Package 'DTComPair'

January 20, 2025

Type Package

Title Comparison of Binary Diagnostic Tests in a Paired Study Design

Version 1.2.6

Description Comparison of the accuracy of two binary diagnostic tests in a ``paired'' study design, i.e. when each test is applied to each subject in the study.

License GPL (>= 3)

URL <https://github.com/chstock/DTComPair>,
<https://chstock.github.io/DTComPair/>

BugReports <https://github.com/chstock/DTComPair/issues>

Encoding UTF-8

LazyData true

Imports gee, ellipse, stats, assertthat

Depends R (>= 3.5.0), PropCIs

Suggests knitr, rmarkdown, spelling

RoxygenNote 7.3.1

Language en-US

NeedsCompilation no

Author Christian Stock [aut, cre] (<<https://orcid.org/0000-0002-3493-3234>>),
Thomas Hielscher [aut],
Andrea Discacciati [aut]

Maintainer Christian Stock <christian.stock@boehringer-ingelheim.com>

Repository CRAN

Date/Publication 2024-09-24 15:50:02 UTC

Contents

DTComPair-package	*Comparison of Binary Diagnostic Tests in a Paired Study Design*

Description

Comparison of the accuracy of two binary diagnostic tests in a "paired" study design, i.e. when each test is applied to each subject in the study.

Details

The accuracy measures that can be compared in the present version are sensitivity, specificity, positive and negative predictive values, and positive and negative diagnostic likelihood ratios.

It is required that results from a binary gold-standard test are also available.

Methods for comparison of sensitivity and specificity: McNemar test (McNemar, 1947) and exact binomial test. Further, several methods to compute confidence intervals for differences in sensitivity and specificity are implemented.

Methods for comparison of positive and negative predictive values: generalized score statistic (Leisenring, Alonzo and Pepe, 2000), weighted generalized score statistic (Kosinski, 2013) and comparison of relative predictive values (Moskowitz and Pepe, 2006).

Methods for comparison of positive and negative diagnostic likelihood ratios: a regression model approach (Gu and Pepe, 2009).

For a general introduction into the evaluation of diagnostic tests see e.g. Pepe (2003), Zhou, Obuchowski and McClish (2011).

Author(s)

Christian Stock, Thomas Hielscher and Andrea Discacciati

Maintainer: Christian Stock <christian.stock@boehringer-ingelheim.com>

References

Gu and Pepe (2009), "Estimating the capacity for improvement in risk prediction with a marker", <doi:10.1093/biostatistics/kxn025>.

Kosinski (2013), "A weighted generalized score statistic for comparison of predictive values of diagnostic tests", <doi:10.1002/sim.5587>.

Leisenring, Alonzo and Pepe (2000), "Comparisons of predictive values of binary medical diagnostic tests for paired designs", <doi:10.1111/j.0006-341X.2000.00345.x>.

McNemar (1947), "Note on the sampling error of the difference between correlated proportions or percentages", <doi:10.1007/BF02295996>.

Moskowitz and Pepe (2006), "Comparing the predictive values of diagnostic tests: sample size and analysis for paired study designs", <doi:10.1191/1740774506cn147oa>.

Pepe (2003, ISBN:978-0198509844), "The statistical evaluation of medical tests for classification and prediction".

Zhou, Obuchowski and McClish (2011), "Statistical Methods in Diagnostic Medicine", <doi:10.1002/9780470906514>.

See Also

Data management functions: `tab.1test`, `tab.paired`, `read.tab.paired`, `generate.paired` and `represent.long`.

Computation of standard accuracy measures for a single test: `acc.1test` and `acc.paired`.

Comparison of sensitivity and specificity: `sesp.mcnemar`, `sesp.exactbinom` and `sesp.diff.ci`.

Comparison of positive and negative predictive values: `pv.gs`, `pv.wgs` and `pv.rpv`.

Comparison of positive and negative diagnostic likelihood ratios: `dlr.regtest` and `DLR`.

Examples

```
data(Paired1) # Hypothetical study data
hsd <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
acc.paired(hsd)
sesp.mcnemar(hsd)
pv.rpv(hsd)
dlr.regtest(hsd)
```

| acc.1test | *Accuracy of a Single Binary Diagnostic Test* |

Description

Sensitivity and specificity, (positive and negative) predictive values and (positive and negative) diagnostic likelihood ratios of a single binary diagnostic test.

Usage

```
acc.1test(tab, alpha, testname, method.ci, ...)
```

Arguments

tab	An object of class `tab.1test`.
alpha	Significance level alpha for 100(1-alpha)%-confidence intervals, the default is 0.05.
testname	A character string containing the name of the diagnostic test.
method.ci	A character string with the name of the function to compute the confidence intervals for sensitivity, specificity, and predictive values. The default is ''waldci'' for a Wald asymptotic normal-based confidence intervals. See 'Details'.
...	Additional arguments (usually not required).

Details

The calculation of accuracy measures follows standard methodology, e.g. described in Pepe (2003) or Zhou et al. (2011).

The confidence intervals for sensitivity, specificity, and predictive values are computed using the methodology implemented in the function passed to the argument `method.ci`.

Available options are:

1. waldci - Wald asymptotic normal-based confidence interval; the default,
2. logitci - asymptotic normal-based confidence interval on the logit scale and then back-transformed,
3. exactci - Clopper-Pearson exact confidence interval,
4. add4cici - Agresti-Coull add-4 confidence interval,
5. addz2ci - Agresti-Coull add-z^2/2 confidence interval,
6. blakerci - Blaker exact confidence interval,
7. scoreci - Wilson score confidence interval,
8. midPci - mid-P confidence interval.

Options (3) to (8) are based on the homonymous functions from the {PropCIs} package. Please see the respective package documentation for more details.

Confidence intervals for diagnostic likelihood ratios are computed according to Simel et al. (1991).

Value

A list of class `acc.1test`:

tab   A contingency table (matrix) of test results; the same `tab` that is supplied as an argument.

	Diseased	Non-diseased	Total
Test pos.	...	...	...
Test neg.	...	...	...
Total	...	...	...

sensitivity   A numeric vector containing the estimated sensitivity (est), its standard error (se), lower confidence limit (lcl) and upper confidence limit (ucl).

specificity   A numeric vector containing the estimated specificity (est), its standard error (se), lower confidence limit (lcl) and upper confidence limit (ucl).

ppv   A numeric vector containing the estimated positive predictive value (est), its standard error (se), lower confidence limit (lcl) and upper confidence limit (ucl).

npv   A numeric vector containing the estimated negative predictive value (est), its standard error (se), lower confidence limit (lcl) and upper confidence limit (ucl).

pdlr   A numeric vector containing the estimated positive diagnostic likelihood ratio (est), the standard error of the logarithm of the positive diagnostic likelihood ratio (se.ln), the lower confidence limit (lcl) and the upper confidence limit (ucl).

ndlr   A numeric vector containing the estimated negative diagnostic likelihood ratio (est), the standard error of the logarithm of the negative diagnostic likelihood ratio (se.ln), the lower confidence limit (lcl) and the upper confidence limit (ucl).

alpha   The significance level alpha used to compute 100(1-alpha)%-confidence intervals, the default is 0.05.

testname   A character string containing the name of the diagnostic test.

method.ci   A character string describing the method used to compute the confidence intervals for sensitivity, specificity, and predictive values.

References

Pepe, M. (2003). The statistical evaluation of medical tests for classification and prediction. Oxford Statistical Science Series. Oxford University Press, 1st edition.

Simel, D.L., Samsa, G.P., Matchar, D.B. (1991). Likelihood ratios with confidence: sample size estimation for diagnostic test studies. *J Clin Epidemiol*, 44(8):763-70.

Zhou, X., Obuchowski, N., and McClish, D. (2011). Statistical Methods in Diagnostic Medicine. Wiley Series in Probability and Statistics. John Wiley & Sons, Hoboken, New Jersey, 2nd edition.

See Also

`tab.1test`, `print.acc.1test`, `acc.paired`.

Examples

```
data(Paired1) # Hypothetical study data
a1 <- tab.1test(d=d, y=y1, data=Paired1)
a2 <- acc.1test(a1)
print(a2)
a3 <- acc.1test(a1, method="exactci", conf.level=0.99)
print(a3)
```

---

acc.paired   *Accuracy of Two Binary Diagnostic Tests in a Paired Study Design*

Description

Sensitivity and specificity, (positive and negative) predictive values and (positive and negative) diagnostic likelihood ratios of a two binary diagnostic tests in a paired study design.

Usage

```
acc.paired(tab, alpha, method.ci, ...)
```

Arguments

tab	An object of class tab.paired.
alpha	Significance level alpha for 100(1-alpha)%-confidence intervals, the default is 0.05.
method.ci	A function used to compute the confidence intervals for sensitivity, specificity, and predictive values. The default is waldci for Wald's asymptotic normal-based confidence intervals. See acc.1test.
...	Additional arguments, usually not required.

Details

The calculation of accuracy measures follows standard methodology, e.g. described in Pepe (2003) or Zhou et al. (2011).

The confidence intervals for sensitivity, specificity, and predictive values are computed using the methodology implemented in the function passed to the argument method.ci.

Confidence intervals for diagnostic likelihood ratios are computed according to Simel et al. (1991).

Value

A list of class acc.paired:

Test1	A list of class acc.1test containing results and accuracy estimates of Test 1.
Test2	A list of class acc.1test containing results and accuracy estimates of Test 2.

References

Pepe, M. (2003). The statistical evaluation of medical tests for classifcation and prediction. Oxford Statistical Science Series. Oxford University Press, 1st edition.

Zhou, X., Obuchowski, N., and McClish, D. (2011). Statistical Methods in Diagnostic Medicine. Wiley Series in Probability and Statistics. John Wiley & Sons, Hoboken, New Jersey, 2nd edition.

See Also

`tab.paired`, `print.acc.paired`, `acc.1test`.

Examples

```
data(Paired1) # Hypothetical study data
b1 <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
b2 <- acc.paired(b1)
print(b2)
```

| DLR | *Estimating the Capacity for Improvement in Diagnostic Risk Prediction with an additional marker based on the Diagnostic Likelihood Ratio (DLR)* |

Description

This function allows for estimating the log diagnostic likelihood ratio in a regression model approach. It can be used to assess the gain in diagnostic accuracy for a new binary or continuous diagnostic marker compared to established markers, to determine the impact of covariates on the risk prediction model, and to estimate the DLR for selected marker/covariate values.

Usage

```
DLR(basemodel, augmentedmodel, diseasestatus, dataset, clustervar = NULL, alpha=0.05)
```

Arguments

basemodel	pre-test/base model X, formula character string
augmentedmodel	post-test/ augmented model V, formula character string, this is usually the basemodel X including the additional diagnostic test of interest Y and interactions XY
diseasestatus	variable name containing disease status, assumed to be a 0/1 variable, for having condition of interest (1) or not (0), character string
dataset	dataframe, needs to be in wide format with one observation per subject
clustervar	optional, cluster variable name in dataset, character string
alpha	significance level alpha used for confidence intervals, the default is 0.05.

Details

This function is an implementation of the algorithm described in the appendix of Gu and Pepe (2009) using the GEE approach in order to get standard error estimates. The definition of I and Zero matrices is slightly more flexible than the ones described in section 3 in order to allow for models without interaction.

Value

Returns a list including

logPreTestModel

        logistic regression model output for prior disease using base model X: P(D=1|X). All estimates are on a log scale.

logPostTestModel

        logistic regression model output for posterior disease using augmented model V: P(D=1|X,Y),i.e. P(D=1|V). All estimates are on a log scale.

logDLRModel   regression model output for log DLR defined as difference between logPostTestModel and logPreTestModel. All estimates are on a log scale.

DLR   Positive/negative DLR for diagnostic marker Y, with all base covariates X set to 1. Results are only sensible for binary marker Y taking values 0/1.

Author(s)

Thomas Hielscher (t.hielscher@dkfz.de)

References

Gu, W. and Pepe, M. S. (2009). Estimating the capacity for improvement in risk prediction with a marker. *Biostatistics*, 10(1):172-86.

See Also

dlr.regtest

Examples

```
library(DTComPair)
data(Paired1)

# test y1 conditioned on null model: DLR+(Y1=1) and DLR-(Y1=0)

DLR("~ 1","~ y1","d",Paired1)

# test y1 conditioned on test y2 with interaction, DLR+(Y1=1|Y2=1) and DLR-(Y1=0|Y2=1)

DLR("~ y2","~ y2 * y1","d",Paired1)
```

---

dlr.regtest *Differences in Diagnostic Likelihood Ratios*

Description

Performs a test for differences in (positive and negative) diagnostic likelihood ratios (DLRs) of two binary diagnostic tests in a paired study using a regression model approach proposed by Gu and Pepe (2009).

Usage

```
dlr.regtest(tab, alpha)
```

Arguments

tab   An object of class `tab.paired`.

alpha   Significance level alpha for 100(1-alpha)%-confidence intervals, the default is 0.05.

Details

The null hypothesis rDLR = DLR of Test 1 / DLR of Test 2 = 1 is tested with respect to both positive and negative DLRs of the two diagnostic tests.

This function calls `DLR`, a general implementation of the method proposed by Gu and Pepe (2009).

Value

A list containing

pdlr   A list with `test1` (the positive DLR of test 1), `test2` (the positive DLR of test 2), `ratio` (the ratio of positive DLRs, computed as `test1/test2`, `se.log` (the standard error of the logarithm of `ratio`), the `test.statistic` and the corresponding `p.value`.

ndlr   A list with `test1` (the negative DLR of test 1), `test2` (the negative DLR of test 2), `ratio` (the ratio of negative DLRs, computed as `test1/test2`, `se.log` (the standard error of the logarithm of `ratio`), the `test.statistic` and the corresponding `p.value`.

alpha   The significance level alpha used to compute 100(1-alpha)%-confidence intervals for the `ratio` of positive and negative DLRs, the default is 0.05.

method   The name of the method used to compare the positive and negative DLRs, here "diagnostic likelihood regression model (regtest)".

References

Gu, W. and Pepe, M. S. (2009). Estimating the capacity for improvement in risk prediction with a marker. *Biostatistics*, 10(1):172-86.

See Also

DLR

Examples

```
data(Paired1) # Hypothetical study data
ptab <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
ptab
dlr.results <- dlr.regtest(ptab)
str(dlr.results)
dlr.results
```

---

ellipse.pv.rpv   *Elliptical joint confidence region for relative positive and negative predictive value*

Description

Returns a 100(1-alpha)% elliptical joint confidence region for the parameter vector {log(relative positive predictive value), log(relative negative predictive value)}.

Usage

```
ellipse.pv.rpv(x, alpha = 0.05, npoints = 100, exponentiate = FALSE)
```

Arguments

x	an object returned by the pv.rpv function.
alpha	significance level alpha used to compute the 100(1-alpha)% region. The default is 0.05, for a 95% region.
npoints	the number of points used in the ellipse. Default is 100.
exponentiate	a logical value indicating whether or not to exponentiate the values for the centre of the ellipse and for the the ellipsoidal outline. Defaults to FALSE.

Value

A list containing:

centre	the centre of the ellipse.
ellipse	an npoints x 2 matrix with the x and y coordinates for the ellipsoidal outline. Suitable for plot-ing.

References

Moskowitz, C.S., and Pepe, M.S. (2006). Comparing the predictive values of diagnostic tests: sample size and analysis for paired study designs. *Clin Trials*, 3(3):272-9.

See Also

pv.rpv and ellipse::ellipse.

Examples

```
data(Paired1) # Hypothetical study data
ftable(Paired1)
paired.layout <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
paired.layout
rpv.results <- pv.rpv(paired.layout)
ellipse.data <- ellipse.pv.rpv(rpv.results)
if(interactive()){
  plot(ellipse.data$ellipse, type = "l", ylim = c(-0.4, 0.2), xlim = c(-0.2, 0.2))
  points(ellipse.data$centre[1], ellipse.data$centre[2], col = "red", pch = 19)
  abline(h = 0, v = 0, lty = 3)
}
```

---

generate.paired   *Generate Dataset from "tab.paired"-Object*

Description

Generates a dataset from contingency tables of binary diagnostic test results in a paired study design.

Usage

```
generate.paired(tab, ...)
```

Arguments

tab	An object of class tab.paired.
...	Additional arguments (usually not required).

Value

A dataframe containing:

d	A numeric vector specifying the gold-standard results (1 = presence of disease, 0 = absence of disease).
y1	A numeric vector specifying the results of diagnostic test 1 (1 = positive, 0 = negative).
y2	A numeric vector specifying the results of diagnostic test 2 (1 = positive, 0 = negative).

See Also

tab.paired and read.tab.paired.

Examples

```
data(Paired1) # Hypothetical study data
ftable(Paired1)
paired.layout <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
new.df <- generate.paired(paired.layout)
head(new.df)
ftable(new.df)
```

Paired1	*DTComPair-dataset 1*

Description

Hypothetical data from a paired study that aims to compare the accuracy of two binary diagnostic tests.

Usage

```
Paired1
```

Format

A dataframe containing 3 columns (d, y1 and y2) and 712 rows.

d is a numeric vector specifying the gold-standard results (1 = presence of disease, 0 = absence of disease).

y1 is a numeric vector specifying the results of diagnostic test 1 (1 = positive, 0 = negative).

y2 is a numeric vector specifying the results of diagnostic test 1 (1 = positive, 0 = negative).

Examples

```
data(Paired1) # Hypothetical study data
ftable(Paired1)
```

---

print.acc.1test	*Print "acc.1test"-Object*

Description

Prints objects of class acc.1test in an easy-to-read form (S3method).

Usage

```
## S3 method for class 'acc.1test'
print(x, ...)
```

Arguments

x	An object of class `acc.1test`.
...	Additional arguments (usually not required).

Value

Creates a list object from parts of its input that is then printed in a tabular layout.

See Also

`acc.1test`.

Examples

```
data(Paired1) # Hypothetical study data
a1 <- tab.1test(d=d, y=y1, data=Paired1)
a2 <- acc.1test(a1)
print(a2)
```

---

print.acc.paired	*Print "acc.paired"-Object*

Description

Prints objects of class `acc.paired` in an easy-to-read form (S3method).

Usage

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

Arguments

x	An object of class `acc.paired`.
...	Additional arguments (usually not required).

Value

Creates a list object from parts of its input that is then printed in a tabular layout.

See Also

`acc.paired`.

Examples

```
data(Paired1) # Hypothetical study data
b1 <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
b2 <- acc.paired(b1)
print(b2)
```

---

print.tab.1test   *Print "tab.1test"-Object*

Description

Prints objects of class `tab.1test` in an easy-to-read form (S3method).

Usage

```
## S3 method for class 'tab.1test'
print(x, ...)
```

Arguments

x   An object of class `tab.1test`.

...   Additional arguments (usually not required).

Value

Creates a list object from parts of its input that is then printed in a tabular layout.

See Also

`tab.1test`, `acc.1test`, `tab.paired`.

Examples

```
data(Paired1) # Hypothetical study data
a <- tab.1test(d=d, y=y1, data=Paired1)
print(a)
```

---

print.tab.paired *Print "tab.paired"-Object*

Description

Prints objects of class `tab.paired` in an easy-to-read form (S3method).

Usage

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

Arguments

x	An object of class `tab.paired`.
...	Additional arguments (usually not required).

Value

Creates a list object from parts of its input that is then printed in a tabular layout.

See Also

`tab.paired`, `acc.paired`, `tab.1test`.

Examples

```
data(Paired1) # Hypothetical study data
b <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
print(b)
```

---

pv.gs *Generalized Score Statistic for Comparison of Predictive Values*

Description

Performs a test for differences in (positive and negative) predictive values of two binary diagnostic tests using a generalized score statistic proposed by Leisenring, Alonzo and Pepe (2000).

Usage

```
pv.gs(tab)
```

Arguments

tab	An object of class `tab.paired`.

Value

A list containing:

ppv   A list with `test1` (the positive predictive value of test 1), `test2` (the positive pre-
             dictive value of test 2), `diff` (the difference in positive predictive values, com-
             puted as `test2` - `test1`), the `test.statistic` and the corresponding `p.value`.

npv   A list with `test1` (the negative predictive value of test 1), `test2` (the nega-
             tive predictive value of test 2), `diff` (the difference in negative predictive val-
             ues, computed as `test2` - `test1`), the `test.statistic` and the corresponding
             `p.value`.

method   The name of the method used to compare predictive values, here "generalized
             score statistic (gs)".

References

Leisenring, W., Alonzo, T., and Pepe, M. S. (2000). Comparisons of predictive values of binary
medical diagnostic tests for paired designs. *Biometrics*, 56(2):345-51.

See Also

`pv.wgs` and `pv.rpv`.

Examples

```
data(Paired1) # Hypothetical study data
ftable(Paired1)
paired.layout <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
paired.layout
gs.results <- pv.gs(paired.layout)
str(gs.results)
gs.results
gs.results$ppv["p.value"]
```

---

pv.prev   *Compute predictive values for theoretical prevalences*

Description

It is often of interest to estimate predictive values assuming the test were applied to a population
with a different prevalence of the disease. Projected predictive values may be calculated using
Bayes theorem and the relation between predictive values and diagnostic likelihood ratios can be
used to derived corresponding confidence intervals.

Usage

```
pv.prev(pi, acc)
```

Arguments

pi	A theoretical prevalence of the disease (proportion).
acc	An object of class 'acc.1test'.

Details

Predictive values, assuming a certain prevalence of the disease, are derived using the relation between predictive values and diagnostic likelihood ratios:

- PPV = 1 / (1 + (1 / pi - 1) / pDLR) - NPV = 1 / (1 + (1 / (1 / pi - 1)) / nDLR).

See Newcombe RG (2013). Confidence Intervals for Proportions and Related Measures of Effect Size. Chapman and Hall/ CRC Biostatistics Series (chapters 12.3+5 and 14.9).

The alpha-level of (1-alpha)-confidence intervals is inherited from 'acc.1test'.

Value

A vector containing the projected values.

See Also

[acc.1test()]

Examples

```
data(Paired1) # Hypothetical study data
a1 <- tab.1test(d=d, y=y1, data=Paired1)
a2 <- acc.1test(a1, alpha = 0.05)
pv.prev(pi=0.2, acc=a2)
pv.prev(pi=0.5, acc=a2)
```

---

pv.rpv *Comparison of Predictive Values using Relative Predictive Values*

Description

Performs a test for differences in (positive and negative) predictive values of two binary diagnostic tests in a paired study design using relative predictive values, as proposed by Moskowitz and Pepe (2006).

Usage

```
pv.rpv(tab, alpha)
```

Arguments

tab	An object of class `tab.paired`.
alpha	Significance level alpha used to compute 100(1-alpha)%-confidence intervals, the default is 0.05.

Value

A list containing:

ppv	named vector containing `test1` (the positive predictive value of test 1), `test2` (the positive predictive value of test 2), `rppv` (the relative positive predictive value, computed as `test2 / test1`), `se.log.rppv` (the standard error of the logarithm of `rppv`), `lcl.rppv` (lower confidence limit of `rppv`), `ucl.rppv` (the upper confidence limit of `rppv`), the `test.statistic` and the corresponding `p.value`.
npv	named vector containing `test1` (the negative predictive value of test 1), `test2` (the negative predictive value of test 2), `rnpv` (the relative negative predictive value, computed as `test2 / test1`), `se.log.rnpv` (the standard error of the logarithm of `rnpv`), `lcl.rnpv` (the lower confidence limit of `rnpv`), `ucl.rnpv` (the upper confidence limit of `rnpv`), the `test.statistic` and the corresponding `p.value`.
Sigma	Estimated variance-covariance matrix for {log(relative positive predictive value), log(relative negative predictive value)}.
method	Name of the method used to compare predictive values, here "relative predictive values (rpv)".
alpha	Significance level alpha used to compute 100(1-alpha)%-confidence intervals for `rppv` and `rnpv`, the default is 0.05.

References

Moskowitz, C.S., and Pepe, M.S. (2006). Comparing the predictive values of diagnostic tests: sample size and analysis for paired study designs. *Clin Trials*, 3(3):272-9.

See Also

`pv.gs` and `pv.wgs`.

Examples

```
data(Paired1) # Hypothetical study data
ftable(Paired1)
paired.layout <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
paired.layout
rpv.results <- pv.rpv(paired.layout)
str(rpv.results)
rpv.results
rpv.results$ppv["p.value"]
```

| pv.wgs | *Weighted Generalized Score Statistic for Comparison of Predictive Values* |

Description

Performs a test for differences in (positive and negative) predictive values of two binary diagnostic tests using a weighted generalized score statistic proposed by Kosinski (2013).

Usage

```
pv.wgs(tab)
```

Arguments

tab   An object of class `tab.paired`.

Value

A list containing:

ppv   A list with `test1` (the positive predictive value of test 1), `test2` (the positive predictive value of test 2), `diff` (the difference in positive predictive values, computed as `test2 - test1`, the `test.statistic` and the corresponding `p.value`.

npv   A list with `test1` (the negative predictive value of test 1), `test2` (the negative predictive value of test 2), `diff` (the difference in negative predictive values, computed as `test2 - test1`, the `test.statistic` and the corresponding `p.value`.

method   The name of the method used to compare predictive values, here "weighted generalized score statistic (wgs)".

References

Kosinski, A.S. (2013). A weighted generalized score statistic for comparison of predictive values of diagnostic tests. *Stat Med*, 32(6):964-77.

See Also

`pv.gs` and `pv.rpv`.

Examples

```
data(Paired1) # Hypothetical study data
ftable(Paired1)
paired.layout <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
paired.layout
wgs.results <- pv.wgs(paired.layout)
str(wgs.results)
```

```
wgs.results
wgs.results$ppv["p.value"]
```

---

read.tab.1test   *Read in "tab.1test"-Objects*

Description

Reads in objects of class `tab.1test` using cell frequencies.

Usage

```
read.tab.1test(a, b, c, d, testname, ...)
```

Arguments

a	The number of diseased subjects with a positive test.
b	The number of non-diseased subjects with a positive test.
c	The number of diseased subjects with a negative test.
d	The number of non-diseased subjects with a negative test.
testname	An optional vector specifying the name of the diagnostic test, e.g. c("Test A"). If not supplied, the corresponding variable name is used as testname.
...	Additional arguments (usually not required).

Value

Returns a list of class `tab.1test` containing:

tab.1test   A contingency table (matrix) of test results.

	Diseased	Non-diseased	Total
Test pos.	a	b	a+b
Test neg.	c	d	c+d
Total	a+c	b+d	a+b+c+d

testname   The name of the diagnostic test.

Note

Objects of class `tab.1test` are required as arguments for `acc.1test`, a function to compute the accuracy of a binary diagnostic test.

See Also

`tab.1test`, `print.tab.1test`, `acc.1test`.

Examples

```
read.t1 <- read.tab.1test(321, 51, 730, 272, testname="Test1")
class(read.t1)
read.t1
acc.1test(read.t1)
```

---

read.tab.paired	*Read in "tab.paired"-Objects*

Description

Reads in objects of class `tab.paired` using `cell frequencies`.

Usage

```
read.tab.paired(d.a, d.b, d.c, d.d, nd.a, nd.b, nd.c, nd.d, testnames, ...)
```

Arguments

`d.a`	The number of diseased subjects with a positive test 1 and a positive test 2.
`d.b`	The number of diseased subjects with a negative test 1 and a positive test 2.
`d.c`	The number of diseased subjects with a positive test 1 and a negative test 2.
`d.d`	The number of diseased subjects with a negative test 1 and a negative test 2.
`nd.a`	The number of non-diseased subjects with a positive test 1 and a positive test 2.
`nd.b`	The number of non-diseased subjects with a negative test 1 and a positive test 2.
`nd.c`	The number of non-diseased subjects with a positive test 1 and a negative test 2.
`nd.d`	The number of non-diseased subjects with a negative test 1 and a negative test 2.
`testnames`	An optional vector specifying the names of diagnostic test 1 and diagnostic test 2, e.g. `c("Test A","Test B")`. If not supplied, the variable names are used as testnames.
`...`	Additional arguments (usually not required).

Value

Returns a list of class `tab.paired` containing:

diseased   A contingency table (matrix) of test results among *diseased* subjects.

	Test1 pos.	Test1 neg.	Total
Test 2 pos.	d.a	d.b	d.a+d.b
Test 2 neg.	d.c	d.d	d.c+d.d
Total	d.a+d.c	d.b+d.d	d.a+d.b+d.c+d.d

non.diseased   A contingency table (matrix) of test results among *non-diseased* subjects.

	Test1 pos.	Test1 neg.	Total
Test 2 pos.	nd.a	nd.b	nd.a+nd.b
Test 2 neg.	nd.c	nd.d	nd.c+nd.d
Total	nd.a+nd.c	nd.b+nd.d	nd.a+nd.b+nd.c+nd.d

testnames   The names of the diagnostic tests.

Note

Objects of class `tab.paired` are essential arguments for various functions in the `DTComPair`-package.

See Also

`tab.paired`, `print.tab.paired`, `acc.paired`, `generate.paired`.

Examples

```
read.t2 <- read.tab.paired(321, 51, 730, 272,
                           120, 8, 74, 109,
                           testnames=c("Test A", "Test B"))
class(read.t2)
read.t2
acc.paired(read.t2)
```

---

represent.long	*Long Representation of Results from Two Binary Diagnostic Tests*

Description

Long representation of results from two binary diagnostic tests.

Usage

```
represent.long(d, y1, y2)
```

Arguments

d	A numeric vector specifying the gold-standard results (1 = presence of disease, 0 = absence of disease).
y1	A numeric vector specifying the results of diagnostic test 1 (1 = positive, 0 = negative).
y2	An numeric vector specifying the results of diagnostic test 2 (1 = positive, 0 = negative).

Details

Sometimes a long representation of data from a "paired" study of binary diagnostic tests is required, e.g. to run regression analyses.

In a *wide* representation each subject has 1 record in the dataset containing d, y1 and y2.

In a *long* representation each subjects has 2 records in the dataset, one for each test. The data format is shown below.

Value

A dataframe containing:

id	A numeric vector specifying the patient identifier.
d	A numeric vector specifying the gold-standard results (1 = presence of disease, 0 = absence of disease).
x	A numeric vector specifying the diagnostic test (1 = test 1, 0 = test 2).
y	A numeric vector specifying the test results (1 = positive, 0 = negative).

See Also

`tab.paired` and `read.tab.paired`.

Examples

```
data(Paired1) # Hypothetical study data
names(Paired1)
new.long <- represent.long(d=Paired1$d, y1=Paired1$y1, y2=Paired1$y2)
str(new.long)
head(new.long)
```

---

sesp.diff.ci	*Confidence Intervals for Differences in Sensitivity and Specificity*

Description

Calculates confidence intervals for differences in sensitivity and specificity of two binary diagnostic tests in a paired study design.

Usage

```
sesp.diff.ci(tab, ci.method, alpha, cont.corr)
```

Arguments

tab	An object of class `tab.paired`.
ci.method	The available methods are "`wald`" (Altman, 1991), "`agresti-min`" (Agresti and Min, 2005), "`bonett-price`" (Bonett and Price, 2011), and "`tango`" (Tango, 1998), the default is "`wald`".
alpha	Significance level alpha for 100(1-alpha)%-confidence intervals for the difference in sensitivity and specificity, the default is 0.05.
cont.corr	A logical value indicating whether the continuity correction should be used (only available for `ci.method="wald"`), the default is `FALSE`.

Details

For details and recommendations see Newcombe (2012) and Wenzel and Zapf (2013).

Value

A list containing:

sensitivity	A vector containing `test1` (the sensitivity of test 1), `test2` (the specificity of test 2), `diff` (the difference between the two sensitivities, computed as `test2 - test1`), `diff.se` (the standard error of `diff`), `diff.lcl` (the lower confidence limit of `diff`) and `diff.ucl` (the upper confidence limit of `diff`).
specificity	A vector containing `test1` (specificity of test 1), `test2` (specificity of test 2), `diff` (the difference between the two specificities, computed as `test2 - test1`), `diff.se` (the standard error of `diff`), `diff.lcl` (the lower confidence limit of `diff`) and `diff.ucl` (the upper confidence limit of `diff`).
ci.method	The name of the method used to calculate confidence intervals.
alpha	The level alpha used to compute 100(1-alpha)%-confidence intervals.
cont.corr	A logical value indicating whether the continuity correction was applied.

References

Altman, D.G. (1991). Practical statistics for medical research. Chapman & Hall, London.

Agresti, A. and Min, Y. (2005). Simple improved confidence intervals for comparing matched proportions. *Stat Med*, 24(5): 729-40.

Bonett, D.G., and Price, R.M. (2011). Adjusted Wald confidence intervals for a difference of binomial proportions based on paired data. *J Educ Behav Stat*, 37(4): 479-488.

Newcombe R.G. (2012). Confidence intervals for proportions and related measures of effect size. Chapman and Hall/CRC Biostatistics Series.

Tango, T. (1998). Equivalence test and confidence interval for the difference in proportions for the paired-sample design. *Stat Med*, 17(8): 891-908.

Wenzel, D., and Zapf, A. (2013). Difference of two dependent sensitivities and specificities: comparison of various approaches. *Biom J*, 55(5): 705-718.

Examples

```
library(DTComPair)
t1 <- read.tab.paired(18, 14, 0, 18,
                      18, 12, 2, 18)
t1
sesp.diff.ci(t1, ci.method="wald", cont.corr=FALSE)
sesp.diff.ci(t1, ci.method="wald", cont.corr=TRUE)
sesp.diff.ci(t1, ci.method="agresti-min")
sesp.diff.ci(t1, ci.method="tango")
```

---

sesp.exactbinom   *Exact Binomial Test for Differences in Sensitivity and Specificity*

Description

Performs an exact binomial test for differences in sensitivity and specificity of two binary diagnostic tests in a paired study design.

Usage

```
sesp.exactbinom(tab)
```

Arguments

tab   An object of class `tab.paired`.

Details

The function performs a standard exact binomial test.

An exact binomial test is recommended when the number of patients with differing results for test 1 and test 2 (discordant pairs) is small, i.e. <20 (Zhou et al., 2011).

Value

A list containing:

sensitivity   A list containing `test1` (sensitivity of test 1), `test2` (specificity of test 2), `diff` (the difference in sensitivity, computed as `test2 - test1`) and the corresponding `p.value`.

specificity   A list containing `test1` (specificity of test 1), `test2` (specificity of test 2), `diff` (the difference in specificity, computed as `test2 - test1`) and the corresponding `p.value`.

method   The name of the method used to compare sensitivity and specificity, here "exactbinom".

References

Zhou, X., Obuchowski, N., and McClish, D. (2011). Statistical Methods in Diagnostic Medicine. Wiley Series in Probability and Statistics. John Wiley & Sons, Hoboken, New Jersey, 2nd edition.

See Also

`sesp.mcnemar` and `tab.paired`.

Examples

```
data(Paired1) # Hypothetical study data
ftable(Paired1)
paired.layout <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
paired.layout
exact.results <- sesp.exactbinom(paired.layout)
str(exact.results)
exact.results
exact.results$sensitivity["p.value"]
```

---

sesp.gen.mcnemar   *Generalized McNemar's test*

Description

Performs a generalized McNemar's test to jointly compare sensitivity and specificity.

Usage

```
sesp.gen.mcnemar(tab)
```

Arguments

tab   An object of class 'tab.paired'.

Value

A vector containing the test statistic and the p-value.

References

Lachenbruch P.A., Lynch C.J. (1998). Assessing screening tests: extensions of McNemar's test. *Stat Med*, 17(19): 2207-17.

See Also

[tab.paired()], [read.tab.paired()] and [sesp.mcnemar()]

Examples

```
# Example 1:
data(Paired1) # Hypothetical study data
ftable(Paired1)
paired.layout1 <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
print(paired.layout1)
sesp.gen.mcnemar(paired.layout1)

# Example 2 (from Lachenbruch and Lynch (1998)):
paired.layout2 <- read.tab.paired(
  d.a  = 850, d.b  = 40, d.c  = 60, d.d  =  50,
  nd.a =  60, nd.b = 25, nd.c = 15, nd.d = 900,
  testnames = c("T1", "T2")
)
print(paired.layout2)
sesp.gen.mcnemar(paired.layout2)
```

---

sesp.mcnemar   *McNemar Test for Comparison of Sensitivities and Specificities*

Description

Performs a McNemar Test for comparison of sensitivities and specificities of two binary diagnostic tests in a paired study design.

Usage

```
sesp.mcnemar(tab)
```

Arguments

tab   An object of class `tab.paired`.

Details

The test is performed as described by McNemar (1947).

Value

A list containing:

sensitivity	A list containing `test1` (sensitivity of test 1), `test2` (specificity of test 2), `diff` (the difference in sensitivity, computed as `test2 - test1`), the `test.statistic` and the corresponding `p.value`.
specificity	A list containing `test1` (specificity of test 1), `test2` (specificity of test 2), `diff` (the difference in specificity, computed as `test2 - test1`), the `test.statistic` and the corresponding `p.value`.
method	The name of the method used to compare sensitivity and specificity, here "mcnemar".

References

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153-7.

See Also

`sesp.exactbinom` and `tab.paired`.

Examples

```
data(Paired1) # Hypothetical study data
ftable(Paired1)
paired.layout <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
paired.layout
mcnem.results <- sesp.mcnemar(paired.layout)
str(mcnem.results)
mcnem.results
mcnem.results$sensitivity["p.value"]
```

---

sesp.rel	*Comparison of the accuracy of two tests using relative sensitivity and specificity*

Description

Calculates two-sided Wald confidence intervals and performs a Wald test for the relative sensitivity and specificity of two binary diagnostic tests in a paired study design.

Usage

```
sesp.rel(tab, alpha)
```

Arguments

tab	an object of class `tab.paired`.
alpha	significance level alpha used to compute two-sided 100(1-alpha)%-confidence intervals, the default is 0.05.

Details

If relative sensitivity>1, the percentage increase in sensitivity for `test2` relative to `test1` is computed as 100(relative sensitivity-1)%. If relative sensitivity<1 the percentage decrease in sensitivity for `test2` relative to `test1` is computed as 100(1-relative sensitivity)%. Percentage increase/decrease in specificity is computed in an analogous fashion.

Given the independence of relative sensitivity and relative specificity, a possible joint 100(1-alpha)% confidence region for {relative sensitivity, relative specificity} is formed by the rectangle {lcl.rel.sens, ucl.rel.sens} x {lcl.rel.spec, ucl.rel.spec}, where {lcl.rel.sens, ucl.rel.sens} and

{lcl.rel.spec, ucl.rel.spec} are 100(1-alpha*)% confidence intervals for relative sensitivity and relative specificity, respectively, and alpha*=1-sqrt(1-alpha).

The McNemar's test implemented in sesp.mcnemar is asymptotically equivalent to the Wald test implemented here.

Value

A list containing:

sensitivity	a named vector containing test1 (the sensitivity for test 1), test2 (the sensitivity for test 2), rel.sens (the relative difference between the two sensitivities, computed as test2/test1), se.log.rel.sens (the standard error for log(rel.sens)), lcl.rel.sens (the lower confidence limit for rel.sens), ucl.rel.sens (the upper confidence limit for rel.sens), and pval.rel.sens (the p-value from the test for the null hypothesis: relative sensitivity=1).
specificity	a named vector containing test1 (the specificity for test 1), test2 (the specificity for test 2), rel.spec (the relative difference between the two specificities, computed as test2/test1), se.log.rel.spec (the standard error for log(rel.spec)), lcl.rel.spec (the lower confidence limit for rel.spec), ucl.rel.spec (the upper confidence limit for rel.spec), and pval.rel.spec (the p-value from the test for the null hypothesis: relative specificity=1).
alpha	significance level alpha for 100(1-alpha)%-confidence intervals for rel.sens and rel.spec.

References

Alonzo, T. A., Pepe, M. S., & Moskowitz, C. S. (2002). Sample size calculations for comparative studies of medical tests for detecting presence of disease. *Statistics in medicine*, 21(6), 835-852.

See Also

sesp.diff.ci, sesp.mcnemar, and sesp.exactbinom.

Examples

```
data(Paired1) # Hypothetical study data
ftable(Paired1)
paired.layout <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
paired.layout
sesp.rel.results <- sesp.rel(paired.layout)
str(sesp.rel.results)
sesp.rel.results
```

---

tab.1test   *Tabulate Single Binary Diagnostic Test vs. Gold-Standard*

Description

Produces a contingency table of results from a single binary diagnostic test vs. the gold-standard results.

Usage

```
tab.1test(d, y, data = NULL, testname, ...)
```

Arguments

d	A numeric vector specifying the gold-standard results (1 = presence of disease, 0 = absence of disease).
y	A numeric vector specifying the results of the diagnostic test (1 = positive, 0 = negative).
data	An optional data frame, list or environment containing the required variables d and y.
testname	An optional character variable specifying the name of the diagnostic test, e.g. c("Test A"). If not supplied, the variable name is used as the testname.
...	Additional arguments (usually not required).

Value

Returns a list of class `tab.1test`:

tab.1test   A contingency table (matrix) of test results.

	Diseased	Non-diseased	Total
Test pos.	...	...	...
Test neg.	...	...	...
Total	...	...	...

testname   The name of the diagnostic test.

Note

Objects of class `tab.1test` are required as arguments for `acc.1test`, a function to compute the accuracy of a binary diagnostic test.

See Also

`tab.paired`, `acc.1test`, `acc.paired`.

Examples

```
data(Paired1) # Hypothetical study data
a <- tab.1test(d=d, y=y1, data=Paired1)
str(a)
a$tab.1test
a
```

---

tab.paired | *Tabulate Results from Two Binary Diagnostic Tests in a Paired Study Design*

Description

Produces contingency tables of results from two binary diagnostic tests evaluated in a paired study design.

Usage

```
tab.paired(d, y1, y2, data = NULL, testnames, ...)
```

Arguments

d	A numeric vector specifying the gold-standard results (1 = presence of disease, 0 = absence of disease).
y1	A numeric vector specifying the results of diagnostic test 1 (1 = positive, 0 = negative).
y2	A numeric vector specifying the results of diagnostic test 2 (1 = positive, 0 = negative).
data	An optional data frame, list or environment containing the required variables d, y1 and y2.
testnames	An optional vector specifying the names of diagnostic test 1 and diagnostic test 2, e.g. c("Test A","Test B"). If not supplied, the variable names are used as testnames.
...	Additional arguments (usually not required).

Value

Returns a list of class `tab.paired`:

diseased   A contingency table (matrix) of test results among *diseased* subjects.

	Test1 pos.	Test1 neg.	Total
Test2 pos.	...	...	...
Test2 neg.	...	...	...
Total	...	...	...

non.diseased   A contingency table (matrix) of test results among *non-diseased* subjects.

	Test1 pos.	Test1 neg.	Total
Test2 pos.	...	...	...
Test2 neg.	...	...	...
Total	...	...	...

testnames   The names of the diagnostic tests.

Note

Objects of class `tab.paired` are essential arguments for various functions in the `DTComPair`-package.

See Also

`print.tab.paired`, `read.tab.paired`, `tab.1test`.

Examples

```
data(Paired1) # Hypothetical study data
b <- tab.paired(d=d, y1=y1, y2=y2, data=Paired1)
str(b)
b$diseased
b$non.diseased
print(b)
```

---

tpffpf.rel   *Comparison of the accuracy of two tests using relative true positive
                                    and false positive fraction*

Description

Calculates two-sided Wald confidence intervals and performs a Wald test for the relative true positive fraction (TPF) (sensitivity) and false positive fraction (FPF) (i.e., one minus specificity) of two binary diagnostic tests in a paired study design.

This function is primarily intended for the analysis of paired **screen positive** studies, i.e. those paired studies where the disease (outcome) is ascertained using the gold standard test only in subjects who screen positive to either or both diagnostic tests. However, this function can also be used with data from standard paired studies, i.e. where the gold standard test is applied to all subjects.

Usage

```
tpffpf.rel(tab, alpha)
```

Arguments

tab   an object of class `tab.paired`.

alpha   significance level alpha used to compute two-sided 100(1-alpha)%-confidence
      intervals, the default is 0.05.

Details

If relative true positive fraction>1, the percentage increase in true positive fraction for Test2 relative
to Test1 is computed as 100(relative true positive fraction-1)%. If relative true positive fraction<1
the percentage decrease in true positive fraction for Test2 relative to Test1 is computed as 100(1-
relative true positive fraction)%. Percentage increase/decrease in false positive fraction is computed
in an analogous fashion.

Given the independence of relative TPR and relative TNR, a possible joint 100(1-alpha)% confi-
dence region for {relative TPF, relative FPF} is formed by the rectangle {`lcl.rel.tpf`, `ucl.rel.tpf`}
x {`lcl.rel.fpf`, `ucl.rel.fpf`}, where {`lcl.rel.tpf`, `ucl.rel.tpf`} and {`lcl.rel.fpf`, `ucl.rel.fpf`}
are 100(1-alpha*)% confidence intervals for relative TPF and relative FPF, respectively, and alpha*=1-
sqrt(1-alpha).

In screen positive studies, only relative TPF and relative FPF can be estimated from the data. Their
constituents, i.e. TPF and FPF for the two tests, are not estimable. Relative specificity is not es-
timable either. Therefore, `sesp.rel` should not be used to attempt to estimate those quantities from
studies with a paired screen positive design. McNemar's test (`sesp.mcnemar`) can, however, be used
to test the null hypothesis of equality in specificities in paired screen positive studies (Schatzkin et
al., 1987).

Value

A list containing:

tpf   a named vector containing `rel.tpf` (the relative true positive fraction, Test2 vs.
      Test1), `se.log.rel.tpf` (the standard error for log(`rel.tpf`)), `lcl.rel.tpf`
      (the lower confidence limit for `rel.tpf`), `ucl.rel.tpf` (the upper confidence
      limit for `rel.tpf`), and `pval.rel.tpf` (the p-value from the test for the null
      hypothesis: relative true positive fraction=1).

fpf   a named vector containing `rel.fpf` (the relative false positive fraction, Test2 vs.
      Test1), `se.log.rel.fpf` (the standard error for log(`rel.fpf`)), `lcl.rel.fpf`
      (the lower confidence limit for `rel.fpf`), `ucl.rel.fpf` (the upper confidence
      limit for `rel.fpf`), and `pval.rel.fpf` (the p-value from the test for the null
      hypothesis: relative false positive fraction=1).

alpha   significance level alpha for 100(1-alpha)%-confidence intervals for `rel.tpf` and
      `rel.fpf`.

References

Schatzkin, A., Connor, R. J., Taylor, P. R., & Bunnag, B. (1987). Comparing new and old screening
tests when a reference procedure cannot be performed on all screenees: example of automated
cytometry for early detection of cervical cancer. *American Journal of Epidemiology*, 125(4), 672-
678.

Cheng, H., & Macaluso, M. (1997). Comparison of the accuracy of two tests with a confirmatory procedure limited to positive results. *Epidemiology*, 104-106.

Alonzo, T. A., Pepe, M. S., & Moskowitz, C. S. (2002). Sample size calculations for comparative studies of medical tests for detecting presence of disease. *Statistics in medicine*, 21(6), 835-852.

See Also

`sesp.rel`.

Examples

```
# Data from Cheng and Macaluso (Table 2)
breast.cancer.data <- read.tab.paired(
 10, 24, 21, NA,
 13, 144, 95, NA,
 testnames=c("Mammography", "Physical examination")
)
breast.cancer.data
tpffpf.rel.results <- tpffpf.rel(breast.cancer.data)
str(tpffpf.rel.results)
tpffpf.rel.results
```

Index