Package: mada (via r-universe)

September 9, 2024

Type Package

Title Meta-Analysis of Diagnostic Accuracy

Version 0.5.11

Date 2022-07-11

Author Philipp Doebler with contributions from Bernardo Sousa-Pinto

Maintainer Philipp Doebler <`philipp.doebler@googlemail.com`>

Description Provides functions for diagnostic meta-analysis. Next to basic analysis and visualization the bivariate Model of Reitsma et al. (2005) that is equivalent to the HSROC of Rutter & Gatsonis (2001) can be fitted. A new approach based to diagnostic meta-analysis of Holling et al. (2012) is also available. Standard methods like summary, plot and so on are provided.

Depends R (>= 3.5.0), stats, mvtnorm, ellipse, mvmeta, metafor

Imports methods

License GPL-2

URL r-forge.r-project.org/projects/mada/

NeedsCompilation no

Date/Publication 2022-07-15 19:40:02 UTC

Repository https://doebler.r-universe.dev

RemoteUrl https://github.com/cran/mada

RemoteRef HEAD

RemoteSha 5faba4c4f61ddfde156fda9f9d9b23c4d20443fd

Contents

`mada-package` *Meta-Analysis of diagnostic accuracy studies mada*

Description

This package provides functions for diagnostic meta-analysis. Next to basic analysis and visualization the bivariate Model of Reitsma et al. (2005) that is equivalent to the HSROC of Rutter&Gatsonis (2001) can be fitted. A new approach based to diagnostic meta-analysis of Holling et al. (2012) is also available. Standard methods like summary, plot and so on are provided.

Details

Package:	mada
Type:	Package
Version:	0.5.8
Date:	2017-10-06
License:	GPL-2

The package provides tools for the meta-analysis of diagnostic accuracy data. For this the number true positives (TP), false negatives (FN), true negatives (TN) and false postives (FP) for each study must be known. The package can fit the bivariate model of Reitsma et al (2005), a bivariate random effects model. This model has been shown by Harbord et al. (2007) to be equivalent to the HSROC proposed by Rutter & Gatsonis (2001). We approach this model as a linear mixed effects model to avoid the complications of non-linear mixed effects model. The main function to fit such model is `reitsma` and standard methods are available for the output of this function.

Author(s)

Author and Maintainer: Philipp Doebler

References

Rutter, C., & Gatsonis, C. (2001). "A hierarchical regression approach to meta-analysis of diagnostic test accuracy evaluations." *Statistics in Medicine*, **20**, 2865–2884.

Reitsma, J., Glas, A., Rutjes, A., Scholten, R., Bossuyt, P., & Zwinderman, A. (2005). "Bivariate analysis of sensitivity and specificity produces informative summary measures in diagnostic reviews." *Journal of Clinical Epidemiology*, **58**, 982–990.

Harbord, R., Deeks, J., Egger, M., Whiting, P., & Sterne, J. (2007). "A unification of models for meta-analysis of diagnostic accuracy studies." *Biostatistics*, **8**, 239–251.

See Also

`reitsma`

`AUC` *Area under the curve (AUC)*

Description

Calculates the area under the curve given a function or a fitted model.

Usage

```
## Default S3 method:
AUC(x, fpr = 1:99/100, ...)
## S3 method for class 'phm'
AUC(x, level = 0.95, ...)
## S3 method for class 'reitsma'
AUC(x, fpr = 1:99/100, sroc.type = "ruttergatsonis", ...)
```

Arguments

`x`	a function with range and domain in ROC space (default method) or an object of class `phm` or `reitsma`.
`fpr`	numeric vector, points on which the (S)ROC curve is evaluated
`level`	numeric, confidence level for the calculations of confidence intervals.
`sroc.type`	character, which SROC curve should be used to calculate the AUC? Besides the default `ruttergatsonis` the option `naive` is available.
`...`	further arguments, currently not used.

Details

The area under the curve is calculated using the trapezoidal rule. The argument `fpr` is the grid on which the (S)ROC curve is evaluated. In many cases the default grid will contain points on which the SROC curve can only be calculated by extrapolation; however if only a subinterval is specified a *partial AUC* is calculated and the AUC value might differ substantially.

For `phm` objects the AUC and its confidence interval is calculated analytically, for `reitsma` objects a call to the default method is invoked.

Value

An object of the class `AUC` which is really a list with component `AUC` and an optional component `ci`, which is currently only available from the `AUC` method for `phm` ojects.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

Examples

```
data(AuditC)
AUC(phm(AuditC))
```

`CIrho` *Confidence intervals for Spearman's ρ.*

Description

Using Fisher's z-transformation (`atanh`) and the classic normal approximation confidence intervals for a vector of correlations is computed.

Usage

```
CIrho(rho, N, level = 0.95)
```

Arguments

`rho`	numeric vector, must be between -1 and 1.
`N`	integer vector, sample sizes.
`level`	numeric, confidence level.

Value

A matrix with first column `rho` and two further columns with the lower and upper bound.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

Examples

```
CIrho(c(0.34,0.19), c(22, 48), level = 0.80)
```

`cochran.Q` *Cochran's Q statistic*

Description

Given estimates from primary studies and the weights of the single studies calculate Cochran's Q as a measure of heterogeneity.

Usage

```
cochran.Q(x, weights)
```

Arguments

`x`	numeric, typically a vector of effect sizes like (log-)OR
`weights`	numeric, see Details

Details

In fixed effects settings the weights are often inverse proportional to the variances of the primary studies. Cochran's Q is known to have low power to detect heterogeneity.

Value

A named vector of length 3. First element is `Q` followed by the `p-value` and the degrees of freedom.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

`crosshair` *Crosshair plot*

Description

Produces a crosshair plot or adds such a plot to an existing plot.

Usage

```
## Default S3 method:
crosshair(x, correction = 0.5, level = 0.95, method = "wilson",
                      xlim = c(0,1), ylim = c(0,1), length = 0.1, pch = 1,
                      add = FALSE, suppress = TRUE, ...)
```

Arguments

`x`	a data frame with variables including `TP`, `FN`, `FP`, `TN`, alternatively a matrix with column names including these.
`correction`	numeric, continuity correction applied to zero cells.
`level`	numeric, confidence level for the calculations of confidence intervals.
`method`	character, method used to calculate the confidence intervals for sensitivities, specificities and false positive rates. One of `"wald"`, `"wilson"`, `"agresti-coull"`, `"jeffreys"`, `"modified wilson"`, `"modified jeffreys"`, `"clopper-pearson"`, `"arcsine"`, `"logit"`, `"witting"`
`xlim`	part of ROC space to be plotted
`ylim`	part of ROC space to be plotted
`length`	length of "whiskers" of the crosshair.
`pch`	Symbol used to plot point estimates. Use `pch = ""` to suppress plotting point estimates.
`add`	logical, should the plot be added to the current plot?
`suppress`	logical, should the warnings produced by the internal call to `madad` be suppressed? Defaults to `TRUE`, since only the diagnostic accuracies and their confidence intervals are used in subsequent calculations.
`...`	further arguments passed on to `plot`.

Details

Crosshair plots go back to Phillips et al. (2010). Note that for fits of the `reitsma` function a crosshair method is available to plot pooled estimate, see `reitsma-class`.

Value

Besides plotting, the function returns an invisible `NULL`.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

References

Phillips, B., Stewart, L.A., & Sutton, A.J. (2010). "'Cross hairs' plots for diagnostic meta-analysis." *Research Synthesis Methods*, **1**, 308–315.

See Also

`ROCellipse`, `reitsma-class`

Examples

```
data(AuditC)
crosshair(AuditC)
```

`forest` *Forest plot for univariate measures*

Description

Produce a forest plot. Includes graphical summary of results if applied to output of suitable model-fitting function. `forest` methods for `madad` and `madauni` objects are provided.

Usage

```
## S3 method for class 'madad'
forest(x, type = "sens", log = FALSE, ...)
## S3 method for class 'madauni'
forest(x, log = TRUE, ...)
forestmada(x, ci, plotci = TRUE, main = "Forest plot", xlab = NULL,
          digits = 2L,  snames = NULL, subset = NULL, pch = 15,
          cex = 1, cipoly = NULL, polycol = NA, ...)
```

Arguments

`x`	an object for which a `forest` method exists or (in the case of `foresmada`) a vector of point estimates.
`ci`	numeric matrix, each row corresponds to a confidence interval (the first column being the lower bound and the second the upper).
`plotci`	logical, should the effects sizes and their confidence intervals be added to the plot (as text)?
`main`	character, heading of plot.
`xlab`	label of x-axis.
`digits`	integer, number of digits for axis labels and confidence intervals.
`snames`	character vector, study names. If `NULL`, generic study names are generated.
`subset`	integer vector, allows to study only a subset of studies in the plot. One can also reorder the studies with the help of this argument.
`pch`	integer, plotting symbol, defaults to a small square. Also see `plot.default`.
`cex`	numeric, scaling parameter for study names and confidence intervals.
`cipoly`	logical vector, which confidence interval should be plotted as a polygon? Useful for summary estimates. If set to `NULL`, regular confidence intervals will be used.
`polycol`	color of the polygon(s), passed on to `polygon`. The default value of `NA` implies no color.
`type`	character, one of `sens`, `spec`, `negLR`, `posLR` or `DOR`.
`log`	logical, should the log-transformed values be plotted?
`...`	arguments to be passed on to `forestmada` and further on to other plotting functions

Details

Produces a forest plot to graphically assess heterogeneity. Note that `forestmada` is called internally, so that the `...` argument can be used to pass on arguments to this function; see the examples.

Value

Returns and invisible `NULL`.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

See Also

`madad`, `madauni`

Examples

```
data(AuditC)

## Forest plot of log DOR with random effects summary estimate
forest(madauni(AuditC))

## Forest plot of negative likelihood ratio (no log transformation)
## color of the polygon: light grey
## draw the individual estimate as filled circles
forest(madauni(AuditC, type = "negLR"),
       log = FALSE, polycol = "lightgrey", pch = 19)

## Paired forest plot of sensitivities and specificities
## Might look ugly if device region is too small
old.par <- par()
AuditC.d <- madad(AuditC)

plot.new()
par(fig = c(0, 0.5, 0, 1), new = TRUE)
forest(AuditC.d, type = "sens", xlab = "Sensitivity")
par(fig = c(0.5, 1, 0, 1),  new = TRUE)
forest(AuditC.d, type = "spec", xlab = "Specificity")

par(old.par)

## Including study names
## Using Letters as dummies
forest(AuditC.d, type = "spec",  xlab = "Specificity",
      snames = LETTERS[1:14])
```

`mada-data` *Diagnostic accuracy data*

Description

Six data frames with diagnostic accuracy data from binary test outcomes.

Usage

```
data("AuditC")
data("Dementia")
data("IAQ")
data("SAQ")
data("skin_tests")
data("smoking")
```

Format

Six data frames with frequencies of true positives, false negatives, false positives and true negatives. The data set `smoking` combines the `IAQ` and `SAQ` data sets and these are the only ones with variables in addition to the frequencies.

TP numeric. number of true positives

FN numeric. number of false negatives

FP numeric. number of false positives

TN numeric. number of true negatives

type factor. self-administered (SAQ) or interviewer-administered questionnaire (IAQ)

author factor. Author(s) of review and year

study_id numeric. ID variable for study

result_id integer. ID variable for (dependent) 2x2-tables from the same study

population factor. general (G) or student (S) population

Details

The `AuditC` data is from Kriston et al. (2008). The `Dementia` from Mitchell (2009) and the `SAQ` and `IAQ` data are subsets from the data in Patrick et al. (1994), while `smoking` is the complete data. The `skin_tests` data is part of the data from Sousa-Pinto et al. (2021) and concerns the accuracy of penicillin allergy skin tests.

Source

Kriston, L., H\"oelzel, L., Weiser, A., Berner, M., & Haerter, M. (2008).“ Meta-analysis: Are 3 Questions Enough to Detect Unhealthy Alcohol Use?” *Annals of Internal Medicine*, **149**, 879–888.

Mitchell, A. (2009). “A meta-analysis of the accuracy of the mini-mental state examination in the detection of dementia and mild cognitive impairment.” *Journal of Psychiatric Research*, **43**, 411–431.

Patrick, D., Cheadle, A., Thompson, D., Diehr, P., Koepsell, T., & Kinne, S. (1994). “The validity of self-reported smoking: a review and meta-analysis.” *American Journal of Public Health*, **84**, 1086–1093.

Sousa-Pinto, B., Tarrio, I., Blumenthal, K.G., Araujo, L., Azevedo, L.F., Delgado, L. & Fonseca, J.A. (2021). “Accuracy of penicillin allergy diagnostic tests: A systematic review and meta-analysis.” *Journal of Allergy and Clinical Immunology*, **147**, 296–308.

`madad` *Descriptive statistics for meta-analysis of diagnostic accuracy*

Description

Given the frequencies of true positives, false negative, false positives and true negatives from primary diagnostic studies `madad` calculates various summary statistics. Apart from sensitivities, specificities and false positive rates the function also calculates the diagnostic odds ratio (DOR) and the positve and negative likelihood ratios, together with their respective confidence intervals. Also two hypothesis tests are calculated: one testing the equality of the sensitivities and the same for the false positive rates.

Usage

```
madad(x = NULL, TP, FN, FP, TN, level = 0.95, correction = 0.5,
  correction.control = "all", method = "wilson", yates = TRUE,
  suppress = TRUE, ...)

## S3 method for class 'madad'
print(x, digits = 3, ...)
```

Arguments

`x`	any object that can be converted to a data frame with integer variables `TP`, `FN`, `FP` and `TN`, alternatively a matrix with column names including `TP`, `FN`, `FP` and `TN`.
`TP`	vector of integers, ingored if `X` is not `NULL`.
`FN`	vector of integers, ingored if `X` is not `NULL`.
`FP`	vector of integers, ingored if `X` is not `NULL`.
`TN`	vector of integers, ingored if `X` is not `NULL`.
`correction`	numeric, continuity correction applied to zero cells.
`correction.control`	character, if set to `"all"` (the default) the continuity correction is added to the whole data if only one cell in one study is zero. If set to `"single"` the correction is only applied to rows of the data which have a zero.

`level`	numeric, confidence level for the calculations of confidence intervals.
`method`	character, method used to calculate the confidence intervals for sensitivities, specificities and false positive rates. One of `"wald"`, `"wilson"`, `"agresti-coull"`, `"jeffreys"`, `"modified wilson"`, `"modified jeffreys"`, `"clopper-pearson"`, `"arcsine"`, `"logit"`, `"witting"`
`yates`	logical, should a Yates correction be used for testing the equality of sensitivities and specificities?
`digits`	integer, to what decimal place is the output to be rounded?
`suppress`	logical, suppress the warning that is generated by `prop.test` when Chi-square approximation may be incorrect.
`...`	further arguments to be passed on the other funtions (currently none).

Details

All calculations are performed using the continuity corrected cell counts, so if there are zero cells, the sensitivities and specificities not equal to 1. This can be avoided by setting `correction.control` to `"none"`.

The test for the equality of sensitivities and its counterpart for the specificities is based on `prop.test`. This function will occasionally output warnings.

Value

An object of class `madad` which is essentially a list with the following components:

`sens`	A list of two components, `sens` (the sensitivities) and `sens.ci` the confidence intervals (a matrix with 2 columns).
`spec`	A list of two components, `spec` (the specificities) and `spec.ci` the confidence intervals (a matrix with 2 columns).
`fpr`	A list of two components, `fpr` (the false positive rates) and `fpr.ci` the confidence intervals (a matrix with 2 columns).
`sens.htest`	An object of class `htest`.
`spec.htest`	An object of class `htest`.
`DOR`	A list of two components, `DOR` the diagnostic odds ratios and `DOR.ci` the confidence intervals (a matrix with 2 columns).
`posLR`	A list of two components, `posLR` the positive likelihood ratios and `posLR.ci` the confidence intervals (a matrix with 2 columns).
`negLR`	A list of two components, `negLR` the negative likelihood ratios and `negLR.ci` the confidence intervals (a matrix with 2 columns).
`cor_sens_fpr`	numeric, the correlation of the sensitivities and false-positive rates.
`level`	numeric
`method`	character
`names`	character vector, if the main argument of `madad` is a data frame with a variable `names` these names are stored here.
`nobs`	integer, number of primary studies.

`data`	data frame, with columns `TP`, `FN`, `FP` and `TN`.
`data.name`	character, name of the main argument.
`correction`	numeric
`correction.control`	character

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

See Also

`madauni`

Examples

```
data(AuditC)
AuditC.d <- madad(AuditC)
print(AuditC.d, digits = 2) #round everything to 2 digits
```

`madauni` *Meta-Analyisis of univariate measures of diagnostic accuracy*

Description

The classic strategy to meta-analysis of diagnostic accuracy data is to pool a univariate measure of accuracy like the diagnostic odds ratio, the positive likelihood ratio or the negative likelihood ratio. For fixed effect estimation a Mantel-Haenszel estimator is implemented and for random effect estimation a DerSimonian-Laird estimator is available.

Usage

```
madauni(x, type = "DOR", method = "DSL", suppress = TRUE, ...)
```

Arguments

`x`	any object that can be converted to a data frame with integer variables `TP`, `FN`, `FP` and `TN`, alternatively a matrix with column names including `TP`, `FN`, `FP` and `TN`.
`type`	character, what effect size should be pooled? Either `"DOR"`, `"posLR"` or `"negLR"`.
`method`	character, method of estimation. Either `"MH"` or `"DSL"`.
`suppress`	logical, should warnings produced by the internal call to `madad` be suppressed?
`...`	further arguments to be passed on to `madad`, for example `correction.control`.

Details

First note that the function `madad` is used to calculate effect measures. You can pass on arguments to this function via the `...` arguments. This is especially useful for the `correction.control` and `correction` arguments, see the example.

The Mantel-Haenszel method performs fixed effect estimation of the effect sizes. For the DOR the variance of this estimator is calculated according to Robins et al. (1986) and for the likelihood ratios the variance is based on Greenland et al. (1985).

The DerSimonian-Laird method performs a random effects meta-analysis. For this τ^2, the variance of the log-transformed effect size (DOR, positive or negative likelihood ratio) is calculated by the DerSimonian and Laird (1986) method. The confidence interval for τ^2 is derived by inverting the Q-Test of Viechtbauer (2007).

Zwindermann and Bossuyt (2008) argue, that univariate summary points like the likelihood ratios should be derived from the bivariate model of Reitsma et al (2005). The function `SummaryPts`, using output of `reitsma` supports this approach.

Value

An object of class `madauni`, for which some standard methods are available, see `madauni-class`

Note

Performing univariate meta-analysis of diagnostic studies can not be recommended anymore now that bivariate methods are available, at least not if a reasonable number of primary studies is availabel. The package `mada` provides this functionality for exploratory purposes and for meta-analysis of a small number of studies. The prefered way is to use `reitsma` in conjunction with `SummaryPts`.

The default value of `correction.control` used `madad` (and hence in the calculation of the effect sizes for `madauni`) is `"all"`, i.e. the continuity correction is added to all studies if any has a zero cell. This is a different default value than the `metafor` package uses. Set `correction.control` to `"single"` to arrive at the same values.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

References

DerSimonian, R. and Laird, N. (1986). “Meta-analysis in clinical trials.” *Controlled clinical trials*, **7**, 177–188.

Greenland, S. and Robins, J.M. (1985). “Estimation of a Common Effect Parameter from Sparse Follow-Up Data.” *Biometrics*, **41**, 55–68.

Reitsma, J., Glas, A., Rutjes, A., Scholten, R., Bossuyt, P., & Zwinderman, A. (2005). “Bivariate analysis of sensitivity and specificity produces informative summary measures in diagnostic reviews.” *Journal of Clinical Epidemiology*, **58**, 982–990.

Robins, J. and Greenland, S. and Breslow, N.E. (1986). “A general estimator for the variance of the Mantel-Haenszel odds ratio.” *American Journal of Epidemiology*, **124**, 719–723.

Viechtbauer, W. (2007). “Confidence intervals for the amount of heterogeneity in meta-analysis.” *Statistics in Medicine*, **26**, 37–52.

Zwinderman, A., & Bossuyt, P. (2008). “We should not pool diagnostic likelihood ratios in systematic reviews.”*Statistics in Medicine*, **27**, 687–697.

See Also

`madauni-class`, `reitsma`, `SummaryPts`

Examples

```
data(AuditC)

## First example: DOR meta-analysis
AuditC.uni <- madauni(AuditC)
summary(AuditC.uni)

## Second example: sensitivity analysis
## Do continuity corrections make a difference?
AuditC.uni_low <- madauni(AuditC, correction = 0.1)
AuditC.uni_single <-  madauni(AuditC,
          correction.control = "single") ## default is "all"
confint(AuditC.uni)
confint(AuditC.uni_low)
confint(AuditC.uni_single)
```

`madauni-class` *Methods for the class* `madauni`.

Description

Various methods for the output of the function `madauni`. Also the default method `confint` works for this class.

Usage

```
## S3 method for class 'madauni'
print(x, digits = 3, ...)
## S3 method for class 'madauni'
vcov(object, ...)
## S3 method for class 'madauni'
summary(object, level = 0.95, ...)
## S3 method for class 'summary.madauni'
print(x, digits = 3, ...)
```

Arguments

`x`	An object of class `madauni`.
`object`	An object of class `madauni`.
`level`	numeric, the confidence level for the confidence intervals in the summary.
`digits`	integer indicating the number of decimal places to round to.
`...`	arguments to be passed to methods

Value

`summary.madauni` returns a list of class `summary.madauni` which is printed with `print.summary.madauni`.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

See Also

`madauni`

`mslSROC` *Plot the Moses-Shapiro-Littenberg SROC curve*

Description

The approach to SROC curve modeling is described in the paper of Moses, Shapiro and Littenberg (1993). It is considered outdated and is included in `mada` so that users can reproduce older results and compare different SROC curves.

Usage

```
mslSROC(data = NULL, subset=NULL,
  TP="TP", FN="FN", FP="FP", TN="TN",
  fpr = NULL, extrapolate = FALSE,
  correction = 0.5, correction.control = "all",
  add = FALSE, lty = 1, lwd = 1, col = 1, ...)
```

Arguments

`data`	any object that can be converted to a data frame with integer variables for observed frequencies of true positives, false negatives, false positives and true negatives. The names of the variables are provided by the arguments `TP`, `FN`, `FP` and `TN` (see their defaults). Alternatively the data can be a matrix with column names including `TP`, `FN`, `FP` and `TN`. If no `data` is specified, the function will check the `TP`, `FN`, `FP` and `TN` arguments.

`TP`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`FN`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`FP`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`TN`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`subset`	the rows of `data` to be used as a subset in all calculations. If `NULL` (the default) then the complete data is considered.
`fpr`	Points between 0 and 1 on which to draw the SROC curve. Should be tightly spaced. If set to `NULL`, the default, it will be the vector of numbers `0.01`, `0.02`, `...`, `0.99` and is truncated if the `extrapolate` argument is `FALSE`.
`extrapolate`	logical, should the SROC curve be extrapolated beyond the region where false positive rates are observed?
`correction`	numeric, continuity correction applied if zero cells
`correction.control`	character, if set to `"all"` (the default) the continuity correction is added to the whole data if only one cell in one study is zero. If set to `"single"` the correction is only applied to rows of the data which have a zero.
`add`	logical, should the SROC curve be added to an existing plot?
`lty`	line type, see `lines`.
`lwd`	line width, see `lines`.
`col`	color of SROC, see `lines`.
`...`	arguments to be passed on to plotting functions.

Details

Details are found in the paper of Moses, Shapiro and Littenberg (1993).

Value

Besides plotting the SROC, an `invisible` list is returned which contains the parameters of the SROC.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

References

Moses L.E., Shapiro D., & Littenberg B. (1993) “Combining independent studies of a diagnostic test into a summary ROC curve: data-analytic approaches and some additional considerations.” *Statistics in Medicine*, **12**, 1293–1316.

See Also

`reitsma-class`, `talpha`, `SummaryPts`

Examples

```
## First Example
data(Dementia)
ROCellipse(Dementia)
mslSROC(Dementia, add = TRUE) # Add the MSL-SROC to this plot

## Second Example
# Make a fancy plot and look at the coefficients
msl_Dementia <- mslSROC(Dementia, col = 3, lwd = 3, lty = 3)
msl_Dementia$A2 # intercept on logit SROC space
msl_Dementia$B2 # slope on logit SROC space
```

`phm` *Diagnostic Meta-Analysis with the proportional hazards model approach of Holling et.al (2012)*

Description

The function fits the model of Holling et al. (2012). The adjusted profile maximum likelihood estimator (APMLE) is implemented for homogeneity and heterogeneity of primary studies.

Usage

```
phm(data, ...)
## Default S3 method:
phm(data = NULL, subset=NULL,
    TP="TP", FN="FN", FP="FP", TN="TN",
    correction = 0.5, correction.control = "all",
    hetero = TRUE, estimator = "APMLE", l = 100, ...)
```

Arguments

`data`	any object that can be converted to a data frame with integer variables `TP`, `FN`, `FP` and `TN`, alternatively a matrix with column names including `TP`, `FN`, `FP` and `TN`.
`subset`	the rows of `data` to be used as a subset in all calculations. If `NULL` (the default) then the complete data is considered.

`TP`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`FN`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`FP`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`TN`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`correction`	numeric, continuity correction applied if zero cells
`correction.control`	character, if set to `"all"` (the default) the continuity correction is added to the whole data if only one cell in one study is zero. If set to `"single"` the correction is only applied to rows of the data which have a zero.
`hetero`	logical, should heterogeneity of studies be assumed? Will fit model for homogeneity otherwise.
`estimator`	character, determines estimator used. Currently only `APMLE` is available.
`l`	interger, number of iterations for fixed point algorithm
`...`	arguments passed on to other functions (currently not used)

Details

The model of Holling et al. (2012) assumes that the relationship between false positive rates u and and sensitivities p can be described by

$$u^\theta = p,$$

where θ is the diagnostic accuracy parameter. If homogeneity of the studies can be assumed, θ is estimated as a fixed effect. Under heterogeneity a random effect with variance τ^2 describes the variation of the diagnostic accuracy parameter in the population of studies. Since the error of each observed θ depends only on the sample size and θ the model has only one parameter in the case of homogeneity and two parameters under heterogeneity, making it suitable for diagnostic meta-analysis with low sample size. Estimation proceeds by a fixed point algorithm derived from the adjusted profile likelihood. More details on the computational approach can be found in Holling et al. (2012).

Value

An object of the class `phm` for which many standard methods are available. See `phm-class` for details.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>, Walailuck Boehning (original implementation of estimation algorithm)

References

Holling, H., Boehning W., Boehning, D. (2012) “Meta-Analysis of Diagnostic Studies based upon SROC-Curves: a Mixed Model Approach using a Proportional Hazards Model.” *Statistical Modelling*, **12**, 347???-375.

See Also

`phm-class`

Examples

```
data(AuditC)
(fit <- phm(AuditC))
summary(fit)
plot(fit)
```

---

phm-class *Methods for* `phm` *objects.*

Description

Objects of the class `phm` are output by the function with the same name. Apart from standard methods the function `sroc` provides SROC curves and confidence bands for model fits.

Usage

```
## S3 method for class 'phm'
print(x, ...)
## S3 method for class 'phm'
summary(object, level = 0.95, ...)
## S3 method for class 'phm'
sroc(fit, fpr = 1:99/100, ...)
## S3 method for class 'phm'
plot(x, extrapolate = FALSE, confband = TRUE, level = 0.95,
     ylim = c(0,1), xlim = c(0,1), sroclty = 1, sroclwd = 1,
     confbandlty = 2, confbandlwd = 0.5, ...)
```

Arguments

`x`	a phm object.
`object`	a phm object.
`fit`	a phm object.
`level`	numeric, the confidence level for calculations of confidence intervals (`summary`) or confidence bands (`plot`).
`fpr`	numeric, the false positives rates for which to calculate the predicted sensitivities.

`extrapolate`	logical, should the sroc curve be plotted beyond the observed false positive rates?
`confband`	logical, should confidence bands be plotted?
`ylim`	numeric of length 2, which section of the sensitivities to plot?
`xlim`	numeric of length 2, which section of the false positive rates to plot?
`sroclty`	integer, line type of the SROC curve
`sroclwd`	integer, line width of the SROC curve
`confbandlty`	integer, line type of the SROC curve's confidence band
`confbandlwd`	integer, line width of the SROC curve's confidence band
`...`	arguments to be passed on to other functions

Details

The SROC curve is derived from the model formula. The confidence bands are calculated from the bounds of the confidence interval for the diagnostic accuracy parameter θ. The parameter and its confidence interval are then also used to calculate the AUC and partial AUC using the formulae

$$AUC(a,b) = \int_a^b u^\theta \mathrm{d}u = \frac{1}{\theta+1}[b^{\theta+1} - a^{\theta+1}],$$

$$AUC = AUC(0,1)$$

and

$$pAUC = \frac{1}{b-a} AUC(a,b),$$

where a is the lower bound of the observed false positive rates and b the upper.

Value

The `sroc` function returns a matrix ready for plotting. Each row corresponds to one point in ROC space.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

References

Holling, H., Boehning D., Boehning, W. (2012) “Meta-Analysis of Diagnostic Studies based upon SROC-Curves: a Mixed Model Approach using a Proportional Hazards Model.” *Statistical Modelling*, **12**, 347–375.

See Also

`phm`

Examples

```
# load data
data(AuditC)
# fit model
fit <- phm(AuditC)
#calculate a SROC curve, but do not plot it
sroc.AuditC <- sroc(fit)
# plot the SROC curve in ROC space as a line
plot(sroc.AuditC, type = "l")
# Fancy version using plot
plot(fit)
```

`predv_d` *Estimation of Distributions of Predictive Values Based on Prevalence Probability Distributions and Pooled Sensitivities and Specificities*

Description

Estimation of projected summary predictive values based on a prevalence probability distribution and pooled (meta-analytical) sensitivities and specificities. Probability distributions for negative and positive predictive values are obtained.

Usage

```
predv_d(x,prop_m,prop_sd,zb=TRUE,n_iter=100000,...)
```

Arguments

`x`	dataset containing data from the primary studies. It must correspond to any object that can be converted to a data frame with integer variables `TP`, `FN`, `FP` and `TN`, alternatively a matrix with column names including `TP`, `FN`, `FP` and `TN`. These respectively concern the numbers of true positives, true negatives, false positives, and false negatives for each primary study)
`prop_m`	mean value of the prevalence probability distribution. It must be stated as a proportion (i.e., as a numeric value between 0 and 1). If both prop_m and prop_sd are not defined, a probability distribution for the prevalence based on available primary studies' data will be computed (see details).
`prop_sd`	standard-deviation of the prevalence probability distribution. It must be stated as a value between 0 and 1. If both prop_m and prop_sd are not defined, a probability distribution for the prevalence based on available primary studies' data will be computed (see details).
`zb`	logical. If TRUE (default), the Zwindermann & Bossuyt approach will be used to generate samples for observed sensitivities and false positive rate (as in SummaryPts function). If FALSE, beta distributions will be obtained based on 95 percent confidence interval bounds of pooled sensitivities and specificities (while this latter approach may not take fully into account the correlation between sensitivity and false positive rate, it may lead to faster results).

`n_iter`	number of simulations being performed. Default value is 100,000.
`...`	further arguments to be passed on to `predv_d`.

Details

The predv_d function projects summary predictive values distributions from (i) a prevalence probability distribution, and (ii) pooled sensitivities and specificities obtained in the context of diagnostic test accuracy meta-analysis using a bivariate random-effects model. The bivariate random-effects model is equivalent to the hierarchical summary receiver operating characteristic model. By default, a sampling-based approach is used to generate samples for observed sensitivities and false positive rates. From these samples, and based on the prevalences probability distribution being considered, distributions of predictive values will be obtained based on the application of the Bayes theorem. The prevalence probability distribution can be obtained by providing a value for the mean (argument prop_m) and a value for the standard-deviation (argument prop_sd). If both prop_m and prop_sd are missing/not defined, a probability distribution for the prevalence based on available primary studies' data will be computed. That is, random-effects meta-analysis of log-transformed prevalences will be performed (using metafor) using data from included primary studies; the pooled results will then be used to obtain the probability distribution for prevalences. This may be a suboptimal option (as there may be considerable heterogeneity, diagnostic accuracy primary studies may not be the best ones to estimate the prevalence of a disease/condition...) compared to user-defined arguments, particularly if good prevalence studies exist.

Guided example

The dataset skin_tests contains results from a set of primary studies assessing the accuracy of skin tests for diagnosing penicillin allergy (they are part of the data analysed by Sousa-Pinto et al [2021]). This dataset contains four columns, displaying - for each primary study - the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). Let us now assume that the prevalence of penicillin allergy can be modeled by a probability distribution, having a mean of 0.05 (5 percent) and a standard-deviation of 0.015. Distributions of negative and positive predictive values can be estimated by:

```
predv_d(x=skin_tests,prop_m=0.05,prop_sd=0.015,zb=TRUE)
```

For negative predictive values, we obtain a probability distribution defined by a mean value of 0.96 and a standard-deviation of 0.01 (95 percent credible interval=0.93-0.98). For positive predictive values, we obtain a probability distribution defined by a mean value of 0.31 and a standard-deviation of 0.12 (95 percent credible interval=0.11-0.57). Values may differ slightly from the ones just described, as we are dealing with simulation results.

If we had no information on how the prevalence of penicillin allergy could be modeled by a probability distribution, we would opt for solely relying on data provided by included primary studies:

```
predv_d(x=skin_tests)
```

In that case, in addition to the results, we would get an warning message stating that considerable heterogeneity was found when doing meta-analysis of prevalences. Results should be carefully interpreted.

Value

An object of class `predv_d`, for which some standard methods are available, see `predv_d-class`. Some of the obtainable components include:

`SummaryData`	A dataframe displaying the mean, standard-deviation (SD) and percentiles (p) for the probability distribution of the summary negative predictive values ("NPV" row) and positive predictive values ("PPV" row).
`results_pred`	A dataframe displaying the results for all samples.

Author(s)

Bernardo Sousa-Pinto <bernardo@med.up.pt>

References

Reitsma, J., Glas, A., Rutjes, A., Scholten, R., Bossuyt, P., & Zwinderman, A. (2005). “Bivariate analysis of sensitivity and specificity produces informative summary measures in diagnostic reviews.” *Journal of Clinical Epidemiology*, **58**, 982–990.

Zwinderman, A., & Bossuyt, P. (2008). “We should not pool diagnostic likelihood ratios in systematic reviews.” *Statistics in Medicine*, **27**, 687–697.

Sousa-Pinto, B., Tarrio, I., Blumenthal, K.G., Azevedo, L.F., Delgado, L., & Fonseca, J.A. (2021). “Accuracy of penicillin allergy diagnostic tests: A systematic review and meta-analysis.” *Journal of Allergy and Clinical Immunology*, **147**, 296–308.

Joseph L, Belisle P. (2017). “Computing Beta distribution parameters.” [Internet] Accessible at: https://www.medicine.mcgill.ca/epidemiology/Joseph/PBelisle/BetaParmsFromQuantiles.html

See Also

`reitsma`, `SummaryPts`, `predv_r`

Examples

```
data(skin_tests)
pred_skin_tests <- predv_d(x=skin_tests,prop_m=0.05,prop_sd=0.015,zb=TRUE)
pred_skin_tests
```

`predv_d-class` *Methods for the class* `predv_d`.

Description

Various methods for the output of the function `predv_d`.

Usage

```
## S3 method for class 'predv_d'
print(x, xlim_npv=c(0,1),xlim_ppv=c(0,1), ...)
## S3 method for class 'predv_d'
summary(object, xlim_npv=c(0,1),xlim_ppv=c(0,1), ...)
```

Arguments

`x`	An object of class `predv_d`.
`object`	An object of class `predv_d`.
`xlim_npv`	limits of the x-axis for the plot on projected negative predictive values. Default is c(0,1).
`xlim_ppv`	limits of the x-axis for the plot on projected positive predictive values. Default is c(0,1).
`...`	arguments to be passed to methods

Value

`summary.predv_d` returns a list of class `summary.predv_d`.

Author(s)

Bernardo Sousa-Pinto <bernardo@med.up.pt>

See Also

`predv_d`

`predv_r` *Estimation of Distributions of Predictive Values Based on Prevalence Ranges and Pooled Sensitivities and Specificities*

Description

Estimation of projected summary predictive values based on a prevalence range and pooled (meta-analytical) sensitivities and specificities. A probability distribution for the negative and positive predictive values are obtained for each prevalence value within a predetermined range.

Usage

```
predv_r(x,prop_min,prop_max,zb=TRUE,n_iter=100000,...)
```

Arguments

`x`	dataset containing data from the primary studies. It must correspond to any object that can be converted to a data frame with integer variables `TP`, `FN`, `FP` and `TN`, alternatively a matrix with column names including `TP`, `FN`, `FP` and `TN`. These respectively concern the numbers of true positives, true negatives, false positives, and false negatives for each primary study)
`prop_min`	minimum prevalence value being considered. It must be stated as a proportion (i.e., as a numeric value between 0 and 1). If both prop_min and prop_max are not defined, a prevalence range based on available primary studies' data will be computed (see details).

`zb`	logical. If TRUE (default), the Zwindermann & Bossuyt approach will be used to generate samples for observed sensitivities and false positive rate (as in SummaryPts function). If FALSE, beta distributions will be obtained based on 95 percent confidence interval bounds of pooled sensitivities and specificities (while this latter approach may not take fully into account the correlation between sensitivity and false positive rate, it may lead to faster results).
`prop_max`	maximum prevalence value being considered. It must be stated as a proportion (i.e., as a numeric value between 0 and 1). If both prop_min and prop_max are not defined, a prevalence range based on available primary studies' data will be computed (see details).
`n_iter`	number of simulations being performed. Default value is 100,000.
`...`	further arguments to be passed on to `predv_r`.

Details

The predv_r function projects summary predictive values from (i) a prevalence range, and (ii) pooled sensitivities and specificities obtained in the context of diagnostic test accuracy meta-analysis using a bivariate random-effects model. The bivariate random-effects model is equivalent to the hierarchical summary receiver operating characteristic model. By default, a sampling-based approach is used to generate samples for observed sensitivities and false positive rate. From these samples, and for each prevalence value within the range being considered, distributions of predictive values will be obtained based on the application of the Bayes theorem. The prevalence range can be user-defined, by providing a value for the minimum (argument prop_min) and a value for the maximum value of that range (argument prop_max). If both prop_min and prop_max are missing/not defined, a prevalence range based on available primary studies' data will be computed. That is, the lowest and highest frequency of patients with disease/condition across included primary studies will be considered. This may be a suboptimal option compared to user-defined arguments, particularly if good prevalence studies are available.

Guided example

The dataset skin_tests contains results from a set of primary studies assessing the accuracy of skin tests for diagnosing penicillin allergy (they are part of the data analysed by Sousa-Pinto et al [2021]). This dataset contains four columns, displaying - for each primary study - the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). Let us assume that the prevalence of penicillin allergy ranges between 0.01 and 0.10 (1 and 10 percent). Pooled negative and positive predictive values can be estimated by:

```
predv_r(x=skin_tests,prop_min=0.01,prop_max=0.15,zb=TRUE)
```

The results indicate that the point estimates for the negative predictive value range between 0.88 (prevalence=0.15) and 0.99 (prevalence=0.01). For the positive predictive value, point estimates range between 0.09 (prevalence=0.01) and 0.59 (prevalence=0.15), although uncertainty is particularly high for the latter estimate (95 percent credible interval=0.36-0.80). Values may differ slightly from the ones just described, as we are dealing with simulation results.

If we had no information on how the prevalence range of penicillin allergy, we would opt for solely relying on data provided by included primary studies:

```
pred_skin_tests1 <- predv_r(x=skin_tests)
```

Value

An object of class `predv_r`, for which some standard methods are available, see `predv_r-class`. Some of the obtainable components include:

`NPV`	A dataframe displaying the mean, standard-deviation (SD) and percentiles (p) for the probability distribution of negative predictive values for each prevalence value within the defined range.
`PPV`	A dataframe displaying the mean, standard-deviation (SD) and percentiles (p) for the probability distribution of positive predictive values for each prevalence value within the defined range.

Author(s)

Bernardo Sousa-Pinto <bernardo@med.up.pt>

References

Reitsma, J., Glas, A., Rutjes, A., Scholten, R., Bossuyt, P., & Zwinderman, A. (2005). "Bivariate analysis of sensitivity and specificity produces informative summary measures in diagnostic reviews." *Journal of Clinical Epidemiology*, **58**, 982–990.

Zwinderman, A., & Bossuyt, P. (2008). "We should not pool diagnostic likelihood ratios in systematic reviews." *Statistics in Medicine*, **27**, 687–697.

Sousa-Pinto, B., Tarrio, I., Blumenthal, K.G., Azevedo, L.F., Delgado, L., & Fonseca, J.A. (2021). "Accuracy of penicillin allergy diagnostic tests: A systematic review and meta-analysis." *Journal of Allergy and Clinical Immunology*, **147**, 296–308.

See Also

`reitsma`, `SummaryPts`, `predv_d`

Examples

```
data(skin_tests)
pred_skin_tests <- predv_r(x=skin_tests,prop_min=0.01,prop_max=0.15,zb=TRUE)
pred_skin_tests
```

`predv_r-class` *Methods for the class* `predv_r`.

Description

Various methods for the output of the function `predv_r`.

Usage

```
## S3 method for class 'predv_r'
print(x, ylim_npv=c(0,1),ylim_ppv=c(0,1), ...)
## S3 method for class 'predv_r'
summary(object, ylim_npv=c(0,1),ylim_ppv=c(0,1), ...)
```

Arguments

`x`	An object of class `predv_r`.
`object`	An object of class `predv_r`.
`ylim_npv`	limits of the y-axis for the plot on projected negative predictive values. Default is c(0,1).
`ylim_ppv`	limits of the y-axis for the plot on projected positive predictive values. Default is c(0,1).
`...`	arguments to be passed to methods

Value

`summary.predv_r` returns a list of class `summary.predv_r`.

Author(s)

Bernardo Sousa-Pinto <bernardo@med.up.pt>

See Also

`predv_r`

`reitsma` *Fit the bivariate model of Reitsma et al. (2005) and extensions.*

Description

The function fits the bivariate model of Reitsma et al. (2005) that Harbord et al. (2007) have shown to be equivalent to the HSROC of Rutter&Gatsonis (2001). We specify the model as a linear mixed model with known variances of the random effects, similar to the computational approach by Reitsma et al. (2005). Variance components are estimated by restricted maximum likelihood (REML) as a default but ML estimation is available as well. In addition meta-regression is possible and the use of other transformations than the logit, using the approach of Doebler et al. (2012).

Usage

```
reitsma(data, ...)
## Default S3 method:
reitsma(data = NULL, subset=NULL, formula = NULL,
        TP="TP", FN="FN", FP="FP", TN="TN",
        alphasens = 1, alphafpr = 1,
        correction = 0.5, correction.control = "all",
        method = "reml",
        control = list(), ...)
```

Arguments

`data`	any object that can be converted to a data frame with integer variables for observed frequencies of true positives, false negatives, false positives and true negatives. The names of the variables are provided by the arguments `TP`, `FN`, `FP` and `TN` (see their defaults). Alternatively the data can be a matrix with column names including `TP`, `FN`, `FP` and `TN`. If no `data` is specified, the function will check the `TP`, `FN`, `FP` and `TN` arguments.
`TP`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`FN`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`FP`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`TN`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`subset`	the rows of `data` to be used as a subset in all calculations. If `NULL` (the default) then the complete data is considered.
`formula`	Formula for meta-regression using standard `formula`. The left hand side of this formula must be `cbind(tsens, tfpr)` and if `formula` is `NULL` (the default), then the formula `cbind(tsens, tfpr) ~ 1` is used, i.e. a model without covariates.
`alphasens`	Transformation parameter for (continuity corrected) sensitivities, see details. If set to 1 (the default) the logit transformation is used.
`alphafpr`	Transformation parameter for (continuity corrected) false positive rates, see details
`correction`	numeric, continuity correction applied if zero cells
`correction.control`	character, if set to `"all"` (the default) the continuity correction is added to the whole data if only one cell in one study is zero. If set to `"single"` the correction is only applied to rows of the data which have a zero.
`method`	character, either `"fixed"`, `"ml"`, `"mm"`, `"vc"` or `"reml"` (the default)

`control`	a list of control parameters, see the documentation of `mvmeta`
.	
`...`	arguments to be passed on to other functions, currently ignored

Details

In a first step the observed frequencies are continuity corrected if values of 0 or 1 would result for the sensitivity or false positive rate otherwise. Then the sensitivities and false positive rates are transformed using the transformation

$$x \mapsto t_\alpha(x) := \alpha \log(x) - (2-\alpha)\log(1-x).$$

Note that for $\alpha = 1$, the default value, the logit transformation results, i.e. the approach of Reitsma et al. (2005). A bivariate random effects model is then fitted to the pairs of transformed sensitivities and false positive rates.

Parameter estimation makes use of the fact that the fixed effect parameters can be profiled in the likelihood. Internally the function `mvmeta` is called. Currently only standard errors for the fixed effects are available. Note that when using `method = "mm"` or `method = "vc"`, no likelihood can be computed and hence no AIC or BIC values.

If you want other summary points like negative or positive likelihood ratios, see `SummaryPts`, while for positive or negative predictive values, see `predv_r` and `predv_d`.

Value

An object of the class `reitsma` for which many standard methods are available. See `reitsma-class` for details.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

References

Rutter, C., & Gatsonis, C. (2001). “A hierarchical regression approach to meta-analysis of diagnostic test accuracy evaluations.” *Statistics in Medicine*, **20**, 2865–2884.

Reitsma, J., Glas, A., Rutjes, A., Scholten, R., Bossuyt, P., & Zwinderman, A. (2005). “Bivariate analysis of sensitivity and specificity produces informative summary measures in diagnostic reviews.” *Journal of Clinical Epidemiology*, **58**, 982–990.

Harbord, R., Deeks, J., Egger, M., Whiting, P., & Sterne, J. (2007). “A unification of models for meta-analysis of diagnostic accuracy studies.” *Biostatistics*, **8**, 239–251.

Doebler, P., Holling, H., Boehning, D. (2012) “A Mixed Model Approach to Meta-Analysis of Diagnostic Studies with Binary Test Outcome.” *Psychological Methods*, to appear

See Also

`reitsma-class`, `talpha`, `SummaryPts`

Examples

```
data(Dementia)
(fit <- reitsma(Dementia))
summary(fit)
plot(fit)

## Meta-Regression
data(smoking) # contains more than one 2x2-table
## reduce to subset of independent 2x2-tables by using the
## first table from each study only
smoking1 <- subset(smoking, smoking$result_id == 1)
## use type of questionnaire as covariate
(fit <- reitsma(smoking1, formula = cbind(tsens, tfpr) ~ type))
summary(fit) ## sensitivities significantly lower for SAQ
```

`reitsma-class` *Methods for* `reitsma` *objects.*

Description

Objects of the class `reitsma` are output by the function with the same name. Apart from standard methods the functions `sroc`, `mcsroc` and `ROCellipse` provide SROC curves and confidence regions for fits.

Usage

```
## S3 method for class 'reitsma'
print(x, digits = 4, ...)
## S3 method for class 'reitsma'
summary(object, level = 0.95, sroc.type = "ruttergatsonis", ...)
## S3 method for class 'reitsma'
sroc(fit, fpr = 1:99/100, type = "ruttergatsonis", return_function = FALSE, ...)
## S3 method for class 'reitsma'
mcsroc(fit, fpr = 1:99/100, replications = 10000, lambda = 100, ...)
## S3 method for class 'reitsma'
ROCellipse(x, level = 0.95, add = FALSE, pch = 1, ...)
## S3 method for class 'reitsma'
crosshair(x, level = 0.95, length = 0.1, pch = 1, ...)
## S3 method for class 'reitsma'
plot(x, extrapolate = FALSE, plotsumm = TRUE, level = 0.95,
     ylim = c(0,1), xlim = c(0,1), pch = 1, sroclty = 1, sroclwd = 1,
     predict = FALSE, predlty = 3, predlwd = 1, type = "ruttergatsonis", ...)
## S3 method for class 'reitsma'
anova(object, fit2, ...)
## S3 method for class 'anova.reitsma'
print(x, digits = 4, ...)
```

Arguments

`x`	a `reitsma` object.
`object`	a `reitsma` object.
`fit`	a `reitsma` object.
`fit2`	a `reitsma` object.
`digits`	number of decimal digits to print.
`level`	numeric, the level for calculations of confidence intervals (`summary`) or regions (`ROCellipse`)
`sroc.type`	character, which SROC curve should be used to calculate the AUC in the summary? Besides the default `ruttergatsonis` the option `naive` is available.
`return_function`	logical. Should a function on ROC space be returned or the values at the points given by `fpr`?
`fpr`	numeric, the false positives rates for which to calculate the predicted sensitivities
`replications`	integer, the number of replications for the Monte-Carlo SROC curve
`lambda`	numeric, the parameter lambda of the Monte-Carlo run, see details
`add`	logical, should the confidence region be added to the current plot? If set to `FALSE` a matrix of points of the ellipse is returned
`extrapolate`	logical, should the SROC curve be plotted beyond the observed false positive rates?
`plotsumm`	logical, should the summary pair of sensitivity and false positive rate together with its confidence region be plotted?
`length`	positve numeric, length of the "whiskers" of the crosshairs.
`ylim`	numeric of length 2, which section of the sensitivities to plot?
`xlim`	numeric of length 2, which section of the false positive rates to plot?
`pch`	integer, symbol for the pair of mean sensitivity and false positive rate
`sroclty`	integer, line type of the SROC curve
`sroclwd`	integer, line width of the SROC curve
`predict`	logical, draw prediction region?
`predlty`	integer, line type of prediction region
`predlwd`	integer, line width of prediction region
`type`	character, type of SROC curve to plot. Can be either the generalization of the Rutter & Gatsonis (2001) SROC curve (see below) or the naive curve implied the bivariate model.
`...`	arguments to be passed on to other functions

Details

The confidence regions of `ROCellipse` are first calculated as ellipses on logit-ROC space, so the back-transformed regions that are output are not necessarily ellipses. The Monte-Carlo SROC curves are generated from random samples from the fitted model and a `lowess` smooth through them is output. Many computational details are to be found in Doebler et al. (2012).

The `summary` function for `reitsma` objects also contains the five parameters of the HSROC model by Rutter & Gatsonis (2001) if no regression is performed. These values are calculated by using the formulae from Harbord et al. (2007).

The `plot` method for `reitsma` objects will plot the generalization of the Rutter-Gatsonis curve.

If you require positive or negative likelihood ratios, you should use `SummaryPts`. If you require positive or negative predictive values, see `predv_r` and `predv_d`.

Value

`sroc` returns a matrix ready for plotting. Each row corresponds to one point in ROC space. `mcsroc` returns a `lowess` smooth. `ROCellipse` returns a list, the first element being a matrix of points in ROC space that delimit the confidence region and the second is the point estimate of the pair of sensitivity and false positive rate in ROC space.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

References

Doebler, P., Holling, H., Boehning, D. (2012) “A Mixed Model Approach to Meta-Analysis of Diagnostic Studies with Binary Test Outcome.” *Psychological Methods*, to appear

See Also

`reitsma`, `SummaryPts`

Examples

```
# load data
data(Dementia)
# fit model
fit <- reitsma(Dementia)
# calculate a confidence region but do not plot it
cr.Dementia <- ROCellipse(fit)
#calculate a SROC curve
sroc.Dementia <- sroc(fit)
# plot the confidence region in ROC space as a line
plot(cr.Dementia$ROCellipse, type = "l", xlim = c(0,1), ylim = c(0,1))
# add the point estimate of the mean
points(cr.Dementia$fprsens)
# add the SROC curve
lines(sroc.Dementia)
```

`ROCellipse` *Confidence Regions on ROC space*

Description

Plot individual confidence regions for the estimate from each primary study on ROC space or add such regions to an existing plot.

Usage

```
## Default S3 method:
ROCellipse(x, correction = 0.5, level = 0.95,
           xlim = c(0, 1), ylim = c(0, 1), method = "wilson",
           pch = 1, add = FALSE, corr = 0, suppress = TRUE,
           ellipsecol = "grey", ...)
```

Arguments

`x`	a data frame with variables including `TP`, `FN`, `FP`, `TN`, alternatively a matrix with column names including these.
`correction`	numeric, continuity correction applied to zero cells.
`level`	numeric, confidence level for the calculations of confidence intervals.
`xlim`	numeric of length 2, which portion of ROC space should be plotted? All reasonable values should be within (0,1).
`ylim`	numeric of length 2, which portion of ROC space should be plotted? All reasonable values should be within (0,1).
`method`	character, method used to calculate the confidence intervals for sensitivities, specificities and false positive rates. One of `"wald"`, `"wilson"`, `"agresti-coull"`, `"jeffreys"`, `"modified wilson"`, `"modified jeffreys"`, `"clopper-pearson"`, `"arcsine"`, `"logit"`, `"witting"`
`pch`	Symbol used to plot point estimates. Use `pch = ""` to suppress plotting point estimates.
`add`	logical, should the plot be added to the current plot?
`corr`	numeric or character, the correlation assumed in the calculation of the confidence ellipsoids on logit-ROC space. If set to `"logit"`, the correlation of the logit-transformed sensitivities and false positive rates will be used in the correlations. See details for further explanation.
`suppress`	logical, should the warnings produced by the internal call to `madad` be suppressed? Defaults to `TRUE`, since only the diagnostic accuracies and their confidence intervals are used in subsequent calculations.
`ellipsecol`	The color used for plotting the ellipses.
`...`	further arguments passed on to `plot`.

Details

The confindence regions are ellipses on logit-ROC space, hence the name of the function. The standard deviations underlying confidence intervals for the sensitivities and false positive rates are used to determine the scale of the ellipses on logit-ROC space. These ellipses get backtransformed to ROC space and plotted. As a default no correlation is assumed on logit-ROC space.

The objects of class `reitsma` have their own `ROCellipse` method to add a confidence region for the pooled estimate, see `reitsma-class`.

Value

Besides plotting an invisble `NULL` is returned.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

See Also

`crosshair`, `reitsma-class`

Examples

```
data(AuditC)
ROCellipse(AuditC)
```

---

rsSROC *Plot the Ruecker-Schumacher (2010) SROC curve*

Description

Assuming that a weighted Youden index is maximized in all primary studies, the Ruecker-Schumacher approach estimates individual ROC curves and then averages them.

Usage

```
rsSROC(data = NULL, subset=NULL,
  TP="TP", FN="FN", FP="FP", TN="TN",
  lambda = "from_bivariate",
  fpr = NULL, extrapolate = FALSE, plotstudies = FALSE,
  correction = 0.5, correction.control = "all",
  add = FALSE, lty = 1, lwd = 1, col = 1, ...)
```

Arguments

`data`	any object that can be converted to a data frame with integer variables for observed frequencies of true positives, false negatives, false positives and true negatives. The names of the variables are provided by the arguments `TP`, `FN`, `FP` and `TN` (see their defaults). Alternatively the data can be a matrix with column names including `TP`, `FN`, `FP` and `TN`. If no `data` is specified, the function will check the `TP`, `FN`, `FP` and `TN` arguments.
`TP`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`FN`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`FP`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`TN`	character or integer: name for vector of integers that is a variable of `data` or a vector of integers. If `data` is not `NULL`, names are expected, otherwise integers are.
`subset`	the rows of `data` to be used as a subset in all calculations. If `NULL` (the default) then the complete data is considered.
`lambda`	numeric or `"from_bivariate"`, the weight of the weighted Youden index. Must be between 0 and 1. If set to `"from_bivariate"`, the `reitsma` function is used to calculate lambda from the data.
`fpr`	Points between 0 and 1 on which to draw the SROC curve. Should be tightly spaced. If set to `NULL`, the default, it will be the vector of numbers `0.01, 0.02, ..., 0.99` and is truncated if the `extrapolate` argument is `FALSE`.
`extrapolate`	logical, should the SROC curve be extrapolated beyond the region where false positive rates are observed?
`plotstudies`	logical, should the ROC curves for the individual studies be added to the plot? The plot will become crowded if set to `TRUE`.
`correction`	numeric, continuity correction applied if zero cells
`correction.control`	character, if set to `"all"` (the default) the continuity correction is added to the whole data if only one cell in one study is zero. If set to `"single"` the correction is only applied to rows of the data which have a zero.
`add`	logical, should the SROC curve be added to an existing plot?
`lty`	line type, see `lines`.
`lwd`	line width, see `lines`.
`col`	color of SROC, see `lines`.
`...`	arguments to be passed on to plotting functions.

Details

Details are found in the paper of Ruecker and Schumacher (2010).

Value

Besides plotting the SROC, an `invisible` list is returned which contains the parameters of the SROC.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com> Original code kindly supplied by G. Ruecker.

References

Ruecker G., & Schumacher M. (2010) “Summary ROC curve based on a weighted Youden index for selecting an optimal cutpoint in meta-analysis of diagnostic accuracy.” *Statistics in Medicine*, **29**, 3069–3078.

See Also

`reitsma-class`, `talpha`, `SummaryPts`

Examples

```
## First Example
data(Dementia)
ROCellipse(Dementia)
rsSROC(Dementia, add = TRUE) # Add the RS-SROC to this plot

## Second Example
# Make a crowded plot and look at the coefficients
rs_Dementia <- rsSROC(Dementia, col = 3, lwd = 3, lty = 3,
                      plotstudies = TRUE)
rs_Dementia$lambda
rs_Dementia$aa # intercepts of primary studies on logit ROC space
rs_Dementia$bb # slopes
```

`sens` *Sensitivity, Specificity and False Positive Rate*

Description

Calculate basic measures of diagnostic accuracy for a number of studies.

Usage

```
sens(x)
spec(x)
fpr(x)
```

Arguments

`x`	a data frame with variables including `TP`, `FN`, `FP`, `TN`, alternatively a matrix with column names including these.

Details

These functions are the basic building blocks of many procedures to assess diagnostic accuracy. For a decent summary of set of primary studies it is better to use `madad`, for graphical summaries `crosshair` and `ROCellipse` are available.

Value

A numeric vector.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

See Also

`madad`, `crosshair`, `link{ROC.ellipse}`

Examples

```
data(AuditC)
plot(fpr(AuditC), sens(AuditC), main = "AUDIT-C data on ROC space",
     ylab = "Sensitivity", xlab = "False Positive Rate")
```

`SummaryPts` *Use the Zwindermann & Bossuyt (2008) MCMC procedure to generate summary points (positive and negative likelihood ratio, diagnostic odds ratio) for the Reitsma et al. (2005) bivariate model*

Description

Zwindermann & Bossuyt (2008) argue that likelihood ratios should not be pooled by univariate meta-analysis. They propose a sampling based approach that uses the parameters of a fit to the bivariate model (implemented in `reitsma`) to generate samples for observed sensitivities and false positive rates. From these samples the quantities of interest (and their confidence intervals) are estimated.

Usage

```
SummaryPts(object, ...)
## Default S3 method:
SummaryPts(object, mu,Sigma,alphasens = 1, alphafpr = 1,
                           n.iter = 10^6, FUN, ...)
## S3 method for class 'reitsma'
SummaryPts(object, n.iter = 10^6, FUN = NULL, ...)
## S3 method for class 'SummaryPts'
print(x, ...)
## S3 method for class 'SummaryPts'
summary(object, level = 0.95, digits = 3, ...)
```

Arguments

`object`	an object for which a method exists
`x`	An object of class `SummaryPts`
`mu`	numeric of length 2, expected to be the mean parameter of a bivariate model
`Sigma`	2x2 variance covariance matrix, expected to be the matrix representing the standard error of `mu` and the covariance of these two estimates
`alphasens`	numeric, alpha parameter for the sensitivities. Amounts to logit transformation if set to 1 (the default). See `reitsma`.
`alphafpr`	numeric, alpha parameter for the false positive rates. Amounts to logit transformation if set to 1 (the default). See `reitsma`.
`n.iter`	number of samples
`FUN`	A list of functions with 2 arguments (`sens` and `fpr`); if set to `NULL` in `SummaryPts.reitsma`, the positive, negative and inverse negative likelihood ratios are calculated and also the diagnostic odds ratio (DOR). See the example on how to supply other functions.
`level`	numeric, confidence level for confidence intervals
`digits`	number of significant digits to display
`...`	arguments to be passed on to other functions, currently ignored

Details

Samples are generated from a bivariate normal using `rmvnorm`. Note that the FUN argument

Value

An object of the class `SummaryPts` for which `print` and `summary` methods are available.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

References

Reitsma, J., Glas, A., Rutjes, A., Scholten, R., Bossuyt, P., & Zwinderman, A. (2005). “Bivariate analysis of sensitivity and specificity produces informative summary measures in diagnostic reviews.” *Journal of Clinical Epidemiology*, **58**, 982–990.

Zwinderman, A., & Bossuyt, P. (2008). “We should not pool diagnostic likelihood ratios in systematic reviews.”*Statistics in Medicine*, **27**, 687–697.

See Also

`reitsma`, `talpha`

Examples

```
data(Dementia)
(fit <- reitsma(Dementia))
mcmc_sum <- SummaryPts(fit, n.iter = 10^3)
## n.iter should be larger in applications!
mcmc_sum #just the means
summary(mcmc_sum) # 95% CIs by default
summary(mcmc_sum, level = 0.80, digits = 5) ## more digits, smaller CIs

## Supplying other functions

# Example 1: theta parameter of proportional hazards model
# see "phm" in mada's documentation for details on theta
theta <- function(sens,fpr){log(sens) / log(fpr)}
theta_sum <- SummaryPts(fit, FUN = list(theta = theta), n.iter = 10^3)
## n.iter should be larger in applications!
summary(theta_sum)
# compare with phm:
summary(phm(Dementia)) # the two estimators almost agree in this example

# Example 2: Youden index
Youden <- function(sens, fpr){sens - fpr}
Youden_sum <- SummaryPts(fit, FUN = list(Youden = Youden), , n.iter = 10^3)
## n.iter should be larger in applications!
summary(Youden_sum)
```

`talpha` *The t_α transformation as a link function for binary GLMs.*

Description

A parametric link function, i.e. a family of link functions intended for binary data.

Usage

```
talpha(alpha, verbose = FALSE,
  splineinv = TRUE, eps = 2 * .Machine$double.eps, maxit = 100)
```

Arguments

`alpha`	numeric, must be larger than 0 and smaller than 2.
`verbose`	logical, warn if truncation occurs when link function or inverse are used.
`splineinv`	logical, use spline interpolation for calculation of inverse link?
`eps`	if splineinv is `FALSE`, a Newton-Raphson algorithm is run to calculate the inverse. The argument `eps` determines when to terminate this algorithm. Ignored if splineinv is `TRUE`.
`maxit`	maximum number of iterations for Newton-Raphson. Ignored if splineinv is `TRUE`.

Value

An object of class `"link-glm"`, see `family` and `family`. Intended for use with `glm`.

Author(s)

Philipp Doebler <philipp.doebler@googlemail.com>

Examples

```
canonical <- binomial(link = talpha(1)) # logit-link
talpha_fam <- function(alpha)binomial(link = talpha(alpha)) # talpha family
## A call to glm might look like this: glm(formula, family = talpha_fam(1.5))
```

Index