(Y)),$$

where Y is the Youden index estimate, $z_{1-\alpha/2}$ is the $1-\alpha/2$ quantile of a $N(0,1)$ distribution and $\mathrm{Var}(Y)$ is the variance of the Youden index estimator, which is calculated as $\mathrm{Var}(\mathrm{Sensitivity}) + \mathrm{Var}(\mathrm{Specificity})$.

Value

A `data.frame` with ten rows and three columns containing the point estimate and confidence intervals for the following statistical measures: sensitivity, specificity, positive predictive value, negative predictive value, positive likelihood ratio, negative likelihood ratio, odds ratio, Youden index, accuracy and error rate.

Author(s)

Sara Perez-Jaume, Natalia Pallares

References

Youden, WJ. (1950). Index for rating diagnostic tests. Cancer 3:32-35.

Zhou XH, Obuchowski NA and McClish DK. (2002). Statistical methods in diagnostic medicine. John Wiley and sons.

See Also

thres2

Examples

```
# example 1 (Zhou et al, 2002)
japan <- matrix(c(56, 23, 6, 78), ncol = 2, byrow = TRUE)
colnames(japan) <- c("D", "nD")
rownames(japan) <- c("+", "-")
japan
p <- 0.196 # disease prevalence
diagnostic(japan, "par", casecontrol = TRUE, p = p)

# example 2
table <- matrix(c(22, 2, 3, 3), ncol = 2, byrow = TRUE)
diagnostic(table, "par")
diagnostic(table, "exact")
diagnostic(table, "auto")

# example 3
table2 <- matrix(c(22, 2, 0, 3), ncol = 2, byrow = TRUE)
diagnostic(table2, "exact")

# example 4
# generate a random sample of diseased and non-diased subjects
n1 <- 100
n2 <- 100
set.seed(1234)
par1.1 <- 0
par1.2 <- 1
par2.1 <- 2
par2.2 <- 1
k1 <- rnorm(n1, par1.1, par1.2) # non-diseased
k2 <- rnorm(n2, par2.1, par2.2) # diseased
# threshold estimation
rho <- 0.2 # prevalence
thres <- thres2(k1, k2, rho, method = "equal", ci.method = "delta")$T$thres
# diagnostic test using the threshold estimate
marker <- c(k1, k2) # biomarker
condition <- c(rep("nD", length(k1)), rep("D", length(k2))) # condition
test <- ifelse(marker<thres, "-", "+") # test outcome according to thres
# build the table
table3 <- table(test, condition)[2:1, ]
# diagnostic test
round(diagnostic(table3), 3)
```

`lines.thres2` *Add threshold lines to a plot (two-state setting)*

Description

The function includes vertical lines for the threshold and confidence interval in a plot created with `plot.thres2()`.

Usage

```
## S3 method for class 'thres2'
lines(x, ci = TRUE, which.boot = c("norm", "perc"),
      col = 1, lty = c(1, 2), lwd = 1, ...)
```

Arguments

`x`	an object of class `thres2`.
`ci`	should the confidence interval be plotted? Default, `TRUE`. No confidence interval will be plotted if `x` does not contain one (that is, `x$CI` is `NULL`).
`which.boot`	in case `x` contains confidence intervals calculated by bootstrapping, which one should be printed? The user can choose between `"norm"` (based on normal distribution) or `"perc"` (based on percentiles). Default, `"norm"`. This argument is ignored if the confidence intervals were calculated by the delta method.
`col`	color for the threshold and its corresponding confidence interval. Default, 1.
`lty`	a 2-dimensional vector containing: `lty[1]`: line type for the threshold `lty[2]`: line type for the confidence interval Default, `c(1, 2)`. If `length(lty)` is not 2, `lty` will be recycled.
`lwd`	line width for the threshold and its corresponding confidence interval. Default, 1.
`...`	further arguments to be passed to `abline()`.

Value

With a `plot.thres2` open, this function adds lines for the required threshold.

References

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

See Also

`thres2`, `plot.thres2`

Examples

```
n1 <- 100
n2 <- 100
set.seed(1234)
par1.1 <- 0
```

```
par1.2 <- 1
par2.1 <- 2
par2.2 <- 1
rho <- 0.2
k1 <- rnorm(n1, par1.1, par1.2) # non-diseased
k2 <- rnorm(n2, par2.1, par2.2) # diseased

thres <- thres2(k1, k2, rho, method = "eq", ci.method = "d")
plot(thres, col = c(1, 2, 4), lwd = c(2, 2, 1), leg.pos = "topright")
thresunequal <- thres2(k1, k2, rho, method = "unequal", ci = FALSE)
lines(thresunequal, col = 3) # almost the same; no confidence interval

## Not run:
  thresboot <- thres2(k1, k2, rho, method = "emp", ci.method = "b")
  lines(thresboot, col = 5, which.boot = "norm")

## End(Not run)
```

`lines.thres3` *Add threshold lines to a plot (three-state setting)*

Description

The function includes vertical lines for the thresholds and confidence intervals in a plot created with `plot.thres3()`.

Usage

```
## S3 method for class 'thres3'
lines(x, ci = TRUE, which.boot = c("norm", "perc"),
      col = 1, lty = c(1, 2), lwd = 1, ...)
```

Arguments

`x`	an object of class `thres3`.
`ci`	should the confidence intervals be plotted? Default, `TRUE`. No confidence intervals will be plotted if `x` does not contain one (that is, `x$CI` is `NULL`).
`which.boot`	in case `x` contains confidence intervals calculated by bootstrapping, which one should be printed? the user can choose between `"norm"` (based on normal distribution) or `"perc"` (based on percentiles). Default, `"norm"`. This argument is ignored if parametric confidence intervals were calculated.
`col`	color for the thresholds and their corresponding confidence intervals. Default, 1.
`lty`	a 2-dimensional vector containing: `lty[1]`: line type for the thresholds `lty[2]`: line type for the confidence intervals Default, `c(1, 2)`. If `length(lty)` is not 2, `lty` will be recycled.

`lwd`	line width for the thresholds and their corresponding confidence intervals. Default, 1.
`...`	further arguments to be passed to `abline()`.

Value

With a `plot.thres3` open, this function adds lines for the required threshold estimates.

References

Skaltsa K, Jover L, Fuster D, Carrasco JL. (2012). Optimum threshold estimation based on cost function in a multistate diagnostic setting. Statistics in Medicine, 31:1098-1109.

See Also

`thres3`, `plot.thres3`

Examples

```
set.seed(1234)
n <- 100
k1 <- rlnorm(n)
k2 <- rnorm(n, 3, 1)
k3 <- rnorm(n, 5, 1)
rho <- c(1/3, 1/3, 1/3)

# assuming trinormality
start <- c(mean(k1), mean(k3))
thres1 <- thres3(k1, k2, k3, rho, dist1 = "norm", dist2 = "norm",
                 dist3 = "norm", start = start, ci.method = "param")

# not assuming trinormality
start2 <- c(0.05, 0.6, 0.5, 0.95)
set.seed(2014)
thres2 <- thres3(k1, k2, k3, rho, start = start2, B = 1000,
                ci.method = "boot", dist1 = "lnorm", dist2 = "norm",
                dist3 = "norm")
plot(thres2, leg.pos = "topright", leg.cex = 0.8, col = 1:4)
lines(thres1, col = 5)
```

`plot.thres2` *Threshold and density plot (two-state setting)*

Description

This function provides a graph including the sample densities (diseased and non-diseased populations), the threshold and its confidence interval.

Usage

```
## S3 method for class 'thres2'
plot(x, bw = c("nrd0", "nrd0"), ci = TRUE,
     which.boot = c("norm", "perc"), col = c(1, 2, 3),
     lty = c(1, 1, 1, 2), lwd = c(1, 1, 1),
     legend = TRUE, leg.pos = "topleft", leg.cex = 1,
     xlim = NULL, ylim = NULL,
     main = paste0("Threshold estimate ", ifelse(ci, "and CI ", ""),
                   "(method ", x$T$method, ")"),
     xlab = "Biomarker", ...)
```

Arguments

`x`	an object of class `thres2`.
`bw`	vector containing the bandwith for the non-diseased sample in the first position and the bandwith for the diseased sample in the second position (to be passed to `density()`). Default, `c("nrd0", "nrd0")`.
`ci`	should the confidence interval be plotted? Default, `TRUE`. No confidence interval will be plotted if `x` does not contain one (that is, `x$CI` is `NULL`).
`which.boot`	in case `x` contains confidence intervals calculated by bootstrapping, which one should be printed? The user can choose between `"norm"` (based on normal distribution) or `"perc"` (based on percentiles). Default, `"norm"`. This argument is ignored if the confidence intervals were calculated by the delta method.
`col`	a 3-dimensional vector containing: `col[1]`: color for the density of the non-diseased sample `col[2]`: color for the density of the diseased sample `col[3]`: color for the threshold and its corresponding confidence interval Default, `c(1, 2, 3)`. If `length(col)` is not 3, `col` will be recycled.
`lty`	a 4-dimensional vector containing: `lty[1]`: line type for the density of the non-diseased sample `lty[2]`: line type for the density of the diseased sample `lty[3]`: line type for the threshold `lty[4]`: line type for the confidence interval Default, `c(1, 1, 1, 2)`. If `length(lty)` is not 4, `lty` will be recycled.
`lwd`	a 3-dimensional vector containing: `lwd[1]`: line width for the density of the non-diseased sample `lwd[2]`: line width for the density of the diseased sample `lwd[3]`: line width for the threshold and its corresponding confidence interval Default, `c(1, 1, 1)`. If `length(lwd)` is not 3, `lwd` will be recycled.
`legend`	logical asking if an automatic legend should be added to the graph. Default, `TRUE`.
`leg.pos`	position of the legend. Default, `"topleft"`. Ignored if `legend=FALSE`.
`leg.cex`	number that reescales the size of the legend. Ignored if `legend=FALSE`. Default, 1.

`xlim`	2-dimensional vector indicating the lower and upper limits for x-axis. Default value (NULL) sets those limits automatically.
`ylim`	2-dimensional vector indicating the lower and upper limits for y-axis. Default value (NULL) sets those limits automatically.
`main`, `xlab`, `...`	further arguments to be passed to `plot()`.

Value

Estimates of the density functions for both samples and vertical lines representing the threshold and its confidence limits are drawn.

References

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

See Also

`thres2`, `lines.thres2`

Examples

```
n1 <- 100
n2 <- 100
set.seed(1234)
par1.1 <- 0
par1.2 <- 1
par2.1 <- 2
par2.2 <- 1
rho <- 0.2
k1 <- rnorm(n1, par1.1, par1.2) # non-diseased
k2 <- rnorm(n2, par2.1, par2.2) # diseased

thres <- thres2(k1, k2, rho, method = "eq", ci.method = "d")
plot(thres, col = c(1, 2, 4), lwd = c(2, 2, 1), leg.pos = "topright")
```

`plot.thres3` *Thresholds and density plot (three-state setting)*

Description

This function provides a graph including the three sample densities, the thresholds and their confidence intervals.

Usage

```
## S3 method for class 'thres3'
plot(x, bw = c("nrd0", "nrd0", "nrd0"), ci = TRUE,
     which.boot = c("norm", "perc"), col = c(1, 2, 3, 4),
     lty = c(1, 1, 1, 1, 2), lwd = c(1, 1, 1, 1),
     legend = TRUE, leg.pos = "topleft", leg.cex = 1,
     xlim = NULL, ylim = NULL,
     main = paste0("Threshold estimates", ifelse(ci, " and CIs", "")),
     xlab = "", ...)
```

Arguments

`x`	an object of class `thres3`.
`bw`	vector containing the bandwith for the first sample in the first position, the bandwith for the second sample in the second position and the bandwith for the third sample in the third position (to be passed to `density()`). Default, `c("nrd0", "nrd0", "nrd0")`.
`ci`	should the confidence intervals be plotted? Default, `TRUE`. No confidence intervals will be plotted if `x` does not contain one (that is, `x$CI` is `NULL`).
`which.boot`	in case `x` contains confidence intervals calculated by bootstrapping, which one should be printed? The user can choose between `"norm"` (based on normal distribution) or `"perc"` (based on percentiles). Default, `"norm"`. This argument is ignored if parametric confidence intervals were calculated.
`col`	a 4-dimensional vector containing: `col[1]`: color for the density of the first sample `col[2]`: color for the density of the second sample `col[3]`: color for the density of the third sample `col[4]`: color for the thresholds and their corresponding confidence intervals Default, `c(1, 2, 3, 4)`. If `length(col)` is not 4, `col` will be recycled.
`lty`	a 5-dimensional vector containing: `lty[1]`: line type for the density of the first sample `lty[2]`: line type for the density of the second sample `lty[3]`: line type for the density of the third sample `lty[4]`: line type for the thresholds `lty[5]`: line type for the confidence intervals Default, `c(1, 1, 1, 1, 2)`. If `length(lty)` is not 5, `lty` will be recycled.
`lwd`	a 4-dimensional vector containing: `lwd[1]`: line width for the density of the first sample `lwd[2]`: line width for the density of the second sample `lwd[3]`: line width for the density of the third sample `lwd[4]`: line width for the thresholds and their corresponding confidence intervals Default, `c(1, 1, 1, 1)`. If `length(lwd)` is not 4, `lwd` will be recycled.

`legend`	logical asking if an automatic legend should be added to the graph. Default, `TRUE`.
`leg.pos`	position of the legend. Default, `"topleft"`. Ignored if `legend=FALSE`.
`leg.cex`	number that reescales the size of the legend. Ignored if `legend=FALSE`. Default, 1.
`xlim`	2-dimensional vector indicating the lower and upper limits for x-axis. Default value (NULL) sets those limits automatically.
`ylim`	2-dimensional vector indicating the lower and upper limits for y-axis. Default value (NULL) sets those limits automatically.
`main`, `xlab`, `...`	further arguments to be passed to `plot()`.

Value

Estimates of the density functions for the three samples and vertical lines representing the thresholds and their confidence limits are drawn.

References

Skaltsa K, Jover L, Fuster D, Carrasco JL. (2012). Optimum threshold estimation based on cost function in a multistate diagnostic setting. Statistics in Medicine, 31:1098-1109.

See Also

`thres3`, `lines.thres3`

Examples

```
set.seed(1234)
n <- 100
k1 <- rlnorm(n)
k2 <- rnorm(n, 3, 1)
k3 <- rnorm(n, 5, 1)
rho <- c(1/3, 1/3, 1/3)

# assuming trinormality
start <- c(mean(k1), mean(k3))
thres <- thres3(k1, k2, k3, rho, dist1 = "norm", dist2 = "norm",
                dist3 = "norm", start = start, ci.method = "param")
plot(thres, leg.pos = "topright")

# not assuming trinormality
thres <- thres3(k1, k2, k3, rho, dist1 = "lnorm", dist2 = "norm",
                dist3 = "norm", ci.method = "boot")
plot(thres, leg.pos = "topright", which.boot = "perc")
```

`plotCostROC` *Plot cost function and ROC curve*

Description

This function plots the ROC curve and the cost function associated to the disease prevalence and costs.

Usage

```
plotCostROC(x, type = "l", which.plot, ...)
```

Arguments

`x`	an object of class `thres2` or `thres3`. See **Details**.
`type`	1-character string giving the type of plot desired. Default, `"l"`. See **Details**.
`which.plot`	which plot should be produced? The user can choose between `"both"` (default, which plots both cost function and ROC curve), `"cost"` (only plots cost function) or `"roc"` (only plots ROC curve). It only applies to objects of class `thres2`, therefore this argument has no effect if `x` is an object of class `thres3`.
`...`	other graphical parameters to be passed to `plot()`.

Details

If the argument `x` was constructed with `method="empirical"`, the argument `extra.info` should be switched to `TRUE` (this only applies when `x` is an object of class `thres2`).

For parameter `type` the following values are possible: `"p"` for points, `"l"` for lines, `"b"` for both points and lines, `"c"` for empty points joined by lines, `"o"` for overplotted points and lines, `"s"` and `"S"` for stair steps and `"h"` for histogram-like vertical lines. Finally, `"n"` does not produce any points or lines.

Value

When `x` is an object of class `thres2`, two plots are produced. The first one shows the cost function with the cost minimising threshold in red. The second one is the step ROC curve with the sensitivity and specificity achieved in red. If `method = "empirical"` is used when building `x`, empirical cost function and ROC curve are plotted. If `method = "smooth"` is used when building `x`, smooth cost function and ROC curve are plotted.

When `x` is an object of class `thres3`, two plots are produced. The first one shows the cost function C(T1) with the first cost minimising threshold in red. The second one shows the cost function C(T2) with the second cost minimising threshold in red.

Note

This function uses the `plot()` function and further arguments can be set to customise the resulting plot.

References

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

Skaltsa K, Jover L, Fuster D, Carrasco JL. (2012). Optimum threshold estimation based on cost function in a multistate diagnostic setting. Statistics in Medicine, 31:1098-1109.

See Also

`thres2`

Examples

```
## Not run:
# example 1: x is an object of class 'thres2'
n1 <- 100
n2 <- 100
set.seed(19998)
par1.1 <- 0
par1.2 <- 1
par2.1 <- 2
par2.2 <- 1
rho <- 0.3
k1 <- rnorm(n1, par1.1, par1.2) # non-diseased
k2 <- rnorm(n2, par2.1, par2.2) # diseased
x <- thres2(k1, k2, rho, method = "emp", ci.method = "boot",
            extra = TRUE)

par(mfrow=c(1, 2))
plotCostROC(x)


# example 2: x is an object of class 'thres3'
set.seed(2015)
n <- 100
k1 <- rlnorm(n)
k2 <- rnorm(n, 3, 1)
k3 <- rnorm(n, 5, 1)
rho <- c(1/3, 1/3, 1/3)
y <- thres3(k1, k2, k3, rho, B = 1000, ci.method = "boot",
            dist1 = "lnorm", dist2 = "norm", dist3 = "norm")

par(mfrow=c(1, 2))
plotCostROC(y)

## End(Not run)
```

`secondDer2` *Second partial derivative of the cost function (two-state setting)*

Description

This function calculates the second partial derivative of the cost function at a given threshold.

Usage

```
secondDer2(x)
```

Arguments

`x`	an object of class `thres2`.

Details

This function evaluates the second derivative of the cost function at the threshold estimate so that the user can assess if this is positive (meaning that the estimation of the threshold leads to a minimum in the cost function) or close to zero and hence the minimum of the cost function does not exist. A closed formula is given when assuming binormality, otherwise the second derivative is evaluated numerically in the threshold estimate using functions from the package `numDeriv`.

Value

The value of the second derivative of the cost function at the given threshold.

References

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

See Also

`thres2`

Examples

```
n1 <- 100
n2 <- 100
set.seed(1234)
par1.1 <- 0
par1.2 <- 1
par2.1 <- 2
par2.2 <- 1
rho <- 0.2
k1 <- rnorm(n1, par1.1, par1.2) # non-diseased
k2 <- rnorm(n2, par2.1, par2.2) # diseased
x <- thres2(k1, k2, rho, method = "equal", ci.method = "delta")
secondDer2(x)
```

`secondDer3` *Second partial derivative of the cost function (three-state setting)*

Description

This function calculates the second partial derivatives of the cost function at a given pair of thresholds.

Usage

```
secondDer3(x)
```

Arguments

`x`	an object of class `thres3`.

Details

This function evaluates the second partial derivatives of the cost function at the threshold estimates so that the user can assess if these are positive (meaning that the estimation of the thresholds leads to a minimum in the cost function) or close to zero and hence the minimum of the cost function does not exist. A closed formula is given when assuming trinormality, otherwise the second derivatives are evaluated numerically in the threshold estimates using functions from the package `numDeriv`.

Value

The value of the second derivative of the cost function at the given thresholds.

References

Skaltsa K, Jover L, Fuster D, Carrasco JL. (2012). Optimum threshold estimation based on cost function in a multistate diagnostic setting. Statistics in Medicine, 31:1098-1109.

See Also

`thres3`

Examples

```
set.seed(1234)
n <- 100
k1 <- rlnorm(n)
k2 <- rnorm(n, 3, 1)
k3 <- rnorm(n, 5, 1)
rho <- c(1/3, 1/3, 1/3)
start <- c(mean(k1), mean(k3))
x <- thres3(k1, k2, k3, rho, dist1 = "norm", dist2 = "norm",
            dist3 = "norm", start = start, ci.method = "param")

secondDer3(x)
```

SS *Sample size estimation (two-state setting)*

Description

Estimates the sample size and the optimum sample size ratio needed for a given width, costs, disease prevalence and significance level under the assumption of binormality.

Usage

```
SS(par1.1, par1.2, par2.1, par2.2=NULL, rho, width,
   costs = matrix(c(0, 0, 1, (1-rho)/rho), 2, 2, byrow = TRUE),
   R = NULL, var.equal = FALSE, alpha = 0.05)
```

Arguments

`par1.1`	healthy population mean.
`par1.2`	healthy population standard deviation.
`par2.1`	diseased population mean.
`par2.2`	diseased population standard deviation. It can be omitted when assuming equal variances (that is, when `var.equal=TRUE`) and in this situation the common variance is assumed to be equal to `par1.2`.
`rho`	disease prevalence.
`width`	desired interval width.
`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 are a combination of costs that yields R=1, which is equivalent to the Youden index method (for details about this concept, see **References**). It must be set to `NULL` if the user prefers to set `R` (see next argument).
`R`	if the cost matrix `costs` is not set, 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`).
`var.equal`	a logical variable indicating whether to use equal variances. Default, `FALSE`.
`alpha`	significance level for the confidence interval. Default, 0.05.

Value

an object of class `SS` which is a list with eight components:

`ss2`	sample size for the diseased group
`ss1`	sample size for the healthy group
`epsilon`	sample size ratio between non-diseased and diseased subjects
`width`	width of the confidence interval provided by the user

`alpha`	significance level provided by the user
`costs`	cost matrix provided by the user
`R`	R term, the product of the non-disease odds and the cost ratio (for further details about this concept, see **References**)
`prev`	disease prevalence provided by the user

References

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

Examples

```
par1.1 <- 0
par1.2 <- 1
par2.1 <- 2
par2.2 <- 1
rho <- 0.3
width <- 0.5
SS(par1.1, par1.2, par2.1, par2.2, rho, width, var.equal = TRUE)
```

`thres2` *Threshold point estimation and confidence intervals (two-state setting)*

Description

This function calculates the threshold estimate and its corresponding confidence interval in a two-state setting.

Usage

```
thres2(k1, k2, rho,
       costs = matrix(c(0, 0, 1, (1 - rho)/rho), 2, 2, byrow = TRUE),
       R = NULL,
       method = c("equal", "unequal", "empirical", "smooth", "parametric"),
       dist1 = NULL, dist2 = NULL, ci = TRUE, ci.method = c("delta", "boot"),
       B = 1000, alpha = 0.05, extra.info = FALSE, na.rm = FALSE,
       q1 = 0.05, q2 = 0.95)
```

Arguments

`k1`	vector containing the healthy sample values.
`k2`	vector containing the diseased sample values.
`rho`	disease prevalence.

`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 are a combination of costs that yields R=1, which is equivalent to the Youden index method (for details about this concept, see **References**). It must be set to `NULL` if the user prefers to set `R` (see next argument).
`R`	if the cost matrix `costs` is not set, 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`).
`method`	method used in the estimation. The user can specify just the initial letters. Default, `"equal"`. See **Details** for more information about the methods available.
`dist1`	distribution to be assumed for the healthy population. See **Details**.
`dist2`	distribution to be assumed for the diseased population. See **Details**.
`ci`	should a confidence interval be calculated? Default, `TRUE`. The user can set it to `FALSE` to supress the calculation of any confidence interval (in that case, arguments `ci.method`, `B` and `alpha` are ignored).
`ci.method`	method to be used for the confidence intervals calculation. The user can specify just the initial letters. Default, `"delta"`. See **Details** for more information about the methods available.
`B`	number of bootstrap resamples when `ci.method = "boot"`. Otherwise, ignored. Default, 1000.
`alpha`	significance level for the confidence interval. Default, 0.05.
`extra.info`	when using `method="empirical"`, if set to `TRUE` the function returns extra information about the calculation of the threshold. Ignored when `method` is not `"empirical"`. Default, `FALSE`.
`na.rm`	a logical value indicating whether `NA` values in `k1` and `k2` should be stripped before the computation proceeds. Default, `FALSE`.
`q1`	probability of the left distribution in order to determine a low quantile when `method="parametric"` (ignored otherwise). Default, 0.05.
`q2`	probability of the right distribution in order to determine a high quantile when `method="parametric"` (ignored otherwise). Default, 0.95.

Details

For parameter `method` the user can choose between `"equal"` (assumes binormality and equal variances), `"unequal"` (assumes binormality and unequal variances), `"empirical"` (leaves out any distributional assumption), `"smooth"` (leaves out any distributional assumption, but uses a kernel to estimate the densities) or `"parametric"` (based on the distribution assumed for the two populations).

Parameters `dist1` and `dist2` can be chosen between the following 2-parameter distributions: `"beta"`, `"cauchy"`, `"chisq"` (chi-squared), `"gamma"`, `"lnorm"` (lognormal), `"logis"` (logistic), `"norm"` (normal) and `"weibull"`. Notice that `dist1` and `dist2` are only needed when `method = "parametric"`.

For parameter `ci.method` the user can choose between `"delta"` (delta method is used to estimate the threshold standard error assuming a binormal underlying model) or `"boot"` (the confidence interval is calculated by bootstrap).

Value

An object of class `thres2`, which is a list with two components:

. `T`, which is a list of at least seven components:

- `thres` threshold estimate.
- `prev` disease prevalence provided by the user.
- `costs` cost matrix provided by the user.
- `R` R term, the product of the non-disease odds and the cost ratio (for further details about this concept, see **References**).
- `method` method used in the estimation.
- `k1` vector containing the healthy sample values provided by the user.
- `k2` vector containing the diseased sample values provided by the user.

When `method = "empirical"`, `T` also contains:

- `sens` sensitivity obtained.
- `spec` specificity obtained.
- `cost` the minimum cost associated with `T$thres`.
- `tot.thres` vector of possible thresholds. Only if `extra.info = TRUE`.
- `tot.cost` vector of empirical costs. Only if `extra.info = TRUE`.
- `tot.spec.c` complementary of the vector of empirical specificities (1-spec). Only if `extra.info = T`.
- `tot.sens` vector of empirical sensitivities. Only if `extra.info = TRUE`.

When `method = "parametric"`, `T` also contains:

- `dist1` distribution assumed for the healthy population.
- `dist2` distribution assumed for the diseased population.
- `pars1` a numeric vector containing the estimation of the parameters of `dist1`.
- `pars2` a numeric vector containing the estimation of the parameters of `dist2`.

. `CI`, a list of five components when `ci.method = "delta"`:

- `lower` the lower limit of the confidence interval.
- `upper` the upper limit of the confidence interval.
- `se` the standard error used in the calculation of the confidence interval.
- `alpha` significance level provided by the user.
- `ci.method` method used for the confidence intervals calculation.

A list of eight components when `ci.method = "boot"`:

- `low.norm` the lower limit of the bootstrap confidence interval based on the normal distribution.
- `up.norm` the upper limit of the bootstrap confidence interval based on the normal distribution.
- `se` the bootstrap standard error used in the calculation of the confidence interval based on the normal distribution.

- `low.perc` the lower limit of the bootstrap confidence interval based on percentiles.
- `up.perc` the upper limit of the bootstrap confidence interval based on percentiles.
- `alpha` significance level provided by the user.
- `B` number of bootstrap resamples used.
- `ci.method` method used for the confidence intervals calculation.

When `ci = FALSE`, `CI` is `NULL`.

Note

It is assumed that `k1` is the sample with lower values. If that is not the case, `k1` and `k2` (and the corresponding parameters) are exchanged.

References

Efron B, Tibshirani RJ. (1993). An introduction to the bootstrap, Chapman & Hall.

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

See Also

`thresTH2`, `plot.thres2`, `lines.thres2`

Examples

```
# example 1
n1 <- 100
n2 <- 100
set.seed(1234)
par1.1 <- 0
par1.2 <- 1
par2.1 <- 2
par2.2 <- 1
rho <- 0.2
k1 <- rnorm(n1, par1.1, par1.2) # non-diseased
k2 <- rnorm(n2, par2.1, par2.2) # diseased

thres2(k1, k2, rho, method = "eq", ci.method = "d")
thres2(k1, k2, rho, method = "uneq", ci.method = "d")
# specify R instead of (default) costs
thres2(k1, k2, rho, costs = NULL, R = 2, method = "uneq", ci.method = "d")
## Not run:
thres2(k1, k2, rho, method = "empirical", ci.method = "b")

# example 2
set.seed(1234)
k1 <- rnorm(50, 10, 3)
k2 <- rlnorm(55)
rho <- 0.3
```

```
thres2(k1, k2, rho, method = "param", ci.method = "boot",
       dist1 = "norm", dist2 = "lnorm")

## End(Not run)

# supress confidence intervals calculation
thres2(k1, k2, rho, method="equal", ci=FALSE)
thres2(k1, k2, rho, method="empirical", ci=FALSE)

# example 3
n1 <- 100
n2 <- 100
set.seed(1234)
par1.1 <- 0
par1.2 <- 1
par2.1 <- 2
par2.2 <- 1
rho <- 0.2
k1 <- rnorm(n1, par1.1, par1.2) # non-diseased
k2 <- rnorm(n2, par2.1, par2.2) # diseased
## Not run:
thres2(k1, k2, rho, method="smooth", ci.method="b")

## End(Not run)
```

`thres3` *Threshold point estimation and confidence intervals (three-state setting)*

Description

This function calculates the threshold estimates and their corresponding confidence intervals in a three-state setting.

Usage

```
thres3(k1, k2, k3, rho,
       costs = matrix(c(0, 1, 1, rho[1]/rho[2], 0, rho[3]/rho[2], 1, 1, 0),
                      3, 3, byrow = TRUE), dist1 = "norm", dist2 = "norm",
       dist3 = "norm", start = NULL, ci = TRUE,
       ci.method = c("param", "boot"), B = 1000, alpha = 0.05,
       na.rm = FALSE)
```

Arguments

`k1`	vector containing the first sample values.
`k2`	vector containing the second sample values.
`k3`	vector containing the third sample values.

`rho`	3-dimensional vector of prevalences.
`costs`	cost matrix. Costs should be entered as a 3x3 matrix, where the first row corresponds to the costs associated with the classification of subjects in state 1 (C11, C12 and C13), second row corresponds to the costs associated with the classification of subjects in state 2 (C21, C22 and C23) and the third row corresponds to the costs associated with classification of subjects in state 3 (C31, C32, C33), where Cij is the cost of classifying an individual of class i as class j. Default cost values are a combination of costs that leads to the same thresholds as the Youden index method (see **References** for details).
`dist1`	distribution to be assumed for the first population. Default, `"norm"`. See **Details**.
`dist2`	distribution to be assumed for the second population. Default, `"norm"`. See **Details**.
`dist3`	distribution to be assumed for the third population. Default, `"norm"`. See **Details**.
`start`	when the three distributions `dist1`, `dist2` and `dist3` are `"norm"`, a 2-dimensional vector containing starting values for the thresholds. The authors recommend to use the mean of the distribution with lower values and the mean of the distribution with higher values. If any distribution is not `"norm"`, this parameter is ignored. See **Details**.
`ci`	should a confidence interval be calculated? Default, `TRUE`. The user can set it to `FALSE` to supress the calculation of any confidence interval (in that case, arguments `ci.method`, `B` and `alpha` are ignored).
`ci.method`	method to be used for the confidence intervals calculation. The user can specify just the initial letters. Default, `"param"`. See **Details**.
`B`	number of bootstrap resamples when `ci.method = "boot"`. Otherwise, ignored. Default, 1000.
`alpha`	significance level for the confidence interval. Default, 0.05.
`na.rm`	a logical value indicating whether `NA` values in `k1`, `k2` and `k3` should be stripped before the computation proceeds. Default, `FALSE`.

Details

Parameters `dist1`, `dist2` and `dist3` can be chosen between the following 2-parameter distributions: `"beta"`, `"cauchy"`, `"chisq"` (chi-squared), `"gamma"`, `"lnorm"` (lognormal), `"logis"` (logistic), `"norm"` (normal) and `"weibull"`.

For parameter `ci.method` the user can choose between `"param"` (parametric confidence intervals are calculated when assuming a trinormal underlying model) and `"boot"` (the confidence intervals are calculated by bootstrap).

When at least one of the distributions is not `"norm"`, the function internally uses the `thresTH3()` function, which requires two intervals in which the two thresholds are expected to be found. These intervals are determined by the default values of `thresTH3()`. When all the distributions are `"norm"`, the function uses the `nlm()` function, which requires two starting values (passed through the argument `start`).

Value

An object of class `thres3`, which is a list with two components:

. `T`, which is a list of at least ten components:

- `thres1` first threshold estimate.
- `thres2` second threshold estimate.
- `prev` prevalences provided by the user.
- `costs` cost matrix provided by the user.
- `k1` vector containing the first sample values provided by the user.
- `k2` vector containing the second sample values provided by the user.
- `k3` vector containing the third sample values provided by the user.
- `dist1` distribution assumed for the first population.
- `dist2` distribution assumed for the second population.
- `dist3` distribution assumed for the third population.

When not all distributions are normal, `T` also contains:

- `pars1` a numeric vector containing the estimation of the parameters of `dist1`.
- `pars2` a numeric vector containing the estimation of the parameters of `dist2`.
- `pars3` a numeric vector containing the estimation of the parameters of `dist3`.

. `CI`, a list of six components when `ci.method = "param"`:

- `lower1` the lower limit of the confidence interval for the first threshold.
- `upper1` the upper limit of the confidence interval for the first threshold.
- `lower2` the lower limit of the confidence interval for the second threshold.
- `upper2` the upper limit of the confidence interval for the second threshold.
- `alpha` significance level provided by the user.
- `ci.method` method used for the confidence intervals calculation.

A list of eleven components when `ci.method = "boot"`:

- `low.norm1` the lower limit of the bootstrap confidence interval for the first threshold based on the normal distribution.
- `up.norm1` the upper limit of the bootstrap confidence interval for the first threshold based on the normal distribution.
- `low.norm2` the lower limit of the bootstrap confidence interval for the second threshold based on the normal distribution.
- `up.norm2` the upper limit of the bootstrap confidence interval for the second threshold based on the normal distribution.
- `low.perc1` the lower limit of the bootstrap confidence interval for the first threshold based on percentiles.
- `up.perc1` the upper limit of the bootstrap confidence interval for the first threshold based on percentiles.
- `low.perc2` the lower limit of the bootstrap confidence interval for the second threshold based on percentiles.

- `up.perc2` the upper limit of the bootstrap confidence interval for the second threshold based on percentiles.
- `alpha` significance level.
- `B` number of bootstrap resamples.
- `ci.method` method used for the confidence intervals calculation.

When `ci = FALSE`, `CI` is `NULL`.

Note

It is assumed that `k1` is the sample with lower values and `k3` is the one taking higher values. If that is not the case, `k1`, `k2` and `k3` (and the corresponding parameters) are re-ordered as needed.

References

Efron B, Tibshirani RJ. (1993). An introduction to the bootstrap, Chapman & Hall.

Skaltsa K, Jover L, Fuster D, Carrasco JL. (2012). Optimum threshold estimation based on cost function in a multistate diagnostic setting. Statistics in Medicine, 31:1098-1109.

See Also

`thresTH3`, `plot.thres3`, `lines.thres3`

Examples

```
set.seed(1234)
n <- 100
k1 <- rlnorm(n)
k2 <- rnorm(n, 3, 1)
k3 <- rnorm(n, 5, 1)
rho <- c(1/3, 1/3, 1/3)

# assuming trinormality
start <- c(mean(k1), mean(k3))
thres3(k1, k2, k3, rho, dist1 = "norm", dist2 = "norm", dist3 = "norm",
       start = start, ci.method = "param")

# not assuming trinormality
thres3(k1, k2, k3, rho, B = 1000, ci.method = "boot", dist1 = "lnorm",
       dist2 = "norm", dist3 = "norm")

# supress confidence intervals calculation
thres3(k1, k2, k3, rho, ci = FALSE, dist1 = "lnorm", dist2 = "norm",
       dist3 = "norm")
```

`ThresholdROC` *Optimum threshold estimation based on cost function in a two- and three- state settings*

Description

The ThresholdROC package provides point and interval estimations of the optimum threshold as well as graphical tools for continuous diagnostic tests (two- and three- state settings). The point estimation is based on the definition of a cost function which we opt to minimise. An analytical estimator is available for the binormal and trinormal model and the empirical one is used for all settings. The interval estimation is based on the Delta method variance estimator in a binormal parametric setting and on methods on non-linear equations for the trinormal setting. Bootstrap methods are also provided for the confidence intervals.

Details

The most important functions are `thres2` and `thres3`. They offer a wide range of options for threshold estimation and inference in two and three state settings. We also include the function `diagnostic`, which calculates common measures of accuracy for binary diagnostic tests involving 2x2 contingency tables of classification results.

Author(s)

Sara Perez-Jaume, Natalia Pallares, Konstantina Skaltsa

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

References

Efron B, Tibshirani RJ. (1993). An introduction to the bootstrap, Chapman & Hall.

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

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

Skaltsa K, Jover L, Fuster D, Carrasco JL. (2012). Optimum threshold estimation based on cost function in a multistate diagnostic setting. Statistics in Medicine, 31:1098-1109.

Zhou XH, Obuchowski NA and McClish DK. (2002). Statistical methods in diagnostic medicine. John Wiley and sons.

`thresTH2` *Population-based threshold calculation (two-state setting)*

Description

This function estimates the theoretical optimum threshold for the specific distribution parameters, decision costs and disease prevalence in a two-state setting.

Usage

```
thresTH2(dist1, dist2, par1.1, par1.2, par2.1, par2.2, rho,
         costs = matrix(c(0, 0, 1, (1 - rho)/rho), 2, 2, byrow = TRUE),
         R = NULL, q1 = 0.05, q2 = 0.95, tol = 10^(-8))
```

Arguments

`dist1`	distribution to be assumed for the healthy population. See **Details**.
`dist2`	distribution to be assumed for the diseased population. See **Details**.
`par1.1`	first parameter of the distribution chosen for the healthy population.
`par1.2`	second parameter of the distribution chosen for the healthy population.
`par2.1`	first parameter of the distribution chosen for the diseased population.
`par2.2`	second parameter of the distribution chosen for the diseased population.
`rho`	disease prevalence.
`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 are a combination of costs that yields R=1, which is equivalent to the Youden index method (for details about this concept, see **References**). It must be set to `NULL` if the user prefers to set `R` (see next argument).
`R`	if the cost matrix `costs` is not set, 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`).
`q1`	probability of the left distribution in order to determine a low quantile. Default, 0.05.
`q2`	probability of the right distribution in order to determine a high quantile. Default, 0.95.
`tol`	tolerance to be used in function `uniroot`. Default, 10^(-8).

Details

Parameters `dist1` and `dist2` can be chosen between the following 2-parameter distributions: `"beta"`, `"cauchy"`, `"chisq"` (chi-squared), `"gamma"`, `"lnorm"` (lognormal), `"logis"` (logistic), `"norm"` (normal) and `"weibull"`.

Value

An object of class `thresTH2`, which is a list with five components:

`thres`	threshold estimate.
`prev`	disease prevalence provided by the user.
`costs`	cost matrix provided by the user.
`R`	R term, the product of the non-disease odds and the cost ratio (for further details about this concept, see **References**).
`method`	method used in the estimation. For an object of class `thresTH2` it is always equal to `"theoretical"` (meaning that the population-based method has been used).

Note

It is assumed that `dist1` is the distribution with lower values. If not, `dist1` and `dist2` (and the corresponding parameters) are exchanged.

References

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

Examples

```
# example 1
dist1 <- "norm"
dist2 <- "norm"
par1.1 <- 0
par1.2 <- 1
par2.1 <- 2
par2.2 <- 1
rho <- 0.1

thresTH2(dist1, dist2, par1.1, par1.2, par2.1, par2.2, rho)


# example 2
dist1 <- "norm"
dist2 <- "lnorm"
par1.1 <- 0
par1.2 <- 1
par2.1 <- 1
par2.2 <- 0.5
rho <- 0.3

thresTH2(dist1, dist2, par1.1, par1.2, par2.1, par2.2, rho)
```

---

thresTH3 *Population-based threshold calculation (three-state setting)*

Description

This function estimates the theoretical optimum thresholds for the specific distribution parameters, decision costs and prevalences in a three-state setting.

Usage

```
thresTH3(dist1, dist2, dist3, par1.1, par1.2,
         par2.1, par2.2, par3.1, par3.2, rho,
         costs = matrix(c(0, 1, 1, rho[1]/rho[2], 0, rho[3]/rho[2], 1, 1, 0),
                        3, 3, byrow = TRUE), q1=0.05, q2=0.5, q3=0.95,
         tol = 10^(-8))
```

Arguments

`dist1`	distribution to be assumed for the first population. See **Details**.
`dist2`	distribution to be assumed for the second population. See **Details**.
`dist3`	distribution to be assumed for the third population. See **Details**.
`par1.1`	first parameter of the first distribution.
`par1.2`	second parameter of the first distribution.
`par2.1`	first parameter of the second distribution.
`par2.2`	second parameter of the second distribution.
`par3.1`	first parameter of the third distribution.
`par3.2`	second parameter of the third distribution.
`rho`	3-dimensional vector of prevalences.
`costs`	cost matrix. Costs should be entered as a 3x3 matrix, where the first row corresponds to the costs associated with the classification of subjects in state 1 (C11, C12 and C13), second row corresponds to the costs associated with the classification of subjects in state 2 (C21, C22 and C23) and the third row corresponds to the costs associated with classification of subjects in state 3 (C31, C32, C33), where Cij is the cost of classifying an individual of class i as class j. Default cost values are a combination of costs that leads to the same thresholds as the Youden index method (see **References** for details).
`q1`	probability of the distribution taking lower values in order to determine a low quantile. Default, 0.05. See **Details**.
`q2`	probability of the middle distribution in order to determine a medium quantile. Default, 0.5. See **Details**.
`q3`	probability of the the distribution taking higher values in order to determine a high quantile. Default, 0.95. See **Details**.
`tol`	tolerance to be used in function `uniroot`. Default, 10^(-8).

Details

Parameters `dist1`, `dist2` and `dist3` can be chosen between the following 2-parameter distributions: `"beta"`, `"cauchy"`, `"chisq"` (chi-squared), `"gamma"`, `"lnorm"` (lognormal), `"logis"` (logistic), `"norm"` (normal) and `"weibull"`.

Parameters `q1`, `q2` and `q3` are used to determine two intervals where the `uniroot` function should look for the two threshold estimates. Thus, the first threshold is expected to be found between `quantile-1(q1)` and `quantile-2(q2)` and the second one, between `quantile-2(q2)` and `quantile-3(q3)`, being `quantile-i()` the quantile function for the i-th distribution, i=1,2,3.

Value

An object of class `thresTH3`, which is a list with five components:

`thres1`	first threshold estimate.
`thres2`	second threshold estimate.
`prev`	prevalences provided by the user.
`costs`	cost matrix provided by the user.
`method`	method used in the estimation. For an object of class `thresTH3` it is always equal to `"theoretical"` (meaning that the population-based method has been used).

Note

It is assumed that `dist1` is the distribution with lower values and `dist3` is the one taking higher values. If that is not the case, `dist1`, `dist2` and `dist3` (and the corresponding parameters) are re-ordered as needed.

References

Skaltsa K, Jover L, Fuster D, Carrasco JL. (2012). Optimum threshold estimation based on cost function in a multistate diagnostic setting. Statistics in Medicine, 31:1098-1109.

Examples

```
# example 1
dist <- "norm"
par1.1 <- 0
par1.2 <- 1
par2.1 <- 2
par2.2 <- 1
par3.1 <- 4
par3.2 <- 1
rho <- c(1/3, 1/3, 1/3)

thresTH3(dist, dist, dist,
  par2.1, par2.2, par1.1, par1.2,
  par3.1, par3.2, rho)
```

```
# example 2
dist1 <- "norm"
dist2 <- "lnorm"
dist3 <- "lnorm"
par1.1 <- 0
par1.2 <- 1
par2.1 <- 1
par2.2 <- 0.5
par3.1 <- 2
par3.2 <- 0.5
rho <- rep(1/3, 3)

thresTH3(dist1, dist2, dist3, par1.1, par1.2, par2.1, par2.2, par3.1, par3.2, rho)
```

Index