

Package ‘MKclass’

October 10, 2020

Version 0.3

Date 2020-10-04

Title Statistical Classification

Author Matthias Kohl [aut, cre] (<<https://orcid.org/0000-0001-9514-8910>>)

Maintainer Matthias Kohl <Matthias.Kohl@stamats.de>

Depends R(>= 4.0.0)

Imports stats

Suggests knitr, rmarkdown, foreach, parallel, doParallel

VignetteBuilder knitr

Description Performance measures and scores for statistical classification such as accuracy, sensitivity, specificity, recall, similarity coefficients, AUC, GINI index, Brier score and many more. Calculation of optimal cut-offs and decision stumps (Iba and Langley (1991), <[doi:10.1016/B978-1-55860-247-2.50035-8](https://doi.org/10.1016/B978-1-55860-247-2.50035-8)>) for all implemented performance measures. Hosmer-Lemeshow goodness of fit tests (Lemeshow and Hosmer (1982), <[doi:10.1093/oxfordjournals.aje.a113284](https://doi.org/10.1093/oxfordjournals.aje.a113284)>; Hosmer et al (1997), <[doi:10.1002/\(SICI\)1097-0258\(19970515\)16:9%3C965::AID-SIM509%3E3.0.CO;2-O](https://doi.org/10.1002/(SICI)1097-0258(19970515)16:9%3C965::AID-SIM509%3E3.0.CO;2-O)>). Statistical and epidemiological risk measures such as relative risk, odds ratio, number needed to treat (Porta (2014), <[doi:10.1093%2Facref%2F9780199976720.001.0001](https://doi.org/10.1093%2Facref%2F9780199976720.001.0001)>).

License LGPL-3

URL <http://www.stamats.de/>

NeedsCompilation no

Repository CRAN

Date/Publication 2020-10-10 12:50:02 UTC

R topics documented:

MKclass-package	2
AUC	3
AUC.test	4
confMatrix	5
decisionStump	6

HLgof.test	7
optCutoff	9
or2rr	10
pairwise.auc	11
perfMeasures	12
perfScores	15
predValues	17
risks	18
rrCI	19
Index	21

MKclass-package	<i>Statistical Classification.</i>
-----------------	------------------------------------

Description

Performance measures and scores for statistical classification such as accuracy, sensitivity, specificity, recall, similarity coefficients, AUC, GINI index, Brier score and many more. Calculation of optimal cut-offs and decision stumps (Iba and Langley (1991), <doi:10.1016/B978-1-55860-247-2.50035-8>) for all implemented performance measures. Hosmer-Lemeshow goodness of fit tests (Lemeshow and Hosmer (1982), <doi:10.1093/oxfordjournals.aje.a113284>; Hosmer et al (1997), <doi:10.1002/(SICI)1097-0258(19970515)16:9%3C965::AID-SIM509%3E3.0.CO;2-O>). Statistical and epidemiological risk measures such as relative risk, odds ratio, number needed to treat (Porta (2014), <doi:10.1093%2Facref%2F9780199976720.001.0001>).

Details

```

Package:  MKclass
Type:    Package
Version:  0.4
Date:    2020-10-03
Depends: R(>= 4.0.0)
Imports: stats
Suggests: knitr, rmarkdown, foreach, parallel, doParallel
License:  LGPL-3
URL:     http://www.stamats.de/

```

```
library(MKclass)
```

Author(s)

Matthias Kohl <http://www.stamats.de>

Maintainer: Matthias Kohl <matthias.kohl@stamats.de>

AUC	<i>Compute AUC</i>
-----	--------------------

Description

The function computes AUC.

Usage

```
AUC(x, y, group, switchAUC = TRUE)
```

Arguments

x	numeric vector.
y	numeric vector. If missing, group has to be specified.
group	grouping vector or factor.
switchAUC	logical value. Switch AUC; see Details section.

Details

The function computes the area under the receiver operating characteristic curve (AUC under ROC curve).

If $AUC < 0.5$, a warning is printed and $1-AUC$ is returned. This behaviour can be suppressed by using `switchAUC = FALSE`

The implementation uses the connection of AUC to the Wilcoxon rank sum test; see Hanley and McNeil (1982).

Value

AUC value.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References

J. A. Hanley and B. J. McNeil (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, **143**, 29-36.

Examples

```
set.seed(13)
x <- rnorm(100) ## assumed as log2-data
g <- sample(1:2, 100, replace = TRUE)
AUC(x, group = g)
## avoid switching AUC
AUC(x, group = g, switchAUC = FALSE)
```

AUC.test

AUC-Test

Description

Performs tests for one and two AUCs.

Usage

```
AUC.test(pred1, lab1, pred2, lab2, conf.level = 0.95, paired = FALSE)
```

Arguments

pred1	numeric vector.
lab1	grouping vector or factor for pred1.
pred2	numeric vector.
lab2	grouping vector or factor for pred2.
conf.level	confidence level of the interval.
paired	not yet implemented.

Details

If pred2 and lab2 are missing, the AUC for pred1 and lab1 is tested using the Wilcoxon signed rank test; see [wilcox.test](#).

If pred1 and lab1 as well as pred2 and lab2 are specified, the Hanley and McNeil test (cf. Hanley and McNeil (1982)) is computed.

Value

A list with AUC, SE and confidence interval as well as the corresponding test result.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References

J. A. Hanley and B. J. McNeil (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, **143**, 29-36.

See Also

[wilcox.test](#), [AUC](#)

Examples

```
set.seed(13)
x <- rnorm(100) ## assumed as log2-data
g <- sample(1:2, 100, replace = TRUE)
AUC.test(x, g)
y <- rnorm(100) ## assumed as log2-data
h <- sample(1:2, 100, replace = TRUE)
AUC.test(x, g, y, h)
```

confMatrix

Compute Confusion Matrix

Description

The function computes the confusion matrix of a binary classification.

Usage

```
confMatrix(pred, pred.group, truth, namePos, cutoff = 0.5, relative = TRUE)
```

Arguments

pred	numeric values that shall be used for classification; e.g. probabilities to belong to the positive group.
pred.group	vector or factor including the predicted group. If missing, pred.group is computed from pred, where pred >= cutoff is classified as positive.
truth	true grouping vector or factor.
namePos	value representing the positive group.
cutoff	cutoff value used for classification.
relative	logical: absolute and relative values.

Details

The function computes the confusion matrix of a binary classification consisting of the number of true positive (TP), false negative (FN), false positive (FP) and true negative (TN) predictions.

In addition, their relative counterparts true positive rate (TPR), false negative rate (FNR), false positive rate (FPR) and true negative rate (TNR) can be computed.

Value

matrix or list of matrices with respective numbers of true and false predictions.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References

Wikipedia contributors. (2019, July 18). Confusion matrix. In Wikipedia, The Free Encyclopedia. Retrieved 06:00, August 21, 2019, from https://en.wikipedia.org/w/index.php?title=Confusion_matrix&oldid=906886050

Examples

```
## example from dataset infert
fit <- glm(case ~ spontaneous+induced, data = infert, family = binomial())
pred <- predict(fit, type = "response")

## with group numbers
confMatrix(pred, truth = infert$case, namePos = 1)

## with group names
my.case <- factor(infert$case, labels = c("control", "case"))
confMatrix(pred, truth = my.case, namePos = "case")

## on the scale of the linear predictors
pred2 <- predict(fit)
confMatrix(pred2, truth = infert$case, namePos = 1, cutoff = 0)

## only absolute numbers
confMatrix(pred, truth = infert$case, namePos = 1, relative = FALSE)
```

decisionStump

Compute Decision Stumps

Description

The function computes a decision stump for binary classification also known as 1-level decision tree or 1-rule.

Usage

```
decisionStump(pred, truth, namePos, perfMeasure = "YJS",
              MAX = TRUE, parallel = FALSE, ncores, delta = 0.01, ...)
```

Arguments

pred	numeric values that shall be used for classification; e.g. probabilities to belong to the positive group.
truth	true grouping vector or factor.
namePos	value representing the positive group.
perfMeasure	a single performance measure computed by function perfMeasure.
MAX	logical value. Whether to maximize or minimize the performacne measure.
parallel	logical value. If TRUE packages foreach and doParallel are used to parallelize the computations.

ncores integer value, number of cores that shall be used to parallelize the computations.
 delta numeric value for setting up grid for optimization; start is minimum of pred-delta, end is maximum of pred+delta.
 ... further arguments passed to function perfMeasures.

Details

The function is able to compute a decision stump for various performance measures, all performance measures that are implemented in function perfMeasures. Of course, for several of them the computation is not really useful such as sensitivity or specificity where one will get trivial decision rules.

Value

Object of class decisionStump.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References

W. Iba and P. Langley (1992). Induction of One-Level Decision Trees. In: *Machine Learning Proceedings 1992*, pages 233-240. URL: <https://doi.org/10.1016/B978-1-55860-247-2.50035-8>

R.C. Holte (1993). Very simple classification rules perform well on most commonly used datasets. In: *Machine Learning*, pages 63-91. URL: <https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.67.2711>

Examples

```
## example from dataset infert
fit <- glm(case ~ spontaneous+induced, data = infert, family = binomial())
pred <- predict(fit, type = "response")
res <- decisionStump(pred, truth = infert$case, namePos = 1)
predict(res, newdata = seq(from = 0, to = 1, by = 0.1))
```

HLgof.test

Hosmer-Lemeshow goodness of fit tests.

Description

The function computes Hosmer-Lemeshow goodness of fit tests for C and H statistic as well as the le Cessie-van Houwelingen-Copas-Hosmer unweighted sum of squares test for global goodness of fit.

Usage

```
HLgof.test(fit, obs, ngr = 10, X, verbose = FALSE)
```

Arguments

<code>fit</code>	numeric vector with fitted probabilities.
<code>obs</code>	numeric vector with observed values.
<code>ngr</code>	number of groups for C and H statistic.
<code>X</code>	covariate(s) for le Cessie-van Houwelingen-Copas-Hosmer global goodness of fit test.
<code>verbose</code>	logical, print intermediate results.

Details

Hosmer-Lemeshow goodness of fit tests are computed; see Lemeshow and Hosmer (1982).

If `X` is specified, the le Cessie-van Houwelingen-Copas-Hosmer unweighted sum of squares test for global goodness of fit is additionally determined; see Hosmer et al. (1997). A more general version of this test is implemented in function `residuals.lrm` in package `rms`.

Value

A list of test results.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References

S. Lemeshow and D.W. Hosmer (1982). A review of goodness of fit statistics for use in the development of logistic regression models. *American Journal of Epidemiology*, **115**(1), 92-106.

D.W. Hosmer, T. Hosmer, S. le Cessie, S. Lemeshow (1997). A comparison of goodness-of-fit tests for the logistic regression model. *Statistics in Medicine*, **16**, 965-980.

See Also

[residuals.lrm](#)

Examples

```
set.seed(111)
x1 <- factor(sample(1:3, 50, replace = TRUE))
x2 <- rnorm(50)
obs <- sample(c(0,1), 50, replace = TRUE)
fit <- glm(obs ~ x1+x2, family = binomial)
HLgof.test(fit = fitted(fit), obs = obs)
HLgof.test(fit = fitted(fit), obs = obs, X = model.matrix(obs ~ x1+x2))
```

`optCutoff`*Compute the Optimal Cutoff for Binary Classification*

Description

The function computes the optimal cutoff for various performance weasures for binary classification.

Usage

```
optCutoff(pred, truth, namePos, perfMeasure = "YJS",  
          MAX = TRUE, parallel = FALSE, ncores, delta = 0.01, ...)
```

Arguments

<code>pred</code>	numeric values that shall be used for classification; e.g. probabilities to belong to the positive group.
<code>truth</code>	true grouping vector or factor.
<code>namePos</code>	value representing the positive group.
<code>perfMeasure</code>	a single performance measure computed by function <code>perfMeasure</code> .
<code>MAX</code>	logical value. Whether to maximize or minimize the performacne measure.
<code>parallel</code>	logical value. If TRUE packages <code>foreach</code> and <code>doParallel</code> are used to parallelize the computations.
<code>ncores</code>	integer value, number of cores that shall be used to parallelize the computations.
<code>delta</code>	numeric value for setting up grid for optimization; start is minimum of <code>pred-delta</code> , end is maximum of <code>pred+delta</code> .
<code>...</code>	further arguments passed to function <code>perfMeasures</code> .

Details

The function is able to compute the optimal cutoff for various performance measures, all performance measures that are implemented in function `perfMeasures`. Of course, for several of them the computation is not really usefull such as sensitivity or specificity where one will get trivial cutoffs.

Value

Optimal cutoff and value of the optimized performance measure based on a simple grid search.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

Examples

```
## example from dataset infert
fit <- glm(case ~ spontaneous+induced, data = infert, family = binomial())
pred <- predict(fit, type = "response")
optCutoff(pred, truth = infert$case, namePos = 1)
```

or2rr

Transform OR to RR

Description

The function transforms a given odds-ratio (OR) to the respective relative risk (RR).

Usage

```
or2rr(or, p0, p1)
```

Arguments

or	numeric vector: OR (odds-ratio).
p0	numeric vector of length 1: incidence of the outcome of interest in the nonexposed group.
p1	numeric vector of length 1: incidence of the outcome of interest in the exposed group.

Details

The function transforms a given odds-ratio (OR) to the respective relative risk (RR). It can also be used to transform the limits of confidence intervals.

The formulas can be derived by combining the formulas for RR and OR; see also Zhang and Yu (1998).

Value

relative risk.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References

Zhang, J. and Yu, K. F. (1998). What's the relative risk? A method of correcting the odds ratio in cohort studies of common outcomes. *JAMA*, **280**(19):1690-1691.

Examples

```

## We use data from Zhang and Yu (1998)

## OR to RR using OR and p0
or2rr(14.1, 0.05)

## compute p1
or2rr(14.1, 0.05)*0.05

## OR to RR using OR and p1
or2rr(14.1, p1 = 0.426)

## OR and 95% confidence interval
or2rr(c(14.1, 7.8, 27.5), 0.05)

## Logistic OR and 95% confidence interval
logisticOR <- rbind(c(14.1, 7.8, 27.5),
                   c(8.7, 5.5, 14.3),
                   c(27.4, 17.2, 45.8),
                   c(4.5, 2.7, 7.8),
                   c(0.25, 0.17, 0.37),
                   c(0.09, 0.05, 0.14))
colnames(logisticOR) <- c("OR", "2.5%", "97.5%")
rownames(logisticOR) <- c("7.4", "4.2", "3.0", "2.0", "0.37", "0.14")
logisticOR

## p0
p0 <- c(0.05, 0.12, 0.32, 0.27, 0.40, 0.40)

## Compute corrected RR
## helper function
or2rr.mat <- function(or, p0){
  res <- matrix(NA, nrow = nrow(or), ncol = ncol(or))
  for(i in seq_len(nrow(or)))
    res[i,] <- or2rr(or[i,], p0[i])
  dimnames(res) <- dimnames(or)
  res
}
RR <- or2rr.mat(logisticOR, p0)
round(RR, 2)

## Results are not completely identical to Zhang and Yu (1998)
## what probably is caused by the fact that the logistic OR values
## provided in the table are rounded and are not exact values.

```

pairwise.auc

Compute pairwise AUCs

Description

The function computes pairwise AUCs.

Usage

```
pairwise.auc(x, g)
```

Arguments

x numeric vector.
g grouping vector or factor

Details

The function computes pairwise areas under the receiver operating characteristic curves (AUC under ROC curves) using function [AUC](#).

The implementation is in certain aspects analogously to [pairwise.t.test](#).

Value

Vector with pairwise AUCs.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

[AUC](#), [pairwise.t.test](#)

Examples

```
set.seed(13)
x <- rnorm(100)
g <- factor(sample(1:4, 100, replace = TRUE))
levels(g) <- c("a", "b", "c", "d")
pairwise.auc(x, g)
```

perfMeasures

Compute Performance Measures or Binary Classification

Description

The function computes various performance measures for binary classification.

Usage

```
perfMeasures(pred, pred.group, truth, namePos, cutoff = 0.5,
             weight = 0.5, wACC = weight, wLR = weight,
             wPV = weight, beta = 1, measures = "all")
```

Arguments

pred	numeric values that shall be used for classification; e.g. probabilities to belong to the positive group.
pred.group	vector or factor including the predicted group. If missing, pred.group is computed from pred, where $\text{pred} \geq \text{cutoff}$ is classified as positive.
truth	true grouping vector or factor.
namePos	value representing the positive group.
cutoff	cutoff value used for classification.
weight	weight used for computing weighted values. Must be in [0,1].
wACC	weight used for computing the weighted accuracy, where sensitivity is multiplied by wACC and specificity by $1 - \text{wACC}$. Must be in [0,1].
wLR	weight used for computing the weighted likelihood ratio, where PLR is multiplied by wLR and NLR by $1 - \text{wLR}$. Must be in [0,1].
wPV	weight used for computing the weighted predictive value, where PPV is multiplied by wPV and NPV by $1 - \text{wPV}$. Must be in [0,1].
beta	beta coefficient used for computing the F beta score. Must be nonnegative.
measures	character vector giving the measures that shall be computed; see details. Default "all" computes all measures available.

Details

The function `perfMeasures` can be used to compute various performance measures. For computing specific measures, the abbreviation given in parentheses have to be specified in argument `measures`. Single measures can also be computed by respective functions, where their names are identical to the abbreviations given in the parentheses.

The measures are: accuracy (ACC), probability of correct classification (PCC), fraction correct (FC), simple matching coefficient (SMC), Rand (similarity) index (RSI), probability of misclassification (PMC), error rate (ER), fraction incorrect (FIC), sensitivity (SENS), recall (REC), true positive rate (TPR), probability of detection (PD), hit rate (HR), specificity (SPEC), true negative rate (TNR), selectivity (SEL), detection rate (DR), false positive rate (FPR), fall-out (FO), false alarm (rate) (FAR), probability of false alarm (PFA), false negative rate (FNR), miss rate (MR), false discovery rate (FDR), false omission rate (FOR), prevalence (PREV), (positive) pre-test probability (PREP), (positive) pre-test odds (PREO), detection prevalence (DPREV), negative pre-test probability (NPREP), negative pre-test odds (NPREO), no information rate (NIR), weighted accuracy (WACC), balanced accuracy (BACC), (bookmaker) informedness (INF), Youden's J statistic (YJS), deltap' (DPp), positive likelihood ratio (PLR), negative likelihood ratio (NLR), weighted likelihood ratio (WLR), balanced likelihood ratio (BLR), diagnostic odds ratio (DOR), positive predictive value (PPV), precision (PREC), (positive) post-test probability (POSTP), (positive) post-test odds (POSTO), Bayes factor G1 (BFG1), negative predictive value (NPV), negative post-test probability (NPOSTP), negative post-test odds (NPOSTO), Bayes factor G0 (BFG0), markedness (MARK), deltap (DP), weighted predictive value (WPV), balanced predictive value (BPV), F1 score (F1S), Dice similarity coefficient (DSC), F beta score (FBS), Jaccard similarity coefficient (JSC), threat score (TS), critical success index (CSI), Matthews' correlation coefficient (MCC), Pearson's correlation (r phi) (RPHI), Phi coefficient (PHIC), Cramer's V (CRV), proportion of positive predictions (PPP), expected accuracy (EACC), Cohen's kappa coefficient (CKC), mutual information in bits

(MI2), joint entropy in bits (JE2), variation of information in bits (VI2), Jaccard distance (JD), information quality ratio (INFQR), uncertainty coefficient (UC), entropy coefficient (EC), proficiency (metric) (PROF), deficiency (metric) (DFM), redundancy (RED), symmetric uncertainty (SU), normalized uncertainty (NU)

These performance measures have in common that they require a dichotomization of the computed predictions (classification function). For measuring the performance without dichotomization one can apply function `perfScores`.

The prevalence is the prevalence given by the data. This often is not identical to the prevalence of the population. Hence, it might be better to compute PPV and NPV (and derived measures) by applying function `predValues`, where one can specify the assumed prevalence. This holds in general for all measures that depend on the prevalence.

Value

`data.frame` with names of the performance measures and their respective values.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References

- K.H. Brodersen, C.S. Ong, K.E. Stephan, J.M. Buhmann (2010). The balanced accuracy and its posterior distribution. In *Pattern Recognition (ICPR)*, 20th International Conference on, 3121-3124 (IEEE, 2010).
- J.A. Cohen (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* **20**, 3746.
- T. Fawcett (2006). An introduction to ROC analysis. *Pattern Recognition Letters* **27**, 861-874.
- T.A. Gerds, T. Cai, M. Schumacher (2008). The performance of risk prediction models. *Biom J* **50**, 457-479.
- D. Hand, R. Till (2001). A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning* **45**, 171-186.
- J. Hernandez-Orallo, P.A. Flach, C. Ferri (2012). A unified view of performance metrics: Translating threshold choice into expected classification loss. *J. Mach. Learn. Res.* **13**, 2813-2869.
- B.W. Matthews (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure* **405**, 442-451.
- D.M. Powers (2011). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness and Correlation. *Journal of Machine Learning Technologies* **1**, 37-63.
- N.A. Smits (2010). A note on Youden's J and its cost ratio. *BMC Medical Research Methodology* **10**, 89.
- B. Wallace, I. Dahabreh (2012). Class probability estimates are unreliable for imbalanced data (and how to fix them). In *Data Mining (ICDM)*, IEEE 12th International Conference on, 695-04.
- J.W. Youden (1950). Index for rating diagnostic tests. *Cancer* **3**, 32-35.

See Also

[confMatrix](#), [predValues](#), [perfScores](#)

Examples

```
## example from dataset infert
fit <- glm(case ~ spontaneous+induced, data = infert, family = binomial())
pred <- predict(fit, type = "response")

## with group numbers
perfMeasures(pred, truth = infert$case, namePos = 1)

## with group names
my.case <- factor(infert$case, labels = c("control", "case"))
perfMeasures(pred, truth = my.case, namePos = "case")

## on the scale of the linear predictors
pred2 <- predict(fit)
perfMeasures(pred2, truth = infert$case, namePos = 1, cutoff = 0)

## using weights
perfMeasures(pred, truth = infert$case, namePos = 1, weight = 0.3)

## selecting a subset of measures
perfMeasures(pred, truth = infert$case, namePos = 1,
             measures = c("SENS", "SPEC", "BACC", "YJS"))
```

perfScores

Compute Performance Scores for Binary Classification

Description

The function computes various performance scores for binary classification.

Usage

```
perfScores(pred, truth, namePos, wBS = 0.5, scores = "all", transform = FALSE)
```

Arguments

pred	numeric values that shall be used for classification; e.g. probabilities to belong to the positive group.
truth	true grouping vector or factor.
namePos	value representing the positive group.
wBS	weight used for computing the weighted Brier score (BS), where positive BS is multiplied by wBS and negative BS by 1-wBS. Must be in [0,1].
scores	character vector giving the scores that shall be computed; see details. Default "all" computes all scores available.

transform logical value indicating whether the values in pred should be transformed to [0,1]; see details.

Details

The function perfScores can be used to compute various performance scores. For computing specific scores, the abbreviation given in parentheses have to be specified in argument scores. Single scores can also be computed by respective functions, where their names are identical to the abbreviations given in the parentheses.

The available scores are: area under the ROC curve (AUC), Gini index (GINI), Brier score (BS), positive Brier score (PBS), negative Brier score (NBS), weighted Brier score (WBS), balanced Brier score (BBS).

If the predictions (pred) are not in the interval [0,1], the various Brier scores are not valid. By setting argument transform to TRUE, a simple logistic regression model is fit to the provided data and the predicted values are used for the computations.

Value

data.frame with names of the scores and their respective values.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References

- G.W. Brier (1950). Verification of forecasts expressed in terms of probability. *Mon. Wea. Rev.* **78**, 1-3.
- T. Fawcett (2006). An introduction to ROC analysis. *Pattern Recognition Letters* **27**, 861-874.
- T.A. Gerds, T. Cai, M. Schumacher (2008). The performance of risk prediction models. *Biom J* **50**, 457-479.
- D. Hand, R. Till (2001). A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning* **45**, 171-186.
- J. Hernandez-Orallo, P.A. Flach, C. Ferri (2011). Brier curves: a new cost- based visualisation of classifier performance. In L. Getoor and T. Scheffer (eds.) *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 585-592 (ACM, New York, NY, USA).
- J. Hernandez-Orallo, P.A. Flach, C. Ferri (2012). A unified view of performance metrics: Translating threshold choice into expected classification loss. *J. Mach. Learn. Res.* **13**, 2813-2869.
- B.W. Matthews (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure* **405**, 442-451.

See Also

[perfMeasures](#)

Examples

```
## example from dataset infert
fit <- glm(case ~ spontaneous+induced, data = infert, family = binomial())
pred <- predict(fit, type = "response")

## with group numbers
perfScores(pred, truth = infert$case, namePos = 1)

## with group names
my.case <- factor(infert$case, labels = c("control", "case"))
perfScores(pred, truth = my.case, namePos = "case")

## on the scale of the linear predictors
pred2 <- predict(fit)
perfScores(pred2, truth = infert$case, namePos = 1)

## using weights
perfScores(pred, truth = infert$case, namePos = 1, wBS = 0.3)
```

predValues

Compute PPV and NPV.

Description

The function computes the positive (PPV) and negative predictive value (NPV) given sensitivity, specificity and prevalence (pre-test probability).

Usage

```
predValues(sens, spec, prev)
```

Arguments

sens	numeric vector: sensitivities.
spec	numeric vector: specificities.
prev	numeric vector: prevalence.

Details

The function computes the positive (PPV) and negative predictive value (NPV) given sensitivity, specificity and prevalence (pre-test probability).

It's a simple application of the Bayes formula.

One can also specify vectors of length larger than 1 for sensitivity and specificity.

Value

Vector or matrix with PPV and NPV.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

Examples

```
## Example: HIV test
## 1. ELISA screening test (4th generation)
predValues(sens = 0.999, spec = 0.998, prev = 0.001)
## 2. Western-Plot confirmation test
predValues(sens = 0.998, spec = 0.999996, prev = 1/3)

## Example: connection between sensitivity, specificity and PPV
sens <- seq(0.6, 0.99, by = 0.01)
spec <- seq(0.6, 0.99, by = 0.01)
ppv <- function(sens, spec, pre) predValues(sens, spec, pre)[,1]
res <- outer(sens, spec, ppv, pre = 0.1)
image(sens, spec, res, col = terrain.colors(256), main = "PPV for prevalence = 10%",
      xlim = c(0.59, 1), ylim = c(0.59, 1))
contour(sens, spec, res, add = TRUE)
```

risks

Compute RR, OR and Other Risk Measures

Description

The function computes relative risk (RR), odds ration (OR), and several other risk measures; see details.

Usage

```
risks(p0, p1)
```

Arguments

p0	numeric vector of length 1: incidence of the outcome of interest in the nonexposed group.
p1	numeric vector of length 1: incidence of the outcome of interest in the exposed group.

Details

The function computes relative risk (RR), odds-ratio (OR), relative risk reduction (RRR) resp. relative risk increase (RRI), absolute risk reduction (ARR) resp. absolute risk increase (ARI), number needed to treat (NNT) resp. number needed to harm (NNH).

Value

Vector including several risk measures.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References

Porta, M. (2014). A Dictionary of Epidemiology. Oxford University Press. Retrieved 3 Oct. 2020, from <https://www.oxfordreference.com/view/10.1093/acref/9780199976720.001.0001/acref-9780199976720>

Examples

```
## See worked example in Wikipedia
risks(p0 = 0.4, p1 = 0.1)
risks(p0 = 0.4, p1 = 0.5)
```

rrCI

Compute Approximate Confidence Interval for RR.

Description

The function computes an approximate confidence interval for the relative risk (RR).

Usage

```
rrCI(a, b, c, d, conf.level = 0.95)
```

Arguments

a	integer: events in exposed group.
b	integer: non-events in exposed group.
c	integer: events in non-exposed group.
d	integer: non-events in non-exposed group.
conf.level	numeric: confidence level

Details

The function computes an approximate confidence interval for the relative risk (RR) based on the normal approximation; see Jewell (2004).

Value

A list with class "confint" containing the following components:

estimate	the estimated relative risk.
conf.int	a confidence interval for the relative risk.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References

Jewell, Nicholas P. (2004). Statistics for epidemiology. Chapman & Hall/CRC.

Relative risk. (2016, November 4). In Wikipedia, The Free Encyclopedia. Retrieved 19:58, November 4, 2016, from https://en.wikipedia.org/w/index.php?title=Relative_risk&oldid=747857409

Examples

```
## See worked example in Wikipedia
rrCI(a = 15, b = 135, c = 100, d = 150)
rrCI(a = 75, b = 75, c = 100, d = 150)
```

Index

* package

MKclass-package, 2

* univar

AUC, 3

AUC.test, 4

confMatrix, 5

decisionStump, 6

HLgof.test, 7

optCutoff, 9

or2rr, 10

pairwise.auc, 11

perfMeasures, 12

perfScores, 15

predValues, 17

risks, 18

rrCI, 19

ACC (perfMeasures), 12

AUC, 3, 4, 12

AUC.test, 4

BACC (perfMeasures), 12

BBS (perfScores), 15

BFG0 (perfMeasures), 12

BFG1 (perfMeasures), 12

BLR (perfMeasures), 12

BPV (perfMeasures), 12

BS (perfScores), 15

CKC (perfMeasures), 12

confMatrix, 5, 15

CRV (perfMeasures), 12

CSI (perfMeasures), 12

decisionStump, 6

DFM (perfMeasures), 12

DOR (perfMeasures), 12

DP (perfMeasures), 12

DPp (perfMeasures), 12

DPREV (perfMeasures), 12

DR (perfMeasures), 12

DSC (perfMeasures), 12

EACC (perfMeasures), 12

EC (perfMeasures), 12

ER (perfMeasures), 12

F1S (perfMeasures), 12

FAR (perfMeasures), 12

FBS (perfMeasures), 12

FC (perfMeasures), 12

FDR (perfMeasures), 12

FIC (perfMeasures), 12

FNR (perfMeasures), 12

FO (perfMeasures), 12

FOR (perfMeasures), 12

FPR (perfMeasures), 12

GINI (perfScores), 15

HLgof.test, 7

HR (perfMeasures), 12

INF (perfMeasures), 12

INFQR (perfMeasures), 12

JD (perfMeasures), 12

JE2 (perfMeasures), 12

JSC (perfMeasures), 12

MARK (perfMeasures), 12

MCC (perfMeasures), 12

MI2 (perfMeasures), 12

MKclass (MKclass-package), 2

MKclass-package, 2

MR (perfMeasures), 12

NBS (perfScores), 15

NIR (perfMeasures), 12

NLR (perfMeasures), 12

NPOST0 (perfMeasures), 12

NPOSTP (perfMeasures), 12
NPREO (perfMeasures), 12
NPREP (perfMeasures), 12
NPV (perfMeasures), 12
NU (perfMeasures), 12

optCutoff, 9
or2rr, 10

pairwise.auc, 11
pairwise.t.test, 12
PBS (perfScores), 15
PCC (perfMeasures), 12
PD (perfMeasures), 12
perfMeasures, 12, 16
perfScores, 15, 15
PFA (perfMeasures), 12
PHIC (perfMeasures), 12
PLR (perfMeasures), 12
PMC (perfMeasures), 12
POSTO (perfMeasures), 12
POSTP (perfMeasures), 12
PPP (perfMeasures), 12
PPV (perfMeasures), 12
PREC (perfMeasures), 12
predValues, 15, 17
PREO (perfMeasures), 12
PREP (perfMeasures), 12
PREV (perfMeasures), 12
PROF (perfMeasures), 12

REC (perfMeasures), 12
RED (perfMeasures), 12
residuals.lrm, 8
risks, 18
RPHI (perfMeasures), 12
rrCI, 19
RSI (perfMeasures), 12

SEL (perfMeasures), 12
SENS (perfMeasures), 12
SMC (perfMeasures), 12
SPEC (perfMeasures), 12
SU (perfMeasures), 12

TNR (perfMeasures), 12
TPR (perfMeasures), 12
TS (perfMeasures), 12

UC (perfMeasures), 12

VI2 (perfMeasures), 12

WACC (perfMeasures), 12
WBS (perfScores), 15
wilcox.test, 4
WLR (perfMeasures), 12
WPV (perfMeasures), 12

YJS (perfMeasures), 12