Package 'crs'

February 14, 2019

Type Package

Title CRS (composite reference standard) confidence intervals calculator

Version 0.1.0

Author Ty Stanford

Maintainer Ty Stanford <tyman.stanford@lbtinnovations.com>

Description Functions to perform CRS calculations on sample data.

License GPL-2.0

Encoding UTF-8

LazyData true

Suggests testthat

Imports dplyr,
tibble,
binom,
foreach

RoxygenNote 6.1.1

R topics documented:

brenton2019 2
check_col 2
get_sens_spec 3
hawkins2001 4
perform_crs 4

Index **6**

brenton2019 *Data from Benton et al. (2019)*

Description

These are the data from from Benton et al. (2019) containing the APAS (A, the index test), St Vincent's standard workflow (S, imperfect truth) and the panel consensus (P, resolver) variables.

Usage

```
data(brenton2019)
```

Format

A data frame with 881 rows and 3 variables:

A APAS values; 1=not significan growth (0 and 10^6 CFU/L), 2=significant growth (10^7 and 10^8+ CFU/L)

S St Vincent's standard workflow values; 1=not significan growth (0 and 10^6 CFU/L), 2=significant growth (10^7 and 10^8+ CFU/L)

P Panel consensus values; 1=not significan growth (0 and 10^6 CFU/L), 2=significant growth (10^7 and 10^8+ CFU/L)

check_col *Check a vector of column names are all contained in a suppled data frame*

Description

Check a vector of column names are all contained in a suppled data frame

Usage

```
check_col(df, cols)
```

Arguments

df	data frame
cols	character vector of column names to be checked whether they exist in df

Value

TRUE if all cols exist in df, otherwise an error is thrown

Author(s)

Ty Stanford <tyman@lbtinnovations.com>

Examples

```
library(tibble)
adf <- data_frame(A = 1:2, B = 3:4, `C=1` = LETTERS[1:2])
check_col(adf, LETTERS[1:2])
check_col(adf, c("A", "B", "C=1"))
```

---

get_sens_spec *Calculate sensitivity and specificity*

Description

Calculate sensitivity and specificity from a confusion matrix

Usage

```
get_sens_spec(tab, pos = 2, alpha = 0.05)
```

Arguments

tab	a confusion matrix (object of class "matrix" or table")
pos	the levels of the margins that are considered "positive"
alpha	confidence level of CIs (default=0.05)

Value

A tibble with columns `param` ("sens" or "spec"), `cases`, `correct`, `est`, `lo`, `up`.

Author(s)

Ty Stanford <tyman@lbtinnovations.com>

Examples

```
library(tibble)
set.seed(1234)
AB <- data_frame(A=sample(1:3, 20, replace=TRUE), B=sample(1:3, 20, replace=TRUE))
get_sens_spec(with(AB,table(A,B)), pos = 2:3)
```

hawkins2001 *Representation of the data from Hawkins et al. (2001)*

Description

These are a recreation of a sample line data that prodice the cross tabulations seen in Hawkins et al. (2001) containing the index test (`index`), reference (`ref`) and the resolver (`resolve`) variables.

Usage

```
data(hawkins2001)
```

Format

A data frame with 3,000 rows and 3 variables:

index index test values; 1=negative, 2=positive

ref Reference values; 1=negative, 2=positive

resolve Resolver values; 1=negative, 2=positive

perform_crs *Perform CRS sens and spec calulations using a one sample-per-row input dataset*

Description

Performs the CRS analysis

Usage

```
perform_crs(dat, index, imperfect, resolver, trans_method = "probit",
  alpha = 0.05)
```

Arguments

`dat`	data frame that contains the index test, imperfect truth and resolver columns (see details)
`index`	string of column name in `dat` corresponding to the index test (1=negative, 2=positive)
`imperfect`	string of column name in `dat` corresponding to the imperfect truth (1=negative, 2=positive)
`resolver`	string of column name in `dat` corresponding to the resolver (1=negative, 2=positive, NA=not known)
`trans_method`	one of "probit" (default), "logit", "loglog", "arcsin"
`alpha`	confidence level (default=0.05)

Details

Currently `perform_mcrs` requires `dat` to contain variables with negative and positive values as the integers `1`=negative and `2`=positive

Value

A tibble with columns `param`, `p`, `var_p`, `se_p`, `p_lo`, `p_up`

Author(s)

Ty Stanford <tyman@lbtinnovations.com>

Examples

```
library(tibble)
data(brenton2019)
perform_crs(brenton2019, "A", "S", "P")
```

Index

∗Topic **datasets**
brenton2019, 2
hawkins2001, 4

brenton2019, 2

check_col, 2

get_sens_spec, 3

hawkins2001, 4

perform_crs, 4