Package 'testCompareR'

May 8, 2026

Type Package

Title Comparing Two Diagnostic Tests with Dichotomous Results using Paired Data

Version 1.1.1

Maintainer Kyle J. Wilson <kyle.jordan.wilson@googlemail.com>

Description

Provides a method for comparing the results of two binary diagnostic tests using paired data. Users can rapidly perform descriptive and inferential statistics in a single function call. Options permit users to select which parameters they are interested in comparing and methods for correction for multiple comparisons.
Confidence intervals are calculated using the methods with the best coverage. Hypothesis tests use the methods with the best asymptotic performance. A summary of the methods is available in Roldán-Nofuentes (2020) <doi:10.1186/s12874-020-00988-y>.
This package is targeted at clinical researchers who want to rapidly and effectively compare results from binary diagnostic tests.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Depends R (>= 2.10)

URL https://kajlinko.github.io/testCompareR/, https://github.com/Kajlinko/testCompareR

BugReports https://github.com/Kajlinko/testCompareR/issues

NeedsCompilation no

Author Kyle J. Wilson [cre, aut] (ORCID: <https://orcid.org/0000-0003-0151-7530>), Marc Henrion [aut] (ORCID: <https://orcid.org/0000-0003-1242-839X>),

José Antonio Roldán Nofuentes [aut] (ORCID:
<https://orcid.org/0000-0003-0251-5588>)

Repository CRAN

Date/Publication 2024-11-20 13:40:02 UTC

Contents

cass 2
cfpr 3
compareR 4
dataframeR 6
interpretR 7
plot.compareR 8
print.compareR 8
summariseR 9
summary.compareR 10

Index **12**

`cass` *Coronary Artery Surgery Study data*

Description

This data from the Coronary Artery Surgery Study evaluates two tests to determine the presence or absence of coronary artery disease by comparing to coronary angiography, the gold standard. Test 1 is an exercise stress test and Test 2 is a clinical history of chest pain.

Usage

```
cass
```

Format

A data frame with 871 rows and 3 columns:

exercise Dichotomous result on exercise stress testing.

cp Presence of absence of chest pain based on medical history.

angio Dichotomous result on coronary angiography.

Details

All three variables are dichotomous. 1 indicates a positive result; 0 indicates a negative result.

This data was originally presented in Weiner et al. (1979).

Source

doi:10.1056/NEJM197908023010502

References

Weiner et al. (1979)) N Engl J Med. 1979;301(5):230-5 doi:10.1056/NEJM197908023010502

`cfpr` *US Cystic Fibrosis Patient Registry data*

Description

This data from the Cystic Fibrosis Foundation's Patient Registry (USA) evaluates risk factors for pulmonary exacerbation in patients with cystic fibrosis. The two risk factors evaluated are previous pulmonary exacerbation and previous colonisation with Pseudomonas aeruginosa. Each of the two risk factors was evaluated using data from 1995. If an instance occurred at any point in 1995 the 'test' was considered positive. If negative throughout 1995 the 'test' was considered negative. The gold standard was evidence of pulmonary exacerbation at any point in 1996.

Usage

```
cfpr
```

Format

A data frame with 11,960 rows and 3 columns:

pulm.exac Presence or absence of previous pulmonary exacerbation.

pseudomonas Presence or absence of Pseudomonas aeruginosa infection.

infection Presence or absence of severe infection (gold standard).

Details

All three variables are dichotomous. 1 indicates presence; 0 indicates absence.

This data was originally presented in Moskowitz and Pepe (2006).

Source

Data was sourced directly from the referenced paper. For up-to-date data requests contact: Cystic Fibrosis Foundation

References

Moskowitz and Pepe (2006) Clinical Trials. 2006;3(3):272-9. doi:10.1191/1740774506cn147oa

compareR *compareR*

Description

Calculates descriptive statistics and performs statistical inference on two binary diagnostic tests in a single function call. Handles multiple comparisons using methods in 'p.adjust()'.

Usage

```
compareR(
  df = NULL,
  test1 = NULL,
  test2 = NULL,
  gold = NULL,
  interpret = FALSE,
  alpha = 0.05,
  margins = FALSE,
  multi_corr = "holm",
  cc = TRUE,
  dp = 1,
  sesp = TRUE,
  ppvnpv = TRUE,
  plrnlr = TRUE,
  conf.int = "contemporary",
  test.names = c("Test 1", "Test 2"),
  ...
)
```

Arguments

df	A data frame or matrix with 3 columns (test1, test2, gold). Flexible coding of positive and negative results permitted.
test1	Either a vector of values for Test 1 (if df is NULL) or a string or integer value to be used for subsetting df.
test2	Either a vector of values for Test 2 (if df is NULL) or a string or integer value to be used for subsetting df.
gold	Either a vector of values for the gold standard test (if df is NULL) or a string or integer value to be used for subsetting df.
interpret	If TRUE provides a verbose plain English interpretation of the output by calling interpretR(). Defaults to FALSE.
alpha	An alpha value. Defaults to 0.05.
margins	A Boolean value indicating whether the contingency tables should have margins containing summed totals of rows and columns.
multi_corr	Method for multiple comparisons. Uses 'p.adjust.methods'.

cc	A Boolean value indicating whether McNemar's test should be applied with continuity correction.
dp	Number of decimal places of output in summary tables. Defaults to 1.
sesp	A Boolean value indicating whether output should include sensitivity and specificity.
ppvnpv	A Boolean value indicating whether output should include positive and negative predictive values.
plrnlr	A Boolean value indicating whether output should include positive and negative likelihood ratios.
conf.int	A character string, either "contemporary" or "classic". Indicates whether function should use contemporary or classic statistical methods to calculate confidence intervals.
test.names	A vector of length two giving the names of the two different binary diagnostic tests. This argument is not relevant when testing a single binary diagnostic test.
...	Rarely needs to be used. Allows additional arguments to be passed to internal functions.

Details

Confidence intervals for prevalence, diagnostic accuracies and predictive values are calculated using the interval for binomial proportions described by Yu et al. (2014) by default. Setting conf.int = "classic" uses the Clopper-Pearson method. Confidence intervals for likelihood ratios are calculated using the methods recommended by Martín-Andrés and Álvarez-Hernández (2014). Hypothesis testing for diagnostic accuracies uses different methods depending on disease prevalence and number of participants or samples as described by Roldán-Nofuentes and Sidaty-Regad (2019). Global hypothesis testing for predictive values uses a method described by Roldán-Nofuentes et al. (2012), with subsequent individual tests (where indicated) performed using methods described by Kosinksi (2012). The methods for hypothesis testing- for likelihood ratios are taken from Roldán-Nofuentes & Luna del Castillo (2007).

An excellent summary of these methods is provided by Roldán-Nofuentes (2020) along with an open-source program (compbdt) licensed under GPL-2. This R package can be considered an extension of this work and is therefore distributed under the same license. Please consider citing Roldán-Nofuentes (2020) when you are citing this package.

Value

A list object summarising all calculated descriptive and inferential statistics.

References

Yu, Guo & Xu (2014) JSCS. 2014; 84:5,1022-1038 doi:10.1080/00949655.2012.738211

Clopper & Pearson (1934) Biometrika. 1934; 26,404-413 doi:10.2307/2331986

Martín Andrés & Álvarez Hernández (2014) Stat Comput. 2014; 24,65–75 doi:10.1007/s11222-01293535

Roldán-Nofuentes & Sidaty-Regad (2019) JSCS. 2019; 89:14,2621-2644 doi:10.1080/00949655.2019.1628234

Roldán-Nofuentes, Luna del Castillo & Montero-Alonso (2012) Comput Stat Data Anal. 2012; 6,1161–1173. doi:10.1016/j.csda.2011.06.003

Kosinski (2012) Stat Med. 2012; 32,964-977 doi:10.1002/sim.5587

Roldán-Nofuentes, Luna del Castillo (2007) Stat Med. 2007; 26:4179–201. doi:10.1002/sim.2850

Roldán-Nofuentes (2020) BMC Med Res Methodol. 2020; 20,143 doi:10.1186/s1287402000988y

Examples

```
# load data
df <- cfpr

# run compareR function
compareR(df,
  margins = TRUE, multi_corr = "bonf",
  test.names = c("pulm.exac", "pseudomonas")
)
```

`dataframeR` *dataframeR*

Description

Produces a data frame which can be used by the compareR function using values commonly found in published literature. Useful for reviews and meta-analyses.

Usage

```
dataframeR(s11, s10, s01, s00, r11, r10, r01, r00)
```

Arguments

`s11`	Number of cases where Test 1 is positive, Test 2 is positive and gold standard is positive.
`s10`	Number of cases where Test 1 is positive, Test 2 is negative and gold standard is positive.
`s01`	Number of cases where Test 1 is negative, Test 2 is positive and gold standard is positive.
`s00`	Number of cases where Test 1 is negative, Test 2 is negative and gold standard is positive.
`r11`	Number of cases where Test 1 is positive, Test 2 is positive and gold standard is negative.
`r10`	Number of cases where Test 1 is positive, Test 2 is negative and gold standard is negative.
`r01`	Number of cases where Test 1 is negative, Test 2 is positive and gold standard is negative.
`r00`	Number of cases where Test 1 is negative, Test 2 is negative and gold standard is negative.

Details

Understanding the parameter names: s & r represent positive and negative results for the gold standard test, respectively. The first digit represents a positive (1) or negative (0) result for Test 1. The second digit represents a positive (1) or negative (0) result for Test 2.

Value

A data frame populated with zeros and ones indicating positive or negative test results which can be passed to the compareR function.

Examples

```
# build data frame using numbers
dataframeR(3, 3, 3, 3, 3, 3, 3, 3)
```

---

interpretR *interpretR*

Description

Provides a plain English readout of the results of the compareR function.

Usage

```
interpretR(result)
```

Arguments

result	A list object with class 'compareR' output from the compareR function.

Value

A plain English summary of the findings produced by the compareR function.

Examples

```
# simulate data
test1 <- c(rep(1, 300), rep(0, 100), rep(1, 55), rep(0, 145))
test2 <- c(rep(1, 280), rep(0, 120), rep(1, 45), rep(0, 155))
gold <- c(rep(1, 400), rep(0, 200))
dat <- data.frame(test1, test2, gold)

# compare with compareR
result <- compareR(dat)

# provide a plain English readout with interpretR
interpretR(result)
```

plot.compareR *Plot a compareR object*

Description

An S3 method to plot a simple visualisation of the results from the compareR function.

Usage

```
## S3 method for class 'compareR'
plot(x, ...)
```

Arguments

x	An object of class compareR.
...	Arguments such as graphical parameters. Not currently in use.

Details

Method to plot the most commonly used results of the compareR output.

Value

A visualisation of the results for diagnostic accuracies and predictive values from the compareR output.

Examples

```
# generate result
res <- compareR(cass, test1 = "exercise", test2 = "cp",
                gold = "angio",
                test.names = c("ExerciseStressTest", "ChestPain"))

# run print method
plot(res)
```

print.compareR *Print a compareR object*

Description

An S3 method to print the results verbose from the compareR function.

Usage

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

Arguments

x	An object of class compareR.
...	Further arguments passed to or from other methods.

Details

Method to print the pertinent results of the compareR output.

Value

A printed results table of the compareR output.

Examples

```
# generate result
res <- compareR(cass, test1 = "exercise", test2 = "cp",
                gold = "angio",
                test.names = c("ExerciseStressTest", "ChestPain"))

# run print method
print(res)
```

---

summariseR *summariseR*

Description

Summarises descriptive statistics associated with a single binary diagnostic test.

Usage

```
summariseR(df, dp = 1)
```

Arguments

df	A data frame or matrix with 2 columns (test1, gold). Flexible coding of positive and negative results permitted.
dp	Number of decimal places of output in summary tables. Defaults to 1. Kappa defaults to 3 decimal places unless user selects more.

Details

Confidence intervals for prevalence, diagnostic accuracies and predictive values are calculated using the interval for binomial proportions described by Yu et al. (2014). Confidence intervals for likelihood ratios are calculated using the methods recommended by Martín-Andrés and Álvarez-Hernández (2014). Cohen's kappa is a value between -1 and 1 which describes the agreement of the two tests, taking account of random agreement. A score of zero or less indicates the agreement could be entirely due to chance.

Value

A summary of the descriptive statistics of a binary diagnostic test, compared to a gold standard.

References

Yu, Guo & Xu (2014) JSCS. 2014; 84:5,1022-1038 doi:10.1080/00949655.2012.738211

Martín Andrés & Álvarez Hernández (2014) Stat Comput. 2014; 24,65–75 doi:10.1007/s11222-01293535

Cohen (1960) Educ Psychol Meas. 1960; 20(1),37–46 doi:10.1177/001316446002000104

Examples

```
# simulate data
test1 <- c(rep(1, 300), rep(0, 100), rep(1, 55), rep(0, 145))
gold <- c(rep(1, 400), rep(0, 200))
dat <- data.frame(test1, gold)

# summarise descriptive statistics
result <- summariseR(dat, dp = 4)
```

`summary.compareR` *Summarise a compareR object*

Description

An S3 method to summarise the rather verbose output of the compareR function.

Usage

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

Arguments

`object`	An object of class compareR.
`...`	Additional arguments affecting the summary produced.

Details

Method to summarise the verbose compareR output.

Value

A summary of the compareR output.

Examples

```
# generate result
res <- compareR(cass, test1 = "exercise", test2 = "cp",
                gold = "angio",
                test.names = c("ExerciseStressTest", "ChestPain"))

# run summary method
summary(res)
```

Index

∗ **datasets**
cass, 2
cfpr, 3

cass, 2
cfpr, 3
compareR, 4

dataframeR, 6

interpretR, 7

plot.compareR, 8
print.compareR, 8

summariseR, 9
summary.compareR, 10