Package 'DTComPair'

February 19, 2015

Type Package

Title Comparison of Binary Diagnostic Tests in a Paired Study Design

Version 1.0.3

Date 2014-02-15

Author Christian Stock, Thomas Hielscher

Maintainer Christian Stock `<stock@imbi.uni-heidelberg.de>`

Depends gee, PropCIs

Description This package contains functions to compare 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

NeedsCompilation no

Repository CRAN

Date/Publication 2014-02-16 00:43:17

R topics documented:

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

Description

This package contains functions to compare 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

Package:	DTComPair
Type:	Package
Version:	1.0.3
Date:	2014-02-15
License:	GNU >=2

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 et al., 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) or Zhou et al. (2011).

Author(s)

Christian Stock, Thomas Hielscher

Maintainer: Christian Stock <stock@imbi.uni-heidelberg.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.

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

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.

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

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.

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

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, ...)
```

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 variable containing the name of the diagnostic test.
`...`	Additional arguments (usually not required).

Details

The calculation of accuracy measures and their variances follows standard methodology, e.g. described in Pepe (2003) or Zhou et al. (2011). 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 variable containing the name of the diagnostic test.

References

Pepe, M. (2003). The statistical evaluation of medical tests for classifcation 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)
```

`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, ...)
```

Arguments

`tab`	An object of class `tab.paired`.
`alpha`	Significance level alpha for 100(1-alpha)%-confidence intervals, the default is 0.05.
`...`	Additional arguments, usually not required.

Details

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

Value

An 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
```

`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.

Usage

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

Arguments

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

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.

Usage

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

Arguments

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

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.

Usage

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

Arguments

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

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.

Usage

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

Arguments

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

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 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 “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.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`	A list with `test1` (the positive predictive value of test 1), `test2` (the positive predictive value of test 2), `rppv` (the relative positive predictive value, computed as `test1 / test2`), `se.log.rppv` (the standard error of the logarithm of `rppv`), `lcl.rppv` (the lower confidence limit of `rppv`), `ucl.rppv` (the upper confidence limit of `rppv`), 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), `rnpv` (the relative negative predictive value, computed as `test1 / test2`), `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`.
`method`	The name of the method used to compare predictive values, here “relative predictive values (rpv)”.
`alpha`	The 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.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"]
```

`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)
```

Index