Package 'emplikDTA'

November 23, 2026

Type Package

Title Empirical Likelihood Inference for Diagnostic Test Accuracy

Version 0.1.0

Description Estimates diagnostic accuracy for continuous markers in two or three independent ordered groups. Provides empirical receiver operating characteristic (ROC) curves and surfaces, empirical likelihood inference for the area under the curve and the volume under the surface, and kernel-based selection of diagnostic thresholds. Three-class empirical likelihood methods are described in To, Adimari, and Chiogna (2024) <doi:10.1177/09622802241238998>.

License GPL (>= 2)

Depends R (>= 3.5.0), stats, utils, graphics, rgl, misc3d, ggplot2

Imports Rcpp (>= 1.1.1), rlang (>= 1.0.0)

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

LazyData true

LazyLoad yes

ByteCompile yes

NeedsCompilation yes

URL https://github.com/toduckhanh/emplikDTA

BugReports https://github.com/toduckhanh/emplikDTA/issues

Config/roxygen2/version 8.1.1

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Author Duc-Khanh To [aut, cre] (ORCID:
<https://orcid.org/0000-0002-4641-0764>),
Gianfranco Adimari [ctb],
Monica Chiogna [ctb]

Maintainer Duc-Khanh To <tdkhanh@hcmus.edu.vn>

Repository CRAN

Date/Publication 2026-11-23 14:30:02 UTC

Contents

emplikDTA-package . . . 2
ADNeuro . . . 3
antib2 . . . 4
auc . . . 5
CR_tcfs . . . 8
empi_llike_proj . . . 9
geom_confidence_region . . . 10
plot.threshold2 . . . 11
print.threshold2_all . . . 13
ROC_emp_curve . . . 14
ROC_emp_surface . . . 16
tcf2 . . . 18
threshold2 . . . 21
threshold3 . . . 25
vus . . . 28

Index **32**

`emplikDTA-package` *Empirical likelihood inference for diagnostic test accuracy*

Description

Estimation and empirical likelihood inference for continuous markers in two or three independent ordered diagnostic groups.

Details

Use `auc` and `ROC_emp_curve` for two groups; `vus`, `tcf2` and `ROC_emp_surface` for three groups. `threshold2` and `threshold3` select thresholds using kernel CDFs. Formula interfaces use marker ~ group. Larger marker values indicate higher diagnostic classes. Set diag_levels explicitly when the factor order is not the intended diagnostic order. These interfaces do not accept cluster identifiers or covariate models.

Author(s)

Maintainer: Duc-Khanh To <tdkhanh@hcmus.edu.vn> (ORCID)

Authors:

- Duc-Khanh To <tdkhanh@hcmus.edu.vn> (ORCID)

Other contributors:

- Gianfranco Adimari [contributor]
- Monica Chiogna [contributor]

See Also

Useful links:

- https://github.com/toduckhanh/emplikDTA
- Report bugs at https://github.com/toduckhanh/emplikDTA/issues

---

`ADNeuro` *Alzheimer's Disease Neuropsychometric Marker Dataset*

Description

A subset of the longitudinal cohort from the Washington University (WU) Alzheimer's Disease Research Center (ADRC). The dataset contains measurements of 14 neuropsychological markers collected from 118 independent individuals aged 75 years. Individuals were classified into three diagnostic groups according to the published Clinical Dementia Rating (CDR).

Format

A data frame with 118 observations on 15 variables:

`group` A factor with levels D^-, D^0, and D^+. The D^- group has CDR = 0 (no dementia), the D^0 group has CDR = 0.5 (very mild dementia), and the D^+ group has CDR = 1 (Alzheimer's disease).

`FACTOR1` Global factor score.

`ktemp` Temporal factor score.

`kpar` Parietal factor score.

`kfront` Frontal factor score.

`zpsy004` Logical memory score.

`zpsy005` Digit span forward score.

`zpsy006` Digit span backward score.

`zinfo` Information score.

`zbentc` Visual retention (10 seconds) score.

`zbentd` Visual retention (copy) score.

`zboston` Boston Naming Test score.

`zmentcon` Mental control score.

`zworflu` Word fluency score.

`zassc` Associate learning score.

Details

See the references for detailed descriptions of the 14 neuropsychological markers and their scoring procedures.

Source

The dataset is reproduced from the **DiagTest3Grp** package (version 1.6).

Originally published in:

Xiong, C., van Belle, G., Miller, J. P., & Morris, J. C. (2006). Measuring and Estimating Diagnostic Accuracy When There Are Three Ordinal Diagnostic Groups. *Statistics in Medicine*, **25**(7), 1251-1273.

Sorandt, M., & Hill, R. D. (1989). Very Mild Senile Dementia of the Alzheimer Type. II. Psychometric Test Performance. *Archives of Neurology*, **46**, 383-386.

Examples

```
data(ADNeuro)
str(ADNeuro)
summary(ADNeuro)
```

---

antib2 *COVID-19 Antibody Titers After the Second BNT162b2 Dose*

Description

Serum IgG titers against the receptor-binding domain (RBD) of SARS-CoV-2 in 100 healthcare workers. Samples were collected approximately two weeks after the second dose of the Pfizer–BioNTech BNT162b2 mRNA vaccine.

Usage

```
data(antib2)
```

Format

A data frame named `data_antib2` with 100 rows and two columns.

`antib2` Numeric RBD-specific IgG titer in arbitrary units per milliliter (AU/mL).

`status` Numeric indicator of prior SARS-CoV-2 infection. `0` denotes infection-negative (58 workers) and `1` denotes infection-positive (42 workers).

Source

COVID-19 antibody data from the Supporting Information of Bantis et al. (2024).

References

Bantis, L. E., Brewer, B., Nakas, C. T., and Reiser, B. (2024). Statistical Inference for Box–Cox based Receiver Operating Characteristic Curves. *Statistics in Medicine*, 43(30), 6099–6122. doi:10.1002/sim.10252.

Examples

```
data(antib2)
summary(data_antib2$antib2)
table(data_antib2$status)
```

---

auc *Area Under ROC curve (AUC) estimation and Empirical Likelihood Inference for AUC*

Description

This function estimates the AUC of a continuous diagnostic test (biomarker) in two-class setting, together with a confidence interval.

Usage

```
auc(x, ...)

## Default S3 method:
auc(
  x,
  y,
  auc0 = 1/2,
  ci_level = 0.95,
  B = 500,
  seed,
  plot = FALSE,
  alternative = c("greater", "less", "two.sided"),
  ...
)

## S3 method for class 'formula'
auc(formula, data, diag_levels = NULL, subset, na.action, ...)

## S3 method for class 'auc'
print(x, ...)

## S3 method for class 'auc'
summary(object, ...)
```

Arguments

x, y	numeric vectors (default method) contains values of diagnostic test (biomarker) corresponding to the first and second class of disease. The ordering is intended to be "increasing" with respect to the disease severity.
...	for formula method: additional arguments to be passed to the auc such as auc0, ci_level, B, plot, and alternative.
auc0	the null-hypothesis value of the AUC, defaulting to 1/2.
ci_level	a confidence level to be used for constructing the confidence interval; default is 0.95.
B	the number of bootstrap replicates (default: 500).
seed	Integer passed to set.seed; defaults to 34 when omitted. This changes the global random-number generator state.
plot	a logical indicating whether you want to plot empirical likelihood ratio and confidence interval for AUC.
alternative	The alternative hypothesis. Use "greater" (default) to test whether AUC is above auc0, "less" to test whether it is below, or "two.sided" to test whether it differs.
formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	a data frame containing the variables in the formula.
diag_levels	a vector (of strings) containing the ordered name chosen for the disease classes. The ordering is intended to be "increasing" with respect to the disease severity. If diag_levels = NULL (default), the elements of the vector follow the factor levels of the grouping variable (or the levels produced by coercing it to a factor). Groups are not reordered by their sample means.
subset	an optional expression indicating the subset of the rows of data that should be used in the fit. This can be a logical vector, or a numeric vector indicating which observation numbers are to be included, or a character vector of the row names to be included. All observations are included by default.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. The 'factory-fresh' default is na.omit. Another possible value is NULL, no action. Value na.exclude can be useful.
object	An object of class auc to summarise.

Details

Supply independent samples ordered by increasing marker values, with missing values handled before calling the default method. The formula interface uses marker ~ group with exactly two groups. The empirical estimate accounts for ties. Bootstrap resampling is performed independently within groups. The likelihood statistic is scaled by qchisq(0.5, 1) / median(r.bts), and the confidence limits are found by inversion using a chi-squared cutoff with one degree of freedom. If the empirical estimate is zero or one, the function returns early without inference; an estimate of one is adjusted to 1 / (1 + 0.5 / n1 / n2). An unbracketed confidence limit is returned as NA.

Value

auc returns an object of class "auc" which is a list containing the following components:

call	the matched call.
estimate	the estimated AUC.
n	Named vector of group sizes, n1 and n2.
alternative	the alternative hypothesis used for the test.

For interior estimates, the list also contains:

conf.int	a two-sided confidence interval for the AUC.
p.value	the p-value for testing AUC equal to auc0 against the specified alternative.
ll.value	Unadjusted empirical likelihood statistic at auc0.
r.bts	Bootstrap empirical likelihood statistics.
r.adj	the value of bootstrap adjustment factor.
ci_level	confidence level of the interval.

Inference components are absent for boundary estimates. The formula method additionally returns:

formula	The supplied formula.

Group-label metadata are currently not retained. The print method displays the results and returns the object invisibly.

P-value calculation

Let θ be the AUC, $\hat{\theta}$ its empirical estimate, and θ_0 the null value (auc0). For total sample size n, the empirical likelihood (EL) statistic at the null is returned as ll.value and equals

$$D_0 = 2n \left[\hat{\theta} \log \frac{\hat{\theta}}{\theta_0} + (1 - \hat{\theta}) \log \frac{1 - \hat{\theta}}{1 - \theta_0} \right].$$

The test uses the signed root of the bootstrap-adjusted EL statistic

$$R = \operatorname{sign}(\hat{\theta} - \theta_0) \sqrt{r D_0},$$

where r is the bootstrap scale factor returned as r.adj. The function sets $R = 0$ at equality and truncates negative values of rD_0 caused by rounding to zero before taking the square root.

Under $H_0 : \theta = \theta_0$, the test uses the standard normal approximation for the signed root (DiCiccio and Romano, 1989). With Φ denoting the standard normal cumulative distribution function, p.value is computed as follows.

alternative = "greater" $1 - \Phi(R)$ for $\theta > \theta_0$.

alternative = "less" $\Phi(R)$ for $\theta < \theta_0$.

alternative = "two.sided" $2\Phi(-|R|)$ for $\theta \neq \theta_0$. This equals the upper tail probability of a chi-squared distribution with one degree of freedom evaluated at R^2.

References

DiCiccio, T. J., & Romano, J. P. (1989). On adjustments based on the signed root of the empirical likelihood ratio statistic. *Biometrika*, **76**(3), 447–456. doi:10.1093/biomet/76.3.447.

To, D. K., Adimari, G., & Chiogna, M. (2024). Interval estimation in three-class receiver operating characteristic analysis: A fairly general approach based on the empirical likelihood. *Statistical Methods in Medical Research*, **33**, 5, 875-893.

Examples

```
set.seed(123)
x <- rnorm(60)
y <- rnorm(60, 1)
fit <- auc(x, y, B = 100, seed = 34)
summary(fit)
```

`CR_tcfs` *Draw a joint confidence region for three true class fractions*

Description

Add a fixed-threshold empirical likelihood confidence region and empirical operating point to an existing ROC surface scene.

Usage

```
CR_tcfs(x, ...)

## S3 method for class 'ROC_emp_surface'
CR_tcfs(
  x,
  cpts = NULL,
  ci_level = 0.95,
  color1 = "red",
  color2 = "blue",
  smooth = 0,
  alpha = 0.5,
  fill = FALSE,
  ...
)
```

Arguments

`x`	An object of class `ROC_emp_surface`.
`...`	Arguments forwarded to the method.
`cpts`	Numeric pair of ordered thresholds; must be supplied.
`ci_level`	Confidence level in (0, 1).

color1	Colour of the empirical point.
color2	Colour requested for the confidence boundary.
smooth	Smoothing argument passed to misc3d::contour3d.
alpha	Opacity of the confidence region.
fill	Whether to fill the contour surface.

Details

The region uses a chi-squared cutoff with three degrees of freedom on a 51-by-51-by-51 grid. Axes are TCF1, TCF3, TCF2.

Value

The input object, invisibly, if plotting succeeds.

See Also

ROC_emp_surface

---

empi_llike_proj *Projected two-sample empirical likelihood statistic*

Description

Low-level calculation of a two-sample minus twice log empirical likelihood ratio at a fixed threshold.

Usage

```
empi_llike_proj(
  X1,
  X2,
  n1,
  n2,
  tcf1,
  tcf2,
  tau,
  type_F = c("empi", "Adi", "Adi_ties")
)
```

Arguments

X1, X2	Numeric observations from two independent ordered groups.
n1, n2	Sample sizes associated with X1 and X2.
tcf1, tcf2	Proposed probabilities P(X1 <= tau) and P(X2 > tau). Use interior probabilities for finite calculations.

tau	Single threshold in the overlap of the observed group ranges.
type_F	CDF estimator: "empi" uses empirical proportions; "Adi_ties" uses the interpolated tie-aware CDF. "Adi" requires an Fs helper that is not currently defined in the package and is unsupported.

Details

This helper does not validate sample sizes or handle zero log terms by limits. Boundary probabilities can produce NaN. Prefer the high-level inference functions for ordinary analyses.

Value

Numeric likelihood statistic, Inf outside overlapping ranges, or NaN for undefined boundary arithmetic.

Examples

```
empi_llike_proj(1:5, 2:6, 5, 5, tcf1 = 0.5, tcf2 = 0.5, tau = 3)
```

---

geom_confidence_region

Add a Confidence Region to an Empirical ROC Curve

Description

Add a Confidence Region to an Empirical ROC Curve

Functions for adding graphical layers to an empirical ROC curve produced by `plot.ROC_emp_curve`. These functions return standard ggplot2 layers and are intended to be combined with the output of 'plot()' using the '+' operator.

`geom_confidence_region` adds the confidence region for the pair of specificity and sensitivity corresponding to a specified cut-point.

`geom_operating_point()` adds the empirical operating point (false positive rate, true positive rate) corresponding to a specified cut-point.

Usage

```
geom_confidence_region(
  object,
  cpt_type = c("fixed", "optimal"),
  cpt = NULL,
  ci_level = 0.95,
  colour = "blue",
  linewidth = 0.8
)

geom_operating_point(
```

```
  object,
  cpt_type = c("fixed", "optimal"),
  cpt = NULL,
  colour = "red",
  size = 2.5
)
```

Arguments

object	A ROC_emp_curve object for fixed thresholds, or a threshold2 object fitted with ci = TRUE for optimal thresholds.
cpt_type	Use a fixed threshold or a fitted optimal threshold.
cpt	a numeric value specifying the cut-point at which the operating point and confidence region are computed.
ci_level	the confidence level of the confidence region.
colour	a character string specifying the colour of the graphical element.
linewidth	a numeric value specifying the line width of the confidence region boundary.
size	a numeric value specifying the point size.

Details

Both branches currently use a chi-squared cutoff with two degrees of freedom. For the bootstrap-calibrated optimal-threshold region, use `plot` on the threshold2 fit. The cpt argument is ignored for optimal thresholds.

Value

`geom_confidence_region()` returns a `ggplot2::geom_contour()` layer.

`geom_operating_point()` returns a `ggplot2::geom_point()` layer.

See Also

`ROC_emp_curve` and `plot.ROC_emp_curve`

---

plot.threshold2 *Plot sensitivity and specificity at the estimated optimal threshold*

Description

Plot a point estimate and, when available, the boundary of the joint empirical likelihood (EL) confidence region. Plotting uses the stored results and does not repeat estimation or bootstrap sampling.

Usage

```
## S3 method for class 'threshold2'
plot(
  x,
  ci = NULL,
  point = c("auto", "empirical", "kernel"),
  point_colour = "black",
  point_size = 1.5,
  ci_colour = "blue",
  ci_linewidth = 0.8,
  xlab = "Specificity",
  ylab = "Sensitivity",
  ...
)

## S3 method for class 'threshold2_all'
plot(
  x,
  method = c("all", "YI", "CtP", "MA"),
  ci = NULL,
  point = c("auto", "empirical", "kernel"),
  point_colour = "black",
  point_size = 1.5,
  ci_colour = "blue",
  ci_linewidth = 0.8,
  xlab = "Specificity",
  ylab = "Sensitivity",
  ...
)
```

Arguments

x	An object of class threshold2 or, for the all-methods plot method, threshold2_all.
ci	Logical; whether to draw the EL confidence boundary. If NULL, draw it when the object contains the required EL results.
point	Which point estimate to display. "auto" uses the empirical estimates when available, as in the original plot, and otherwise uses the kernel estimates. "empirical" requires x$spe_sen_emp; "kernel" uses x$spe_opt and x$sen_opt.
point_colour	Colour of the point estimate.
point_size	Size of the point estimate.
ci_colour	Colour of the EL confidence boundary.
ci_linewidth	Line width of the EL confidence boundary.
xlab, ylab	Axis labels.
...	Reserved for additional arguments.
method	For the all-methods plot, "all" (default) displays three panels; "YI", "CtP", or "MA" displays only the selected method.

Details

The contour at `x$c_alp` is the boundary of a joint confidence region for specificity and sensitivity, rather than two separate marginal confidence intervals. Changing `point` changes only the displayed point; the stored EL region remains the same.

Value

A `ggplot` object that can be printed or further customised.

`print.threshold2_all` *Print a comparison of the three threshold-selection methods*

Description

Print a comparison of the three threshold-selection methods

Usage

```
## S3 method for class 'threshold2_all'
print(x, ...)
```

Arguments

`x`	An object of class `threshold2_all`.
`...`	Additional arguments passed to `print.data.frame`.

Value

The input object, invisibly.

See Also

`threshold2`

ROC_emp_curve *Empirical ROC curve estimation and Empirical Likelihood Inference for ROC curve*

Description

This function estimates the ROC curve of a continuous diagnostic test (biomarker) in two-class setting.

Usage

```
ROC_emp_curve(x, ...)

## Default S3 method:
ROC_emp_curve(x, y, ncp = 150, main, color = "steelblue", plot = TRUE, ...)

## S3 method for class 'formula'
ROC_emp_curve(formula, data, diag_levels = NULL, subset, na.action, ...)

## S3 method for class 'ROC_emp_curve'
plot(x, color = "steelblue", linewidth = 1, ...)
```

Arguments

x, y	numeric vectors (default method) contains values of diagnostic test (biomarker) corresponding to the first and second class of disease. The ordering is intended to be "increasing" with respect to the disease severity.
...	for formula method: additional arguments to be passed to the ROC_emp_curve i.e., ncp, main, color, and plot.
ncp	an integer specifying the number of cut-points used to construct the empirical ROC curve. Larger values produce a smoother approximation at the expense of additional computation.
main	a character string giving the title of the plot.
color	a character string specifying the color of the ROC curve.
plot	a logical indicating whether the empirical ROC curve should be plotted. If 'FALSE', the computed ROC curve object is returned without producing a graphical display.
formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	a data frame containing the variables in the formula.
diag_levels	a vector (of strings) containing the ordered name chosen for the disease classes. The ordering is intended to be "increasing" with respect to the disease severity. If diag_levels = NULL (default), the elements of the vector follow the factor levels of the grouping variable (or the levels produced by coercing it to a factor). Groups are not reordered by their sample means.

subset	an optional expression indicating the subset of the rows of data that should be used in the fit. This can be a logical vector, or a numeric vector indicating which observation numbers are to be included, or a character vector of the row names to be included. All observations are included by default.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. The 'factory-fresh' default is na.omit. Another possible value is NULL, no action. Value na.exclude can be useful.
linewidth	Line width for the curve plot.

Details

Specificity is $P(X \leq t)$ and sensitivity is $P(Y > t)$. The plot uses false positive rate (1 - specificity) on the horizontal axis. Use geom_confidence_region for a fixed-threshold EL region. The grid contains minus and plus infinity and ncp - 2 equally spaced points over the pooled range. The formula is marker ~ group. The default methods return the fitted object invisibly.

Value

ROC_emp_curve returns an object of class "ROC_emp_curve" which is a list containing at least the following components:

call	the matched call.
n, ncp	Group sizes and number of grid thresholds.
x, y	Original group samples.
vals	Matrix with columns Spe (specificity) and Sen (sensitivity).
cutpoints	Vector of thresholds.

Generic functions such as plot is also used to make a plot of ROC curve with ggplot2 style.

The plot method returns a ggplot object.

References

To, D. K., Adimari, G., & Chiogna, M. (2024). Interval estimation in three-class receiver operating characteristic analysis: A fairly general approach based on the empirical likelihood. *Statistical Methods in Medical Research*, **33**, 5, 875-893.

Examples

```
set.seed(123)
x <- rnorm(40, 0, 1)
y <- rnorm(40, 1, 1)
fit <- ROC_emp_curve(x, y, ncp = 30, plot = FALSE)
head(fit$vals)
plot(fit) + geom_operating_point(fit, cpt = 0.5)
```

ROC_emp_surface *Empirical ROC surface estimation*

Description

This function estimates the ROC surface of a continuous diagnostic test (biomarker) in three-class setting.

Usage

```
ROC_emp_surface(x, ...)

## Default S3 method:
ROC_emp_surface(
  x,
  y,
  z,
  ncp = 150,
  main,
  color = "steelblue",
  alpha = 0.5,
  plot = TRUE,
  ...
)

## S3 method for class 'formula'
ROC_emp_surface(formula, data, diag_levels = NULL, subset, na.action, ...)

## S3 method for class 'ROC_emp_surface'
plot(
  x,
  main = "Empirical ROC surface",
  color = "steelblue",
  alpha = 0.5,
  new_window = TRUE,
  add = FALSE,
  ...
)

## S3 method for class 'ROC_emp_surface'
print(x, ...)
```

Arguments

x, y, z	numeric vectors (default method) contains values of diagnostic tests (biomarkers) corresponding to the first, second and third class of disease. The ordering is intended to be "increasing" with respect to the disease severity.

...	for formula method: additional arguments to be passed to the ROC_emp_surface i.e., ncp, main, color, alpha and plot.
ncp	an integer specifying the number of cut-points used to construct the empirical ROC surface. Larger values produce a smoother approximation at the expense of additional computation.
main	Requested plot title; currently accepted but not drawn.
color	a character string specifying the color of the ROC surface.
alpha	vector of alpha values between 0.0 (fully transparent) and 1.0 (opaque).
plot	a logical indicating whether the empirical ROC surface should be plotted. If 'FALSE', the computed ROC surface object is returned without producing a graphical display.
formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	a data frame containing the variables in the formula.
diag_levels	a vector (of strings) containing the ordered name chosen for the disease classes. The ordering is intended to be "increasing" with respect to the disease severity. If diag_levels = NULL (default), the elements of the vector follow the factor levels of the grouping variable (or the levels produced by coercing it to a factor). Groups are not reordered by their sample means.
subset	an optional expression indicating the subset of the rows of data that should be used in the fit. This can be a logical vector, or a numeric vector indicating which observation numbers are to be included, or a character vector of the row names to be included. All observations are included by default.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. The 'factory-fresh' default is na.omit. Another possible value is NULL, no action. Value na.exclude can be useful.
new_window	Open a new rgl window when not adding to a scene.
add	Add the surface to the current scene.

Details

For ordered thresholds t_1 and t_2, $TCF_1 = P(X <= t_1)$, $TCF_2 = P(t_1 < Y \leq t_2)$, and $TCF_3 = P(Z > t_2)$. The plot axes are TCF_1, TCF_3 and TCF_2. Plotting uses an rgl device. The grid contains minus and plus infinity and ncp - 2 equally spaced points over the pooled range. The formula is marker ~ group. The default methods return the fitted object invisibly.

Value

ROC_emp_surface returns an object of class "ROC_emp_surface" which is a list containing at least the following components:

call	the matched call.
n, ncp	Group sizes and number of grid thresholds.

x, y, z	Original group samples.
vals	Matrix with columns TCF1, TCF2 and TCF3.
cpoint	Matrix of threshold pairs.
cutpoints	Vector of grid thresholds.
tcf1, tcf2, tcf3	Matrices used to draw the surface.

Generic functions such as `plot` is also used to make a 3D plot of ROC surface with `rgl` style.

Plot and print methods return the input invisibly.

References

To, D. K., Adimari, G., & Chiogna, M. (2024). Interval estimation in three-class receiver operating characteristic analysis: A fairly general approach based on the empirical likelihood. *Statistical Methods in Medical Research*, **33**, 5, 875-893.

Examples

```
set.seed(123)
x <- rnorm(40, 0, 1)
y <- rnorm(40, 1, 1)
z <- rnorm(40, 2, 1)
fit <- ROC_emp_surface(x, y, z, ncp = 15, plot = FALSE)
head(fit$vals)
print(fit)
```

`tcf2` *Estimate the middle-class true class fraction*

Description

Estimate TCF2 at specified TCF1 and TCF3 using sample quantiles, with bootstrap-calibrated empirical likelihood inference.

Usage

```
tcf2(x, ...)

## Default S3 method:
tcf2(
  x,
  y,
  z,
  tcf10,
  tcf30,
  tcf20 = NULL,
  ci_level = 0.95,
  B = 500,
```

```
  seed,
  plot = FALSE,
  ...
)

## S3 method for class 'formula'
tcf2(
  formula,
  data,
  diag_levels = NULL,
  subset,
  na.action,
  tcf10,
  tcf30,
  tcf20 = NULL,
  ...
)

## S3 method for class 'tcf2'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'tcf2'
summary(object, ...)
```

Arguments

x, y, z	Numeric vectors for three independent groups, in increasing marker order. Supply finite observations with missing values already handled. For the generic, x may also be a formula.
...	Arguments forwarded to the default method; unused by print/summary.
tcf10, tcf30	Target first- and third-class fractions in (0, 1).
tcf20	Optional null-hypothesis value for TCF2; NULL skips the test.
ci_level	Confidence level in (0, 1).
B	Positive integer number of bootstrap replicates.
seed	Integer passed to set.seed; defaults to 34 when omitted. The global random-number generator state is changed.
plot	Whether to print the likelihood-ratio plot.
formula	Formula marker ~ group with exactly three groups.
data	Data frame containing the formula variables.
diag_levels	Three group labels in increasing disease order. If NULL, the factor levels are used; sample means do not reorder the groups.
subset	Optional rows selected through model.frame.
na.action	Missing-value action passed to model.frame.
digits	Retained by the print method; currently not used.
object	An object of class tcf2 to summarise.

Details

The lower threshold is `quantile(x, tcf10, type = 8)` and the upper threshold is `quantile(z, 1 - tcf30, type = 8)`. TCF2 is the empirical CDF difference in group two. Choose targets giving an ordered pair of thresholds; the default method does not check this condition. Bootstrap samples are augmented by each original group's range. Confidence limits invert the adjusted statistic using a chi-squared cutoff with one degree of freedom. An unbracketed limit is NA.

Value

`tcf2` returns an object of class "tcf2" which is a list containing the following components:

`call`	The matched call.
`estimate`	The estimated middle-class true class fraction.
`tau_est`	Vector of estimated lower and upper thresholds.
`tcf1`, `tcf3`	The supplied target first- and third-class fractions.
`n`	Named vector of group sizes, `n1`, `n2`, and `n3`.

For interior estimates, the list also contains:

`conf.int`	A two-sided confidence interval for TCF2.
`r.bts`	Bootstrap empirical likelihood statistics.
`r.adj`	The bootstrap adjustment factor.
`ci_level`	Confidence level of the interval.

When inference is computed and `tcf20` is supplied, the list also contains:

`p.value`	The p-value for testing TCF2 equal to `tcf20`.
`ll.value`	The adjusted empirical likelihood statistic at `tcf20` minus the confidence cutoff.

An estimate of zero returns immediately; one is replaced by `1 / (1 + 0.5 / length(y))`, also without inference. The formula method additionally returns:

`formula`	The supplied formula.

The `print` method returns the object invisibly. The `summary` method returns a list with components `estimate`, `conf.int`, `p.value`, and `n`, as described above.

Examples

```
set.seed(12)
x <- rnorm(60)
y <- rnorm(60, 1)
z <- rnorm(60, 2)
fit <- tcf2(x, y, z, tcf10 = 0.6, tcf30 = 0.6, B = 50)
summary(fit)
```

threshold2 *Optimal threshold selection and empirical likelihood inference*

Description

Estimate an optimal threshold for a continuous diagnostic marker in two independent groups using kernel-smoothed distribution functions. Optionally compute a bootstrap percentile confidence interval for the threshold and a bootstrap-calibrated joint empirical likelihood (EL) confidence region for specificity and sensitivity.

Usage

```
threshold2(x, ...)

## Default S3 method:
threshold2(
  x,
  y,
  method = c("YI", "CtP", "MA", "all"),
  kernel_type = c("gauss", "epan"),
  bwd = NULL,
  bwd_type = c("RT", "AZZ", "CV"),
  ci = TRUE,
  ci_level = 0.95,
  B = 500,
  seed,
  plot = FALSE,
  ...
)

## S3 method for class 'formula'
threshold2(formula, data, diag_levels = NULL, subset, na.action, ...)
```

Arguments

x	Numeric vector of marker measurements in the reference-negative group (class 0), or a formula supplied as the first argument to the generic. Larger marker values are assumed to indicate the reference-positive group.
...	Arguments passed from the generic or formula method to `threshold2.default`.
y	Numeric vector of marker measurements in the reference-positive group (class 1). Used by the default method.
method	Threshold-selection criterion: `"YI"` maximises the Youden index; `"CtP"` minimises the distance to perfect classification; `"MA"` maximises the product of sensitivity and specificity. The default is `"YI"`. `"all"` runs all three criteria using shared bandwidths and, when requested, shared bootstrap samples.

kernel_type	Kernel used for CDF estimation: "gauss" for the Gaussian kernel (default), or "epan" for the Epanechnikov kernel.
bwd	Numeric vector of two positive bandwidths, in the order c(bandwidth_for_x, bandwidth_for_y). If NULL, bandwidths are selected separately for the two groups using bwd_type.
bwd_type	Bandwidth-selection method when bwd = NULL. "RT" (default) and "AZZ" use the corresponding methods of bwd_plugin; "CV" uses bwd_cv. Ignored when bwd is supplied.
ci	Logical; whether to perform bootstrap and EL inference. Defaults to TRUE. If FALSE, only the threshold and kernel-based specificity and sensitivity estimates are computed.
ci_level	Nominal confidence level, strictly between 0 and 1. Defaults to 0.95. Used for both the percentile interval of the threshold and the bootstrap calibration of the joint EL region.
B	Positive integer giving the number of bootstrap replicates. Defaults to 500; used only when ci = TRUE.
seed	Random seed for the bootstrap. If omitted, 34 is used. Used only when ci = TRUE. The function calls set.seed and therefore changes the global random-number generator state.
plot	Logical; whether to print plot.threshold2 for the fitted object. Defaults to FALSE. Retained for convenience; estimation and plotting can also be performed separately using plot(result). For method = "all", the default plot has one panel per criterion.
formula	A two-sided formula of the form marker ~ group, where marker is numeric and group identifies the two reference classes.
data	Data frame containing the variables used in formula.
diag_levels	Optional vector specifying the two reference-group labels, in reference-negative then reference-positive order. If NULL, group ordering is determined by the formula-data extraction helper. Ensure that this ordering agrees with the assumed marker direction.
subset	Optional expression specifying the subset of observations to use in the formula method; passed to the formula-data extraction helper.
na.action	Missing-value handling for the formula method; passed to the formula-data extraction helper. For the default method, supply finite observations with missing values already handled.

Details

Let F_0 and F_1 denote the distribution functions in the reference-negative and reference-positive groups. An observation is classified as positive when its marker value exceeds the threshold τ. Thus,

$$\mathrm{Sp}(\tau) = F_0(\tau), \qquad \mathrm{Se}(\tau) = 1 - F_1(\tau).$$

The two CDFs are estimated using cdf_kernel. The threshold is optimised over the range of the pooled observations. For "YI", a grid search locates candidate peaks and bounded one-dimensional

optimisation refines them. This search is also used in each bootstrap replicate. For `"CtP"` and `"MA"`, optimisation uses bounded `optim` with method `"L-BFGS-B"`.

The available criteria are:

`"YI"` Maximise $\mathrm{Sp}(\tau) + \mathrm{Se}(\tau) - 1 = F_0(\tau) - F_1(\tau)$.

`"CtP"` Minimise $\sqrt{[1 - \mathrm{Sp}(\tau)]^2 + [1 - \mathrm{Se}(\tau)]^2}$, the distance to $(\mathrm{Sp}, \mathrm{Se}) = (1, 1)$.

`"MA"` Maximise $\mathrm{Sp}(\tau)\mathrm{Se}(\tau)$ by maximising $\log \mathrm{Sp}(\tau) + \log \mathrm{Se}(\tau)$.

With `method = "all"`, bandwidth selection and bootstrap resampling are performed only once. The three criteria are fitted to the same original observations and the same bootstrap replicates. Each criterion retains its own threshold, threshold interval and bootstrap EL cutoff. EL grid coordinates are constructed once; grid statistics are reused only when two fitted thresholds are exactly identical. Shared resamples do not make the three confidence regions jointly calibrated across criteria: each region has nominal level `ci_level` separately.

When `ci = TRUE`, observations are resampled independently within each group. The original minimum and maximum of each group are appended to its resampled observations. The threshold is re-estimated in every replicate, keeping the original bandwidths and optimisation bounds fixed. The threshold interval uses bootstrap quantiles at $(1 - \mathrm{ci_level})/2$ and $(1 + \mathrm{ci_level})/2$, with `quantile(type = 8)`. This interval is a bootstrap percentile interval, not an EL interval.

The bootstrap EL statistic is evaluated at the original kernel estimates of specificity and sensitivity, using each replicate's estimated threshold and empirical CDFs. Its `ci_level` quantile, computed using the default `quantile` type, is stored as `c_alp`. For the original samples, the empirical-CDF EL statistic at the estimated threshold is evaluated on a 501×501 grid over $[0, 1]^2$. The contour $z = \mathrm{c_alp}$ gives the plotted joint confidence boundary. This is not a pair of separate marginal confidence intervals.

The default plot displays the empirical point stored in `spe_sen_emp` when available, matching the original plotting behaviour. The returned `spe_opt` and `sen_opt`, by contrast, are kernel-based estimates. Use `plot(result, point = "kernel")` to display those estimates. Changing the displayed point does not recompute the stored EL region. With `ci = FALSE`, the plot shows only the kernel-based point.

Value

For a single criterion, `threshold2` returns an object of class "threshold2" which is a list containing the following components:

`opt_thresh`	Estimated optimal threshold.
`spe_opt`	Kernel-based specificity at `opt_thresh`.
`sen_opt`	Kernel-based sensitivity at `opt_thresh`.
`n`	Named vector of original group sizes, `n1` and `n2`.
`call`	Matched function call.
`method`	The fitted criterion: `"YI"`, `"CtP"`, or `"MA"`.
`bwd`	Bandwidths used for the two groups.
`kernel_type`	Kernel used for CDF estimation.
`ci_level`	The requested confidence level when inference is computed, or `NULL` otherwise.

When `ci = TRUE`, the object additionally contains:

`out_bts`	A two-row bootstrap matrix: row `tau` contains estimated thresholds and row `ll` contains EL statistics. Columns correspond to bootstrap replicates.
`c_alp`	Bootstrap-calibrated cutoff for the joint EL region.
`ci_tau`	Lower and upper bootstrap percentile confidence limits for the threshold.
`grid_sen_spe`	Data frame with columns `Spe`, `Sen`, and `z`, containing grid coordinates and their EL statistics.
`spe_sen_emp`	Empirical specificity and sensitivity at the estimated threshold, with components `sp` and `se`.

The formula method additionally returns:

`formula`	The supplied formula.

Group-label metadata are currently absent because the wrapper reads a field name not returned by the formula-data helper.

For `method = "all"`, an object of class `"threshold2_all"` is returned. This is a list containing the following components:

`results`	Named list with elements `YI`, `CtP`, and `MA`, each a complete `threshold2` object as described above.
`summary`	Three-row data frame with columns `method`, `opt_thresh`, `spe_opt`, and `sen_opt`. Columns `ci_tau_lower` and `ci_tau_upper` are added when `ci = TRUE`.
`method`	The value `"all"`.
`n`, `bwd`, `kernel_type`, `call`, `ci_level`	Shared metadata as described above.
`formula`	The supplied formula, for the formula interface only.

For the formula interface, the formula and matched call are stored both in the parent object and in each fit. Printing the all-methods object displays its summary table.

Computational considerations

The two groups may have different sample sizes. Bootstrap samples are stored as pairs of vectors rather than as equal-length data-frame columns. EL grid evaluation may be computationally expensive, particularly for `method = "all"` when all three fitted thresholds differ. A non-finite bootstrap EL cutoff causes an error rather than a plot with an undefined confidence boundary.

See Also

`plot.threshold2`

Examples

```
set.seed(123)
negative <- rnorm(60, 0, 1)
positive <- rnorm(60, 1.5, 1)

# Point estimation without bootstrap or EL grid evaluation
fit <- threshold2(negative, positive, method = "YI",
                  bwd = c(0.4, 0.4), ci = FALSE)
fit$opt_thresh
c(Specificity = fit$spe_opt, Sensitivity = fit$sen_opt)

# All criteria, without inference
fits <- threshold2(negative, positive, method = "all",
                   bwd = c(0.4, 0.4), ci = FALSE)
fits$summary
fits$results$MA$opt_thresh

# Formula interface with explicit reference-group ordering
dat <- data.frame(
  marker = c(negative, positive),
  group = factor(rep(c("negative", "positive"), each = 60),
                 levels = c("negative", "positive"))
)
fit_formula <- threshold2(marker ~ group, data = dat,
                          diag_levels = c("negative", "positive"),
                          method = "CtP", bwd = c(0.4, 0.4), ci = FALSE)

## Not run:
# Full inference; the EL grid can take time to compute
fit_el <- threshold2(negative, positive, method = "YI",
                     bwd = c(0.4, 0.4), ci = TRUE,
                     ci_level = 0.95, B = 500, seed = 34)
fit_el$ci_tau
plot(fit_el)
plot(fit_el, point = "kernel", ci_colour = "red")

fits_el <- threshold2(negative, positive, method = "all",
                      bwd = c(0.4, 0.4), ci = TRUE, B = 500, seed = 34)
fits_el$summary
plot(fits_el)                # One panel per criterion
plot(fits_el, method = "YI")  # Only the YI result
plot(fits_el$results$CtP)     # A child fit behaves like any threshold2 object

## End(Not run)
```

`threshold3` *Selecting the optimal pair of thresholds and Empirical Likelihood Inference*

Description

Estimate two thresholds for three independent ordered groups using kernel CDFs, optionally with bootstrap inference.

Usage

```
threshold3(x, ...)

## Default S3 method:
threshold3(
  x,
  y,
  z,
  method = c("GYI", "CtP", "MV", "all"),
  kernel_type = c("gauss", "epan"),
  bwd = NULL,
  bwd_type = c("RT", "AZZ", "CV"),
  ci = TRUE,
  ci_level = 0.95,
  B = 500,
  seed,
  plot = FALSE,
  ...
)

## S3 method for class 'formula'
threshold3(formula, data, diag_levels = NULL, subset, na.action, ...)
```

Arguments

x, y, z	Numeric marker vectors for the three groups in increasing order. Use finite data with missing values already handled.
...	Arguments passed to the default method.
method	Criterion: "GYI" maximises the sum of TCFs; "CtP" minimises distance to (1,1,1); "MV" maximises their product. The default is "GYI". "all" runs all three criteria using shared bandwidths and, when requested, shared bootstrap samples.
kernel_type	Gaussian ("gauss") or Epanechnikov ("epan") kernel.
bwd	Three positive bandwidths in group order, or NULL to estimate them.
bwd_type	RT or AZZ plug-in bandwidth, or CV cross-validation.
ci	Whether to compute bootstrap inference.
ci_level	Confidence level in (0, 1).
B	Positive integer number of bootstrap replicates.
seed	Seed passed to set.seed; default 34. Changes the global RNG state.
plot	Add the joint EL region to an empirical ROC surface when ci = TRUE.

formula	Formula marker ~ group with exactly three groups.
data	Data frame containing formula variables.
diag_levels	Group labels in increasing order; NULL uses factor levels.
subset	Optional rows passed to model.frame.
na.action	Missing-value action passed to model.frame.

Details

At ordered thresholds a and b, the fractions are $F_1(a)$, $F_2(b) - F_2(a)$, and $1 - F_3(b)$. Optimisation uses L-BFGS-B with bounds based on adjacent groups. Bootstrap resampling appends the original group extrema, keeps bandwidths fixed and re-estimates both thresholds. Threshold intervals use quantile type 8; the EL cutoff uses the default quantile type. For method = "all", bandwidth selection and bootstrap resampling are performed once; each criterion then estimates its own thresholds, TCFs, confidence intervals and EL cutoff. Bootstrap samples are stored as lists, allowing unequal group sizes. The optimiser does not guarantee ordered returned thresholds; inspect opt_thresh before using the result.

Value

For a single criterion, threshold3 returns an object of class "threshold3" which is a list containing the following components:

opt_thresh	Vector of two estimated optimal thresholds.
tcf1_opt, tcf2_opt, tcf3_opt	Kernel-based estimates of the three true class fractions at opt_thresh.
n	Named vector of original group sizes, n1, n2, and n3.
call	The matched call.

When ci = TRUE, the object additionally contains:

out_bts	A three-row bootstrap matrix containing two estimated thresholds and an EL statistic per replicate. Columns correspond to bootstrap replicates.
c_alp	Bootstrap-calibrated cutoff for the joint EL region.
ci_tau	A two-by-two matrix with thresholds in rows and lower/upper bootstrap percentile confidence limits in columns.

For method = "all", an object of class "threshold3_all" is returned. This is a list containing the following components:

results	Named list of threshold3 objects in the order GYI, CtP, and MV, each as described above.
summary	Data frame with one row per criterion and columns method, tau1, tau2, tcf1_opt, tcf2_opt, and tcf3_opt. When ci = TRUE, it also contains ci_tau1_lower, ci_tau1_upper, ci_tau2_lower, and ci_tau2_upper.
method	The value "all".
n	Named vector of original group sizes.

bwd	Bandwidths used for the three groups.
kernel_type	Kernel used for CDF estimation.
call	The matched call.
ci_level	The requested confidence level when inference is computed, or NULL otherwise.

The formula method additionally returns the following components in the returned object and in each component of results, when present:

formula	The supplied formula.
group.levels	Group labels in increasing disease order.

Examples

```
set.seed(21)
x <- rnorm(40, 0, 1)
y <- rnorm(40, 1, 1)
z <- rnorm(40, 2, 1)
fit <- threshold3(x, y, z, method = "CtP", bwd = rep(0.4, 3), ci = FALSE)
fit$opt_thresh

fit_all <- threshold3(rnorm(40), rnorm(50, 1), rnorm(60, 2),
                      method = "all", bwd = rep(0.4, 3), ci = FALSE)
fit_all$summary
fit_all$results$GYI$opt_thresh
```

---

vus *Volume Under Surface (VUS) estimation and Empirical Likelihood Inference for VUS*

Description

This function estimates the VUS of a continuous diagnostic test (biomarker) in three-class setting, together with a confidence interval.

Usage

```
vus(x, ...)

## Default S3 method:
vus(
  x,
  y,
  z,
  vus0 = 1/6,
  ci_level = 0.95,
  B = 500,
```

```
  seed,
  plot = FALSE,
  alternative = c("greater", "less", "two.sided"),
  ...
)

## S3 method for class 'formula'
vus(formula, data, diag_levels = NULL, subset, na.action, ...)

## S3 method for class 'vus'
print(x, ...)
```

Arguments

x, y, z	numeric vectors (default method) contains values of diagnostic test (biomarker) corresponding to the first, second and third class of disease. The ordering is intended to be "increasing" with respect to the disease severity.
...	for formula method: additional arguments to be passed to the vus such as vus0, ci_level, B, plot, and alternative.
vus0	the null-hypothesis value of the VUS, defaulting to 1/6.
ci_level	a confidence level to be used for constructing the confidence interval; default is 0.95.
B	the number of bootstrap replicates (default: 500).
seed	Integer passed to set.seed; defaults to 34 when omitted. This changes the global random-number generator state.
plot	a logical indicating whether you want to plot empirical likelihood ratio and confidence interval for VUS.
alternative	The alternative hypothesis. Use "greater" (default) to test whether VUS is above vus0, "less" to test whether it is below, or "two.sided" to test whether it differs.
formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	a data frame containing the variables in the formula.
diag_levels	a vector (of strings) containing the ordered name chosen for the disease classes. The ordering is intended to be "increasing" with respect to the disease severity. If diag_levels = NULL (default), the elements of the vector follow the factor levels of the grouping variable (or the levels produced by coercing it to a factor). Groups are not reordered by their sample means.
subset	an optional expression indicating the subset of the rows of data that should be used in the fit. This can be a logical vector, or a numeric vector indicating which observation numbers are to be included, or a character vector of the row names to be included. All observations are included by default.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. The 'factory-fresh' default is na.omit. Another possible value is NULL, no action. Value na.exclude can be useful.

Details

Supply independent samples ordered by increasing marker values, with missing values handled before calling the default method. The formula interface uses `marker ~ group` with exactly three groups. The empirical estimate accounts for ties. Bootstrap resampling is performed independently within groups. The likelihood statistic is scaled by `qchisq(0.5, 1) / median(r.bts)`, and the confidence limits are found by inversion using a chi-squared cutoff with one degree of freedom. If the empirical estimate is zero or one, the function returns early without inference; an estimate of one is adjusted to `1 / (1 + 0.5 / n1 / n2 / n3)`. An unbracketed confidence limit is returned as `NA`.

Value

`vus` returns an object of class "vus" which is a list containing the following components:

`call`	the matched call.
`estimate`	the estimated VUS.
`n`	Named vector of group sizes, `n1`, `n2`, and `n3`.
`alternative`	the alternative hypothesis used for the test.

For interior estimates, the list also contains:

`conf.int`	a two-sided confidence interval for the VUS.
`p.value`	the p-value for testing VUS equal to `vus0` against the specified alternative.
`ll.value`	Unadjusted empirical likelihood statistic at `vus0`.
`r.bts`	Bootstrap empirical likelihood statistics.
`r.adj`	the value of bootstrap adjustment factor.
`ci_level`	confidence level of the interval.

Inference components are absent for boundary estimates. The formula method additionally returns:

`formula`	The supplied formula.

Group-label metadata are currently not retained. The `print` method displays the results and returns the object invisibly.

P-value calculation

Let θ be the VUS, $\hat{\theta}$ its empirical estimate, and θ_0 the null value (`vus0`). For total sample size n, the empirical likelihood (EL) statistic at the null is returned as `ll.value` and equals

$$D_0 = 2n \left[\hat{\theta} \log \frac{\hat{\theta}}{\theta_0} + (1 - \hat{\theta}) \log \frac{1 - \hat{\theta}}{1 - \theta_0} \right].$$

The test uses the signed root of the bootstrap-adjusted EL statistic

$$R = \operatorname{sign}(\hat{\theta} - \theta_0) \sqrt{r D_0},$$

where r is the bootstrap scale factor returned as `r.adj`. The function sets $R = 0$ at equality and truncates negative values of rD_0 caused by rounding to zero before taking the square root.

Under $H_0 : \theta = \theta_0$, the test uses the standard normal approximation for the signed root (DiCiccio and Romano, 1989). With Φ denoting the standard normal cumulative distribution function, `p.value` is computed as follows.

`alternative = "greater"` $1 - \Phi(R)$ for $\theta > \theta_0$.

`alternative = "less"` $\Phi(R)$ for $\theta < \theta_0$.

`alternative = "two.sided"` $2\Phi(-|R|)$ for $\theta \neq \theta_0$. This equals the upper tail probability of a chi-squared distribution with one degree of freedom evaluated at R^2.

References

DiCiccio, T. J., & Romano, J. P. (1989). On adjustments based on the signed root of the empirical likelihood ratio statistic. *Biometrika*, **76**(3), 447–456. doi:10.1093/biomet/76.3.447.

To, D. K., Adimari, G., & Chiogna, M. (2024). Interval estimation in three-class receiver operating characteristic analysis: A fairly general approach based on the empirical likelihood. *Statistical Methods in Medical Research*, **33**, 5, 875-893.

Examples

```
set.seed(123)
x <- rnorm(60, 0, 1)
y <- rnorm(60, 1, 1)
z <- rnorm(60, 2, 1)
fit <- vus(x, y, z, B = 100, seed = 34)
fit$conf.int
```

Index

∗ **datasets**
    ADNeuro, 3
    antib2, 4

ADNeuro, 3
antib2, 4
auc, *2*, 5, *6*

CR_tcfs, 8

data_antib2 *(*antib2*)*, 4

empi_llike_proj, 9
emplikDTA *(*emplikDTA-package*)*, 2
emplikDTA-package, 2

formula, *6*, *14*, *17*, *29*

geom_confidence_region, 10
geom_operating_point
        *(*geom_confidence_region*)*, 10

na.exclude, *6*, *15*, *17*, *29*
na.fail, *6*, *15*, *17*, *29*
na.omit, *6*, *15*, *17*, *29*

options, *6*, *15*, *17*, *29*

plot.ROC_emp_curve, *10, 11*
plot.ROC_emp_curve *(*ROC_emp_curve*)*, 14
plot.ROC_emp_surface *(*ROC_emp_surface*)*,
        16
plot.threshold2, 11, *24*
plot.threshold2_all *(*plot.threshold2*)*,
        11
print.auc *(*auc*)*, 5
print.ROC_emp_surface
        *(*ROC_emp_surface*)*, 16
print.tcf2 *(*tcf2*)*, 18
print.threshold2_all, 13
print.vus *(*vus*)*, 28

ROC_emp_curve, *2*, *11*, *14*, 14
ROC_emp_surface, *2*, *9*, 16, *17*

summary.auc *(*auc*)*, 5
summary.tcf2 *(*tcf2*)*, 18

tcf2, *2*, 18
threshold2, *2*, *13*, 21
threshold3, *2*, 25

vus, *2*, 28, *29*