

Package ‘ibist’

September 12, 2026

Title Data and Functions for Introduction to Biostatistics with R

Version 0.1-0

Author Elizabeth Schifano [aut] (ORCID:

<<https://orcid.org/0000-0002-9793-332X>>),

Jun Yan [aut, cre] (ORCID: <<https://orcid.org/0000-0003-4401-7296>>)

Maintainer Jun Yan <jun.yan@uconn.edu>

Description Provides datasets and supporting functions for the book
Introduction to Biostatistics with R by Schifano and Yan (2026+),
published by Taylor & Francis. The package is intended for teaching
introductory biostatistics and for reproducing examples in the text.

Depends R (>= 4.4.0)

VignetteBuilder knitr

License GPL (>= 3)

URL <https://github.com/ibist/ibist-R>

BugReports <https://github.com/ibist/ibist-R/issues>

Imports stats, Rcpp

LinkingTo Rcpp

Suggests ggplot2, knitr, rlang, testthat (>= 3.0.0)

LazyData true

Encoding UTF-8

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Repository CRAN

Date/Publication 2026-09-12 13:10:02 UTC

Contents

demo_clt	2
depress	3
fluorosis	5
logistic_gof	6
midbp	8
mihdl	8
nrs	9
power.p1s.test	10
power.p2s.test	12
rate.1s.ci	14
rate.2s.ci	16
rate.test	17
rds	19
wang.paired.ci	20
wang.paired.reject	21
Index	22

demo_clt	<i>Demonstrate the Central Limit Theorem</i>
----------	--

Description

The demo_clt() function generates plots to illustrate the Central Limit Theorem (CLT) using a specified random number generator. The function displays standardized sampling distributions for different sample sizes and overlays the standard normal density.

Usage

```
demo_clt(rng, n, nrep = 10000, ..., pmean = NULL, psd = NULL)
```

Arguments

rng	A random number generator function taking the sample size as its first argument (e.g., runif, rnorm, rgamma).
n	A numeric vector of sample sizes (e.g., c(5, 10, 20, 40)).
nrep	The number of repetitions for generating sample means (default is 10000).
...	Additional arguments passed to the random number generator (e.g., shape and rate for rgamma).
pmean	The population mean of the distribution. If NULL, it is estimated from a large Monte Carlo sample.
psd	The population standard deviation of the distribution. If NULL, it is estimated from a large Monte Carlo sample.

Value

A ggplot2 object showing the standardized sampling distributions for different sample sizes, compared against the standard normal curve.

Examples

```
set.seed(123)
demo_clt(runif, n = c(5, 20), nrep = 100, min = 0, max = 1,
         pmean = 0.5, psd = sqrt(1 / 12))

demo_clt(rgamma, n = c(5, 20), shape = 2, rate = 1,
         nrep = 100, pmean = 2, psd = sqrt(2)
)
```

depress

*Depression, insomnia, and stress dataset (Kalmbach et al., 2018)***Description**

This dataset is derived from a longitudinal study examining the role of insomnia and cognitive intrusions in predicting future depression. It includes participants who were not depressed at baseline and who met inclusion criteria on age, missingness, and stress exposure.

Usage

```
depress
```

Format

A data frame with 1730 observations and 9 variables:

age Age in years.

gender Gender of the participant (coded per original survey).

stressorA Indicator for experiencing at least one stressor in the past year.

depress1yr Depression status at 1-year follow-up.

depress2yr Depression status at 2-year follow-up.

depress1or2yr Depression at either 1 or 2 years.

IEStert Tertile of cognitive intrusion score based on the Impact of Event Scale (IES).

IESgrp Grouped cognitive intrusion: 0 = lowest tertile, 1 = middle tertile, 2 = highest tertile.

insom Insomnia indicator defined as: sleep onset latency (SOL) > 30 minutes or wake after sleep onset (WASO) > 30 minutes.

Details

The dataset is constructed by applying the following criteria:

- Baseline depression score < 10.
- Age >= 18.
- Non-missing gender, cognitive intrusion (IES), and WASO.
- At least one stressor in the past year.
- Exclusion of extreme nocturnal wakefulness (> 90 minutes).

The resulting dataset (n = 1730) includes all three cognitive intrusion tertiles. In the original paper, the analytic sample was further restricted by removing observations with missing `IEStert`, which effectively excludes the middle tertile (coded as `IESgrp = 1`), yielding a smaller sample (n = 1126). This optional restriction is not applied here but can be reproduced by subsetting:

```
depress_paper <- subset(depress, !is.na(IEStert))
```

Variable definitions follow the original study documentation provided in the supplementary spreadsheet (S1 Table).

Source

Kalmbach, D. A., Pillai, V., and Drake, C. L. (2018). Nocturnal insomnia symptoms and stress-induced cognitive intrusions in risk for depression: a 2-year prospective study. *PLOS ONE*, 13(2), e0192088.

References

Kalmbach, D. A., Pillai, V., and Drake, C. L. (2018). Nocturnal insomnia symptoms and stress-induced cognitive intrusions in risk for depression: a 2-year prospective study. *PLOS ONE*, 13(2), e0192088.

Examples

```
data(depress)
str(depress)

# proportion developing depression within 2 years
mean(depress$depress1or2yr)

# insomnia prevalence
mean(depress$insom)

# reproduce analytic sample used in the paper
depress_paper <- subset(depress, !is.na(IEStert))
```

fluorosis*Dental Fluorosis Study Data*

Description

These data arise from a prospective study examining the association between dental fluorosis and amoxicillin use during early childhood.

Usage

fluorosis

Format

A data frame with 24 rows and 5 variables:

om Oral hygiene measure (factor with levels 0, 1)

fl_lev Fluoride level (factor with levels 1, 2, 3)

amox Amoxicillin use during early childhood (factor: 0 = no, 1 = yes)

fluorosis Dental fluorosis status (factor: 0 = absent, 1 = present)

count Number of subjects in the corresponding cell

Details

Dental fluorosis — a developmental defect of tooth enamel — was assessed as a binary outcome (presence or absence) when participants were approximately 9 years old. Information on fluoride intake, infections, and amoxicillin use during early childhood (birth to 32 months) was collected as part of the Iowa Fluoride Study.

The dataset is provided in aggregated form as a four-way contingency table with cell counts.

The data have a 2 x 3 x 2 x 2 factorial structure: oral hygiene (2 levels), fluoride level (3 levels), amoxicillin use (2 levels), and fluorosis status (2 levels). The counts sum to 579 subjects.

Source

Hong L, Levy SM, Warren JJ, Dawson DV, Bergus GR, Wefel JS (2005). Association of amoxicillin use during early childhood with developmental tooth enamel defects. *Archives of Pediatrics & Adolescent Medicine*, 159(10), 943–948.

Examples

```
data(fluorosis)
with(fluorosis,
     xtabs(count ~ om + fl_lev + amox + fluorosis))
```

logistic_gof

*Pearson Chi-squared Goodness-of-Fit Test for Logistic Regression***Description**

Performs the Pearson chi-squared goodness-of-fit (GoF) test for a logistic regression model fitted via `glm(..., family = "binomial")`. The test assesses calibration by comparing observed and expected counts across covariate patterns.

Usage

```
logistic_gof(fit, min_n = 5, min_expected = 5, pool = TRUE)
```

Arguments

<code>fit</code>	A fitted <code>glm</code> object with <code>family = "binomial"</code> .
<code>min_n</code>	Minimum total count per group for pooling. Default is 5.
<code>min_expected</code>	Minimum expected count per group for pooling. Default is 5.
<code>pool</code>	Logical; if <code>TRUE</code> , adjacent groups (ordered by fitted probabilities) are pooled to meet minimum thresholds. Default is <code>TRUE</code> .

Details

The Pearson GoF statistic is defined as

$$T_P = \sum_{\ell=1}^J \frac{(O_{\ell} - E_{\ell})^2}{V_{\ell}},$$

where O_{ℓ} is the observed number of events, $E_{\ell} = n_{\ell} \hat{\pi}_{\ell}$ is the expected number of events, and $V_{\ell} = n_{\ell} \hat{\pi}_{\ell} (1 - \hat{\pi}_{\ell})$ is the variance for the ℓ -th covariate pattern. Under the null hypothesis of correct model specification, T_P approximately follows a chi-squared distribution with $J - (k + 1)$ degrees of freedom, where J is the number of groups and $k + 1$ is the number of model parameters.

This test requires replicated covariate patterns or grouped binomial data. When groups are small, optional pooling can be applied to ensure stable chi-squared approximation.

The function supports three types of binomial responses:

- Binary outcomes (0/1)
- Proportions with weights (grouped binomial)
- Two-column matrix response via `cbind(success, failure)`

When no replication exists (i.e., each observation has a unique covariate pattern), the Pearson GoF test is not appropriate. In such cases, consider alternative methods such as the Hosmer–Lemeshow test.

Pooling stabilizes the test by merging adjacent groups with small counts, trading a small bias for reduced variance in the chi-squared approximation.

Value

An object of class "htest" with components:

statistic	The Pearson chi-squared statistic.
parameter	Degrees of freedom.
p.value	p-value of the test.
method	Description of the test.
data.name	Description of the data.

Additional components (not printed) include:

observed	Observed counts by group.
expected	Expected counts by group.
group_n	Group sizes.
groups	Data frame of grouped summaries.

See Also

[glm](#), [chisq.test](#)

Examples

```
## Example: Surfactant use and birthweight (RDS data)
data(rds, package = "ibist")

fit <- glm(death ~ surf + bwt,
  data = rds, weights = count,
  family = "binomial"
)

## Pearson GoF test with pooling (default)
(res <- logistic_gof(fit))

## Without pooling (may be unstable if small groups exist)
logistic_gof(fit, pool = FALSE)

## Inspect grouped diagnostics
res$groups
```

midbp	<i>Blood Pressure Cohort Data (midbp)</i>
-------	---

Description

Aggregated data of cases and person-years by diastolic blood pressure category and gender.

Usage

midbp

Format

A data frame with 14 rows and 4 variables:

dbp Diastolic blood pressure category (1–7).

gender Gender (Men, Women).

cases Number of observed cases.

pyears Person-years of follow-up.

Details

The dataset is aggregated by DBP category and gender, suitable for rate modeling (e.g., Poisson regression with offset).

Examples

```
data(midbp)
head(midbp)
```

mihdl	<i>HDL Cholesterol and Event Counts by Gender</i>
-------	---

Description

Aggregated data on event counts and person-years by HDL cholesterol category and gender. Suitable for Poisson rate modeling and incidence rate comparisons.

Usage

mihdl

Format

A data frame with 24 rows and 4 variables:

hdl HDL category (integer, 1–12)

gender Gender (factor: Men, Women)

cases Number of events

pyears Person-years at risk

Details

The dataset provides grouped count data for modeling rates using Poisson regression with an offset for log person-years.

Examples

```
data(mihdl)
fit <- glm(cases ~ factor(hdl) + gender,
           offset = log(pyears),
           family = poisson,
           data = mihdl)
summary(fit)
```

nrs

Non-restorative sleep and physical activity (Japan cohort study)

Description

A large observational dataset from a cohort study conducted in Japan to examine the association between non-restorative sleep (NRS) and physical activity, gender, and age. The data are used to illustrate logistic regression modeling for a binary outcome in a large-sample setting.

Usage

nrs

Format

A data frame with 91,795 subjects (90,122 complete records) on the following variables:#'

id Subject identifier.

gender Gender of the subject (F = Female; M = Male).

age Age in years in 2013.

ex Indicator of regular exercise in 2013 (integer-coded).

pa Physical activity measure in 2013 (integer-coded).

nrs Indicator of non-restorative sleep in 2013 (1 = presence; 0 = absence).

Details

Non-restorative sleep (NRS) is defined as a subjective feeling of lack of refreshment on awakening and reflects qualitative aspects of sleep. Hidaka et al. (2019) analyzed these data using logistic regression to assess whether the probability of NRS is associated with physical activity, gender, and age in a large cohort of adult subjects in Japan. Within this package, the dataset is provided for methodological illustration of binary regression models rather than for substantive epidemiological inference.

Missing values are represented as NA, which are present in 'ex', 'pa', and 'nrs'.

Source

Hidaka et al. (2019).

Examples

```
data(nrs)
summary(nrs)
```

power.p1s.test

One-sample proportion power and sample size calculation

Description

Computes power, sample size, or detectable effect size for a one-sample binomial test. The interface mirrors `stats::power.prop.test()`, but for the one-sample setting.

Usage

```
power.p1s.test(
  n = NULL,
  p0 = NULL,
  p1 = NULL,
  power = NULL,
  sig.level = 0.05,
  alternative = c("two.sided", "less", "greater"),
  correct = FALSE,
  exact = FALSE,
  exact.method = c("quantile", "midp", "cp"),
  strict = TRUE,
  tol = .Machine$double.eps^0.5,
  max_n = 1e+07,
  size.rule = c("close", "minimal"),
  alpha.min.frac = 0.9
)
```

Arguments

<code>n</code>	Sample size for the single group.
<code>p0</code>	Null hypothesis proportion.
<code>p1</code>	Alternative hypothesis proportion.
<code>power</code>	Desired power.
<code>sig.level</code>	Significance level.
<code>alternative</code>	Character string specifying the alternative hypothesis; one of "two.sided", "less", or "greater".
<code>correct</code>	Logical; if TRUE, apply continuity correction in the normal approximation.
<code>exact</code>	Logical; if TRUE, use an exact binomial method.
<code>exact.method</code>	Method used for exact binomial power calculation. "quantile" uses fixed binomial rejection regions for stable power and sample-size inversion; "midp" applies the mid-p adjustment; "cp" inverts Clopper–Pearson acceptance regions to guarantee size not exceeding <code>sig.level</code> .
<code>strict</code>	Logical; if TRUE and <code>alternative</code> = "two.sided", computes power accounting for both tails explicitly. Otherwise uses the standard one-sided approximation.
<code>tol</code>	Numerical tolerance used in root finding.
<code>max_n</code>	Maximum allowable sample size when solving for <code>n</code> .
<code>size.rule</code>	Character string specifying the rule used to select the sample size when <code>exact</code> = TRUE. Must be one of "close" (default) or "minimal". "close" selects the smallest sample size n such that the achieved significance level satisfies $\alpha_{\text{act}} \leq \alpha$, the power meets the target, and the achieved level is not too far below the nominal level:

$$\alpha_{\text{act}} \geq c\alpha,$$

where $c = \text{alpha.min.frac}$. This rule avoids overly conservative designs and makes more efficient use of the type I error budget.

"minimal" selects the smallest sample size n such that the achieved significance level satisfies $\alpha_{\text{act}} \leq \alpha$ and the power is at least the target value. This is the classical exact test design but may be conservative, with $\alpha_{\text{act}} \ll \alpha$.

<code>alpha.min.frac</code>	Numeric value in (0, 1) specifying the minimum fraction of the nominal significance level required for the achieved level when <code>size.rule</code> = "close". The achieved significance level must satisfy
-----------------------------	--

$$\alpha_{\text{act}} \geq \text{alpha.min.frac} \times \alpha.$$

The default value 0.9 requires the achieved level to be at least 90% of the nominal level. Smaller values allow more conservative designs, while larger values enforce closer agreement with the nominal level.

See also `achieved.sig.level` in the returned value.

Details

Power can be computed using a normal approximation or exact binomial methods. When `exact = TRUE`, the exact test is determined by `exact.method`.

Exactly one of `n`, `p0`, `p1`, `power`, or `sig.level` must be `NULL`; the missing quantity is solved numerically.

Value

An object of class `"power.htest"` containing the computed quantity and test specifications.

Examples

```
## Normal approximation (default)
power.p1s.test(n = 50, p0 = 0.1, p1 = 0.25)

## Exact binomial power (quantile-based, default exact method)
power.p1s.test(n = 50, p0 = 0.1, p1 = 0.25, exact = TRUE)

## Exact mid-p power
power.p1s.test(
  n = 50, p0 = 0.1, p1 = 0.25,
  exact = TRUE, exact.method = "midp"
)

## Exact Clopper--Pearson power (guaranteed size control)
power.p1s.test(
  n = 50, p0 = 0.1, p1 = 0.25,
  exact = TRUE, exact.method = "cp"
)
```

power.p2s.test	<i>Power Calculation for Two-Sample Proportion Test with Unequal Group Sizes</i>
----------------	--

Description

Computes power, sample size, detectable proportions, or significance level for a two-sample test comparing proportions, allowing unequal group sizes through a user-specified allocation ratio.

Usage

```
power.p2s.test(
  n = NULL,
  p1 = NULL,
  p2 = NULL,
  sig.level = 0.05,
  power = NULL,
```

```

    group.rate = 1,
    alternative = c("two.sided", "one.sided"),
    correct = TRUE,
    strict = FALSE,
    tol = .Machine$double.eps^0.25
  )

```

Arguments

<code>n</code>	Sample size in the first group. The second group size is $n * \text{group.rate}$.
<code>p1</code>	Proportion in the first group.
<code>p2</code>	Proportion in the second group.
<code>sig.level</code>	Significance level (type I error rate), must be in $[0, 1]$.
<code>power</code>	Target power of the test.
<code>group.rate</code>	Ratio of sample sizes between the second and first groups, defined as n_2/n_1 . Defaults to 1 (equal group sizes).
<code>alternative</code>	Character string specifying the alternative hypothesis, either "two.sided" or "one.sided".
<code>correct</code>	Logical; if TRUE, apply continuity correction in the normal approximation.
<code>strict</code>	Logical; if TRUE and <code>alternative = "two.sided"</code> , computes power accounting for both tails explicitly. Otherwise uses the standard one-sided approximation.
<code>tol</code>	Tolerance for numerical root-finding when solving for an unknown parameter.

Details

Exactly one of `n`, `p1`, `p2`, `power`, or `sig.level` must be NULL, and that parameter is solved from the others.

The calculation is based on the normal approximation to the binomial distribution. Let $n_1 = n$ and $n_2 = n \times \text{group.rate}$. The test statistic uses a pooled variance under the null and an unpooled variance under the alternative.

When `strict = TRUE` for a two-sided test, power is computed as the sum of tail probabilities. Otherwise, a one-sided approximation is used.

The code is adapted from `stats::power.power.test()` by allowing different group sizes and continuity correction.

Value

An object of class "power.htest" with components:

<code>n</code>	Sample size in the first group.
<code>p1</code>	Proportion in the first group.
<code>p2</code>	Proportion in the second group.
<code>sig.level</code>	Significance level.
<code>power</code>	Power of the test.

alternative	Type of alternative hypothesis.
note	Clarifies that n refers to the first group size.
method	Description of the method.

Note

This function generalizes `stats::power.prop.test()` by allowing unequal group sizes through `group.rate`. When `group.rate = 1`, the two functions are equivalent.

Examples

```
# Power with unequal group sizes (n2 = 2 * n1)
power.p2s.test(n = 100, p1 = 0.3, p2 = 0.5, group.rate = 2)

# Required sample size in group 1
power.p2s.test(p1 = 0.3, p2 = 0.5, power = 0.8, group.rate = 1.5)
```

rate.1s.ci

*Confidence Intervals for a One-Sample Poisson Rate***Description**

Computes confidence intervals for a Poisson rate parameter $\lambda = x/T$ using several methods.

Usage

```
rate.1s.ci(
  x,
  T = 1,
  conf.level = 0.95,
  method = c("exact", "score", "wh", "wald", "log"),
  correct = TRUE,
  ...
)
```

Arguments

x	Non-negative integer. Observed number of events.
T	Positive numeric. Exposure time (or total time at risk).
conf.level	Confidence level. Default is 0.95.
method	Method for confidence interval. One of: • "exact": Exact (Garwood) interval • "score": Score interval (inversion of score test) • "wh": Wilson-Hilferty interval • "wald": Wald interval (normal approximation)

	• "log": Log-transformed interval
correct	Logical. Apply continuity correction for "score", "wh", and "wald" methods. Default is TRUE.
...	Reserved for future extensions.

Details

This function provides several confidence intervals for the Poisson rate $\lambda = x/T$.

Exact (Garwood): Based on inversion of the Poisson test using chi-square quantiles:

$$\left[\frac{1}{2T} \chi_{2x, \alpha/2}^2, \frac{1}{2T} \chi_{2(x+1), 1-\alpha/2}^2 \right].$$

Score: Obtained by inverting the score test. Computed from solving quadratic equations for boundaries.

Wilson-Hilferty: The Wilson–Hilferty (WH) interval is based on a cube-root transformation of a chi-square approximation, which reduces skewness and yields an approximately normal pivot, leading to the cubic form of the limits. The lower limit is

$$\frac{x}{T} \left(1 - \frac{1}{9x} + \frac{z_{\alpha/2}}{3\sqrt{x}} \right)^3$$

and the upper limit is

$$\frac{x+1}{T} \left(1 - \frac{1}{9(x+1)} + \frac{z_{1-\alpha/2}}{3\sqrt{x+1}} \right)^3.$$

Wald: For $x > 0$,

$$\hat{\lambda} \pm z_{\alpha/2} \sqrt{\hat{\lambda}/T}.$$

For $x = 0$, lower limit 0, upper limit $-\log(\alpha/2)/T$.

Log: For $x > 0$, applies normal approximation to $\log(\lambda)$ and transforms back. For $x = 0$, lower limit 0, upper limit $-\log(\alpha/2)/T$.

When `correct = TRUE`, continuity correction is applied on the count scale for methods that support it.

Value

An object of class "htest".

Examples

```
rate.ls.ci(5, 10)
rate.ls.ci(5, 10, method = "score")
rate.ls.ci(0, 10, method = "exact")
```

rate.2s.ci

*Confidence Intervals for a Two-Sample Poisson Rate Ratio***Description**

Computes confidence intervals for the rate ratio $\rho = \lambda_1/\lambda_2$ from two independent Poisson counts with known exposures.

Usage

```
rate.2s.ci(
  x,
  T = c(1, 1),
  conf.level = 0.95,
  method = c("log", "score", "exact"),
  ...
)
```

Arguments

x	Length-2 vector of non-negative integer event counts.
T	Length-2 positive numeric vector of exposures.
conf.level	Confidence level. Default is 0.95.
method	Method for confidence interval. One of: • "log": Log-Wald interval • "score": Transformed Wilson score interval • "exact": Transformed exact Clopper-Pearson interval
...	Reserved for future extensions.

Details

For two independent Poisson variables $X_i \sim \text{Pois}(\lambda_i T_i)$, the point estimate of the rate ratio is

$$\hat{\rho} = \frac{X_1/T_1}{X_2/T_2}.$$

The "log" method uses the large-sample interval

$$\log(\hat{\rho}) \pm z_{1-\alpha/2} \sqrt{1/X_1 + 1/X_2}$$

and exponentiates the endpoints. This method requires both counts to be positive.

The "score" and "exact" methods use the conditional binomial representation. Given $n = X_1 + X_2$, $X_1 \sim \text{Binom}(n, \pi)$, where

$$\pi = \frac{T_1 \rho}{T_1 \rho + T_2}.$$

Confidence limits L and U for π are transformed to confidence limits for ρ as

$$\left(\frac{LT_2}{(1-L)T_1}, \frac{UT_2}{(1-U)T_1} \right).$$

The "score" method uses Wilson score limits for π ; the "exact" method uses the exact Clopper-Pearson limits returned by `binom.test`.

Value

An object of class "htest".

Examples

```
rate.2s.ci(c(151, 55), T = c(57518.1, 74573.5))
rate.2s.ci(c(151, 55), T = c(57518.1, 74573.5), method = "score")
rate.2s.ci(c(9, 12), T = c(1817.6, 7496.3), method = "exact")
```

rate.test

Test of Poisson Rates Using Normal Approximation

Description

Performs a large-sample (normal) test for one or two Poisson rates with known exposures. The test is carried out on the observed count scale and then translated to rates for reporting.

Usage

```
rate.test(
  x,
  T = 1,
  r = 1,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  correct = TRUE
)
```

Arguments

x	a vector of event counts. A single value specifies a one-sample test; a vector of length two specifies a two-sample comparison.
T	a vector of exposures corresponding to x (e.g., person-time). Must have the same length as x.
r	a positive number specifying the null rate per unit exposure, λ_0 , for a one-sample test. Default to 1.0. Ignored for two-sample tests.
alternative	a character string specifying the alternative hypothesis, one of "two.sided", "less", or "greater".

conf.level	confidence level for the confidence interval. For one-sample tests, the interval is the score confidence interval for the rate. For two-sample tests, the interval is the log-Wald confidence interval for the rate ratio.
correct	logical; if TRUE, applies a continuity correction on the count scale. For one-sample tests, a ± 0.5 correction is applied to x in the direction determined by alternative. For "two.sided", the correction uses the sign of $x - \mu_0$.

Details

One-sample test: Assume $X \sim \text{Pois}(\mu)$ with $\mu = \lambda T$. The null hypothesis is $H_0 : \lambda = \lambda_0$, where $\lambda_0 = r$. Let $\mu_0 = \lambda_0 T$. The normal approximation gives

$$Z = \frac{(x - c) - \mu_0}{\sqrt{\mu_0}} \approx N(0, 1),$$

where c is 0 if `correct = FALSE` and is a continuity correction on the count scale if `correct = TRUE`. Two-sided p-values use $2\{1 - \Phi(|Z|)\}$.

Two-sample test: Assume independent $X_1 \sim \text{Pois}(\lambda_1 T_1)$ and $X_2 \sim \text{Pois}(\lambda_2 T_2)$ with known exposures T_1 and T_2 . The null hypothesis is $H_0 : \lambda_1 = \lambda_2$.

For hypothesis testing, the function uses the exact conditional representation under H_0 :

$$X_1 \mid (X_1 + X_2 = n) \sim \text{Binom}\left(n, \frac{T_1}{T_1 + T_2}\right),$$

and applies the same large-sample normal approximation as `prop.test` (with optional Yates continuity correction) to obtain the test statistic and p-value.

For estimation and confidence intervals, the inference target is the rate ratio, $\rho = \lambda_1/\lambda_2$. The point estimate is $\hat{\rho} = (X_1/T_1)/(X_2/T_2)$, and the confidence interval is constructed on the log scale using estimated standard error

$$\sqrt{1/X_1 + 1/X_2}.$$

The continuity correction affects the hypothesis test but not the confidence interval for ρ .

Confidence intervals: For one-sample tests, the confidence interval is the score interval for λ , as computed by `rate.ls.ci` with `method = "score"`. For two-sample tests, the confidence interval is for the rate ratio λ_1/λ_2 . The log-Wald interval requires both event counts to be positive.

Value

An object of class "htest" containing:

statistic	the standardized normal statistic z for one-sample tests or the chi-square statistic from <code>prop.test</code> for two-sample tests.
parameter	degrees of freedom ($df = 1$).
p.value	the p-value.
conf.int	a confidence interval for the rate in one-sample tests or for the rate ratio, λ_1/λ_2 , in two-sample tests.
estimate	estimated rate (one-sample) or estimated rates and rate ratio (two-sample).
null.value	the null rate (one-sample) or the null rate ratio $\lambda_1/\lambda_2 = 1$ (two-sample).

alternative the alternative hypothesis.
method a character string describing the test.
data.name a character string describing the data.

See Also

[rate.2s.ci](#), [prop.test](#), [poisson.test](#)

Examples

```
## One-sample test: compare observed rate to a reference unit rate
rate.test(x = 411, T = 25800, r = 0.0119)
rate.test(x = 411, T = 25800, r = 0.0119, correct = FALSE)

## Two-sample test: compare two Poisson rates
rate.test(x = c(12, 5), T = c(100, 80))
rate.test(x = c(12, 5), T = c(100, 80), correct = FALSE)

## One-sided alternative
rate.test(x = 411, T = 25800, r = 0.0119, alternative = "greater")
```

rds

Neonatal Respiratory Distress Syndrome Data

Description

Aggregated counts of neonatal outcomes by birth weight, surfactant use, and survival status.

Usage

rds

Format

A data frame with 16 rows and 4 variables:

bwt Birth weight category (factor with 4 levels).

surf Surfactant use (Yes/No).

death Outcome (1 = death; 0 = alive).

count Number of infants in each group.

Details

The data are presented in aggregated form. Each row corresponds to a combination of birth weight category, surfactant use, and outcome, with count indicating the number of observations.

Analyses should account for the grouped structure, e.g., via weighted models or binomial responses.

Source

Classical neonatal RDS dataset (exact source to be specified).

wang.paired.ci

Wang Exact Confidence Interval for Paired Proportions

Description

Computes Wang's exact inductive confidence interval for the paired risk difference $p_{10} - p_{01}$ from a paired binary table.

Usage

```
wang.paired.ci(
  n10,
  t,
  n01,
  conf.level = 0.95,
  CItypes = c("Two.sided", "Lower", "Upper"),
  precision = 1e-05,
  grid.one = 30,
  grid.two = 20
)
```

Arguments

n10	Number of pairs with success under treatment and failure under control.
t	Number of concordant pairs, $n_{11} + n_{00}$.
n01	Number of pairs with failure under treatment and success under control.
conf.level	Confidence level.
CItypes	Type of interval: "Two.sided", "Lower", or "Upper".
precision	Numerical precision for confidence limits.
grid.one	Number of grid points in the first nuisance-parameter search.
grid.two	Number of grid points in the second nuisance-parameter search.

Value

A list with elements conf.level, CItypes, estimate, and ExactCI.

References

Wang, W. (2012). An inductive order construction for the difference of two dependent proportions. *Statistics & Probability Letters*, 82, 1623–1628.

Examples

```
wang.paired.ci(3, 1, 0, CItpe = "Lower")
wang.paired.ci(3, 1, 0, conf.level = 0.9)
```

wang.paired.reject	<i>Wang Interval-Inversion Test for Enumerated Paired Tables</i>
--------------------	--

Description

Evaluates whether Wang's exact paired confidence interval excludes zero for each row of an enumerated paired-table data frame.

Usage

```
wang.paired.reject(
  tabs,
  alpha = 0.05,
  precision = 1e-05,
  grid.one = 30,
  grid.two = 20
)
```

Arguments

tabs	A data frame with columns n11, n10, n01, and n00.
alpha	Test size; the confidence level is $1 - \alpha$.
precision	Numerical precision for confidence limits.
grid.one	Number of grid points in the first nuisance-parameter search.
grid.two	Number of grid points in the second nuisance-parameter search.

Value

A logical vector indicating whether each row rejects $H_0 : p_{10} - p_{01} = 0$.

Examples

```
tabs <- data.frame(n11 = c(0, 1), n10 = c(3, 1), n01 = c(0, 1), n00 = c(1, 1))
wang.paired.reject(tabs, alpha = 0.05, precision = 0.0001)
```

Index

* datasets

- depress, [3](#)
- fluorosis, [5](#)
- midbp, [8](#)
- mihdl, [8](#)
- nrs, [9](#)
- rds, [19](#)

binom.test, [17](#)

chisq.test, [7](#)

demo_clt, [2](#)

depress, [3](#)

fluorosis, [5](#)

glm, [7](#)

logistic_gof, [6](#)

midbp, [8](#)

mihdl, [8](#)

nrs, [9](#)

poisson.test, [19](#)

power.p1s.test, [10](#)

power.p2s.test, [12](#)

prop.test, [18](#), [19](#)

rate.1s.ci, [14](#), [18](#)

rate.2s.ci, [16](#), [19](#)

rate.test, [17](#)

rds, [19](#)

wang.paired.ci, [20](#)

wang.paired.reject, [21](#)