Package 'rashnu'

July 23, 2025

Title Balanced Sample Size and Power Calculation Tools

Version 0.1.2

Description Implements sample size and power calculation methods with a focus on balance and fairness in study design, inspired by the Zoroastrian deity Rashnu, the judge who weighs truth. Supports survival analysis and various hypothesis testing frameworks.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports shiny, DT, stats

Suggests covr, testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://zarathucorp.github.io/rashnu/, https://github.com/zarathucorp/rashnu

BugReports https://github.com/zarathucorp/rashnu/issues

NeedsCompilation no

Author Sungho Choi [aut], Gyeom Hwangbo [ctb, cre], Zarathu [cph, fnd]

Maintainer Gyeom Hwangbo <hbgyeom@gmail.com>

Repository CRAN

Date/Publication 2025-07-23 08:00:02 UTC

Contents

`coxph_size` *Sample Size or Power for Cox Proportional Hazards*

Description

Calculates sample size or power for a cox proportional hazards model.

Usage

```
coxph_size(
  hr,
  hr0 = NULL,
  delta = NULL,
  pE,
  pA,
  alpha,
  beta = NULL,
  n = NULL,
  test_type = "2-side"
)
```

Arguments

`hr`	Numeric. True hazard ratio.
`hr0`	Numeric (optional). Null hypothesis hazard ratio. Required for `"2-side"`, `"non-inferiority"` test.
`delta`	Numeric (optional). Margin for `"equivalence test"`. Required for `"equivalence"` test.
`pE`	Numeric. Overall event probability.
`pA`	Numeric. Proportion of group A.
`alpha`	Numeric. Type I error rate.
`beta`	Numeric (optional). Type II error rate. Required for sample size calculation.
`n`	Integer (optional). Sample size. Required for power calculation.
`test_type`	Character. `"2-side"`, `"non-inferiority"`, or `"equivalence"`. Default is `"2-side"`.

Value

Numeric. Returns sample size (if `beta` is given), or power (if `n` is given).

Note

Only one of `beta` (for sample size calculation) or `n` (for power calculation) should be specified.

Required arguments by `test_type`:

- `"2-side"`/`"non-inferiority"`:
  - For sample size: `hr`, `hr0`, `pE`, `pA`, `alpha`, `beta`
  - For power: `hr`, `hr0`, `pE`, `pA`, `alpha`, `n`
- `"equivalence"`:
  - For sample size: `hr`, `delta`, `pE`, `pA`, `alpha`, `beta`
  - For power: `hr`, `delta`, `pE`, `pA`, `alpha`, `n`

Examples

```
# Sample size for a `"2-side"` test
coxph_size(hr = 2, hr0 = 1, pE = 0.8, pA = 0.5,
           alpha = 0.05, beta = 0.2, test_type = "2-side")

# Power of `"2-side"` test
coxph_size(hr = 2, hr0 = 1, pE = 0.8, pA = 0.5,
           alpha = 0.05, n = 82, test_type = "2-side")

# Sample size for `"non-inferiority"` test
coxph_size(hr = 2, hr0 = 1, pE = 0.8, pA = 0.5,
           alpha = 0.025, beta = 0.2, test_type = "non-inferiority")

# Power of `"non-inferiority"` test
coxph_size(hr = 2, hr0 = 1, pE = 0.8, pA = 0.5,
           alpha = 0.025, n = 82, test_type = "non-inferiority")

# Sample size for `"equivalence"` test
coxph_size(hr = 1, delta = 0.5, pE = 0.8, pA = 0.5,
           alpha = 0.05, beta = 0.2, test_type = "equivalence")

# Power of `"equivalence"` test
coxph_size(hr = 1, delta = 0.5, pE = 0.8, pA = 0.5,
           alpha = 0.05, n = 172, test_type = "equivalence")
```

---

k_mean_size *Sample Size or Power Calculation for K Means*

Description

Calculates sample size or power for a multiple-sample mean test.

Usage

```
k_mean_size(
  muA,
  muB,
  kappa = 1,
  sd = NULL,
  sdA = NULL,
  sdB = NULL,
  tau = 1,
  alpha,
  beta = NULL,
  n = NULL,
  nA = NULL,
  test_type = "2-side"
)
```

Arguments

muA	Numeric. True mean of group A.
muB	Numeric. True mean of group B.
kappa	Numeric. Ratio of sample sizes (nA/nB). Default is 1.
sd	Numeric (optional). Standard deviation. Required for "2-side" test.
sdA	Numeric (optional). Standard deviation of group A. Required for "1-side" test.
sdB	Numeric (optional). Standard deviation of group B. Required for "1-side" test.
tau	Integer. Number of comparisons.
alpha	Numeric. Type I error rate.
beta	Numeric (optional). Type II error rate (1 - power). Required for sample size calculation.
n	Integer (optional). Sample size. Required for power calculation of "2-side" test.
nA	Integer (optional). Sample size of group A. Required for power calculation of "1-side" test.
test_type	Character. "2-side" or "1-side". Default is "2-side"

Value

Numeric. Returns sample size (if beta is given), or power (if n/nA is given).

Note

Only one of beta (for sample size calculation) or n/nA (for power calculation) should be specified.

Required arguments by test_type:

- "2-side":
  - For sample size: muA, muB, sd, tau, alpha, beta

  - For power: muA, muB, sd, tau, alpha, n

-"1-side":

- For sample size: muA, muB, sdA, sdB, tau, alpha, beta
- For power: muA, muB, sdA, sdB, tau, alpha, nA

Examples

```
# Sample size for `"2-side"` test
k_mean_size(muA = 5, muB = 10, sd = 10, tau = 1,
            alpha = 0.05, beta = 0.2, test_type = "2-side")

# Power of `"2-side"` test
k_mean_size(muA = 5, muB = 10, sd = 10, tau = 1,
            alpha = 0.05, n = 63, test_type = "2-side")

# Sample size for `"1-side"` test
k_mean_size(muA = 132.86, muB = 127.44, kappa = 2, sdA = 15.34, sdB = 18.23, tau = 1,
            alpha = 0.05, beta = 0.2, test_type = "1-side")

# Power of `"1-side"` test
k_mean_size(muA = 132.86, muB = 127.44, kappa = 2, sdA = 15.34, sdB = 18.23, tau = 1,
            alpha = 0.05, nA = 85, test_type = "1-side")
```

---

k_prop_size *Sample Size or Power Calculation for K proportion*

Description

Calculates sample size or power for a multiple-sample proportion test.

Usage

```
k_prop_size(pA, pB, tau, alpha, beta = NULL, n = NULL)
```

Arguments

pA	Numeric. True proportion of group A.
pB	Numeric. True proportion of group B.
tau	Integer. Number of comparisons.
alpha	Numeric. Type I error rate.
beta	Numeric (optional). Type II error rate. Required for sample size calculation.
n	Integer (optional). Sample size. Required when calculating power.

Value

Returns sample size (if beta is given), or power (if n is given).

Note

Only one of `beta` (for sample size calculation) or `n` (for power calculation) should be specified.

Required arguments:

- For sample size: `pA`, `pB`, `tau`, `alpha`, `beta`
- For power: `pA`, `pB`, `tau`, `alpha`, `n`

Examples

```
# Sample size
k_prop_size(pA = 0.2, pB = 0.4, tau = 2,
            alpha = 0.05, beta = 0.2)

# Power
k_prop_size(pA = 0.2, pB = 0.4, tau = 2,
            alpha = 0.05, n = 96)
```

---

lakatosSampleSize *Sample Size Calculation Using Lakatos Method for Survival Analysis*

Description

Computes the required sample size and expected event numbers for two-group survival analysis using Lakatos' method under exponential survival assumptions and varying weight functions (log-rank, Gehan, Tarone-Ware).

Usage

```
lakatosSampleSize(
  syear,
  yrsurv1,
  yrsurv2,
  alloc,
  accrualTime,
  followTime,
  alpha,
  power,
  method = c("logrank", "gehan", "tarone-ware"),
  side = c("two.sided", "one.sided"),
  b = 24
)
```

Arguments

`syear`	Survival time horizon in years.
`yrsurv1`	Survival probability of the standard group at `syear`.
`yrsurv2`	Survival probability of the test group at `syear`.
`alloc`	Allocation ratio (Test / Standard). For equal allocation, use 1.
`accrualTime`	Accrual period duration.
`followTime`	Additional follow-up time after last patient is accrued.
`alpha`	Significance level (e.g., 0.05 for two-sided tests).
`power`	Desired statistical power (e.g., 0.8).
`method`	Weighting method for test statistic. One of `"logrank"`, `"gehan"`, or `"tarone-ware"`.
`side`	Type of test: `"two.sided"` or `"one.sided"`.
`b`	Number of time divisions per year for numerical integration (default = 24).

Value

A list containing:

Sample_size_of_standard_group Required sample size in the standard group.

Sample_size_of_test_group Required sample size in the test group.

Total_sample_size Total sample size.

Expected_event_numbers_of_standard_group Expected number of events in the standard group.

Expected_event_numbers_of_test_group Expected number of events in the test group.

Total_expected_event_numbers Total number of expected events.

Actual_power Achieved power given the calculated sample size.

error (Optional) Error message when sample size cannot be calculated.

References

Lakatos E. (1988). Sample sizes based on the log-rank statistic in complex clinical trials. *Biometrics*, 44, 229–241.

Lakatos E, Lan KK. (1992). A comparison of sample size methods for the logrank statistic. *Statistics in Medicine*, 11(2), 179–191.

Web calculator (Superiority): https://nshi.jp/en/js/twosurvyr/

Examples

```
lakatosSampleSize(
  syear = 2,
  yrsurv1 = 0.7,
  yrsurv2 = 0.6,
  alloc = 1,
  accrualTime = 1,
  followTime = 1,
  alpha = 0.05,
```

```
  power = 0.8,
  method = "logrank",
  side = "two.sided"
)
```

`oneSurvSampleSize` *One-Sample Survival Study Sample Size or Power Calculation*

Description

Calculates the required sample size or power for a single-arm survival study using various transformation-based methods, including arcsine-square root, log-log, logit, and others. This function assumes an exponential survival model.

Usage

```
oneSurvSampleSize(
  survTime,
  p1,
  p2,
  accrualTime,
  followTime,
  alpha,
  power,
  side = c("two.sided", "one.sided"),
  method = c("arcsin", "log-log", "logit", "log", "log-swog", "identity")
)
```

Arguments

`survTime`	Time point at which survival is evaluated (e.g., median follow-up time).
`p1`	Expected survival probability under the alternative hypothesis.
`p2`	Survival probability under the null hypothesis.
`accrualTime`	Patient accrual period.
`followTime`	Additional follow-up period after accrual ends.
`alpha`	Significance level (e.g., 0.05).
`power`	Desired statistical power (e.g., 0.8).
`side`	Type of hypothesis test. Either `"two.sided"` (default) or `"one.sided"`.
`method`	Transformation method for comparison. One of `"arcsin"`, `"log-log"`, `"logit"`, `"log"`, `"log-swog"`, `"identity"`.

Value

A named numeric vector with:

SampleSize Calculated required sample size.

Power Achieved power with the calculated sample size.

References

Fleming TR, Harrington DP. (1991). *Counting Processes and Survival Analysis*. New York: Wiley, pp. 236–237, Example 6.3.1.

Andersen PK, Borgan O, Gill RD, Keiding N. (1993). *Statistical Models Based on Counting Processes*. New York: Springer-Verlag, pp. 176–287, Section IV.1–3.

Bie O, Borgan O, Liestol K. (1987). Confidence intervals and confidence bands for the cumulative hazard rate function and their small sample properties. *Scandinavian Journal of Statistics*, 14(3), 221–233.

Borgan O, Liestol K. (1990). A note on confidence intervals and bands for the survival function based on transformations. *Scandinavian Journal of Statistics*, 17(1), 35–41.

Nagashima K, Noma H, Sato Y, Gosho M. (2020). Sample size calculations for single-arm survival studies using transformations of the Kaplan–Meier estimator. *Pharmaceutical Statistics*. https://doi.org/10.1002/pst.2090 Available at: https://arxiv.org/abs/2012.03355

Web calculator (One-sample): https://nshi.jp/en/js/onesurvyr/

Examples

```
oneSurvSampleSize(
  survTime = 2,
  p1 = 0.75,
  p2 = 0.6,
  accrualTime = 1,
  followTime = 1,
  alpha = 0.05,
  power = 0.8,
  side = "two.sided",
  method = "log-log"
)
```

---

one_bino_size *Sample Size or Power for One-Sample Binomial Proportion Test*

Description

Calculates sample size or power for a two-sample binomial proportion test.

Usage

```
one_bino_size(p, p0, alpha, beta = NULL, n = NULL)
```

Arguments

p	Numeric. True proportion.
p0	Numeric. Null hypothesis proportion.
alpha	Numeric. Type I error rate.
beta	Numeric (optional). Type II error rate. Required for sample size calculation.
n	Integer (optional). Sample size. Required for power calculation.

Value

Numeric. Returns sample size (if beta is given), or power (if n is given).

Note

Only one of beta (for sample size calculation) or n (for power calculation) should be specified.

Required arguments:

- For sample size: "p", "p0", "alpha", "beta"
- For power: "p", "p0", "alpha", "n"

Examples

```
# Required sample size
one_bino_size(p = 0.5, p0 = 0.3,
              alpha = 0.05, beta = 0.2)

# Power
one_bino_size(p = 0.5, p0 = 0.3,
              alpha = 0.05, n = 50)
```

---

one_mean_size *Sample Size or Power for One-Sample Mean Test*

Description

Calculates sample size or power for a one-sample mean test.

Usage

```
one_mean_size(
  mu,
  mu0,
  delta = NULL,
  sd,
  alpha,
  beta = NULL,
  n = NULL,
  test_type = "2-side"
)
```

Arguments

mu	Numeric. True mean.
mu0	Numeric. Null hypothesis mean.
delta	Numeric (optional). Margin for "non-inferiority" or "equivalence" test. Required for "non-inferiority" or "equivalence" test.
sd	Numeric. Standard deviation.
alpha	Numeric. Type I error rate.
beta	Numeric (optional). Type II error rate. Required for sample size calculation.
n	Integer (optional). Sample size. Required for power calculation.
test_type	Character. "2-side", "1-side", "non-inferiority", or "equivalence". Default is "2-side".

Value

Numeric. Returns sample size (if beta is given), or power (if n is given).

Note

Only one of beta (for sample size calculation) or n (for power calculation) should be specified.

Required arguments by test_type:

- "2-side" / "1-side":
  - For sample size: mu, mu0, sd, alpha, beta
  - For power: mu, mu0, sd, alpha, n
- "non-inferiority" / "equivalence":
  - For sample size: mu, mu0, delta, sd, alpha, beta
  - For power: mu, mu0, delta, sd, alpha, n

Examples

```
# Sample size for `"2-side"` test
one_mean_size(mu = 2, mu0 = 1.5, sd = 1,
              alpha = 0.05, beta = 0.2, test_type = "2-side")

# Power of `"2-side"` test
one_mean_size(mu = 2, mu0 = 1.5, sd = 1,
              alpha = 0.05, n = 32, test_type = "2-side")

# Sample size for `"1-side"` test
one_mean_size(mu = 115, mu0 = 120, sd = 24,
              alpha = 0.05, beta = 0.2, test_type = "1-side")

# Power of `"1-side"` test
one_mean_size(mu = 115, mu0 = 120, sd = 24,
              alpha = 0.05, n = 143, test_type = "1-side")

# Sample size for `"non-inferiority"` test
one_mean_size(mu = 2, mu0 = 1.5, delta = -0.5, sd = 1,
              alpha = 0.05, beta = 0.2, test_type = "non-inferiority")

# Power of `"non-inferiority"` test
one_mean_size(mu = 2, mu0 = 1.5, delta = -0.5, sd = 1,
              alpha = 0.05, n = 7, test_type = "non-inferiority")

# Sample size for `"equivalence"` test
one_mean_size(mu = 2, mu0 = 2, delta = 0.05, sd = 0.1,
              alpha = 0.05, beta = 0.2, test_type = "equivalence")

# Power of `"equivalence"` test
one_mean_size(mu = 2, mu0 = 2, delta = 0.05, sd = 0.1,
              alpha = 0.05, n = 35, test_type = "equivalence")
```

`one_norm_size` *Sample Size or Power Calculation for One-Sample Normal Mean Test*

Description

Calculates sample size or power for a two-sample normal mean test.

Usage

```
one_norm_size(mu, mu0, sd, alpha, beta = NULL, n = NULL)
```

Arguments

`mu`	Numeric. True mean.
`mu0`	Numeric. Null hypothesis mean.

sd	Numeric. Standard deviation.
alpha	Numeric. Type I error rate.
beta	Numeric (optional). Type II error rate. Required for sample size calculation.
n	Integer (optional). Sample size. Required for power calculation.

Value

Numeric. Returns sample size (if beta is given), or power (if n is given).

Note

Only one of beta (for sample size calculation) or n (for power calculation) should be specified.

Required arguments:

- For sample size: "mu", "mu0", "sd", "alpha", "beta"
- For power: "mu", "mu0", "sd", "alpha", "n"

Examples

```
# Sample size
one_norm_size(mu = 2, mu0 = 1.5, sd = 1,
              alpha = 0.05, beta = 0.2)

# Power
one_norm_size(mu = 2, mu0 = 1.5, sd = 1,
              alpha = 0.05, n = 32)
```

---

one_prop_size *Sample Size or Power for One-Sample Proportion Test*

Description

Calculates sample size or power for a one-sample proportion test.

Usage

```
one_prop_size(
  p,
  p0,
  delta = NULL,
  alpha,
  beta = NULL,
  n = NULL,
  test_type = "2-side"
)
```

Arguments

p	Numeric. True proportion.
p0	Numeric. Null hypothesis proportion.
delta	Numeric (optional). Margin for "non-inferiority" or "equivalence" test. Required for "non-inferiority" or "equivalence" test.
alpha	Numeric. Type I error rate.
beta	Numeric (optional). Type II error rate. Required for sample size calculation.
n	Integer (optional). Sample size. Required for power calculation.
test_type	Character. "2-side", "1-side", "non-inferiority", or "equivalence". Default is "2-side".

Value

Numeric. Returns sample size (if beta is given), or power (if n is given).

Note

Only one of beta (for sample size calculation) or n (for power calculation) should be specified.

Required arguments by test_type:

- "2-side"/"1-side":
  - For sample size: p, p0, alpha, beta
  - For power: p, p0, alpha, n
- "non-inferiority"/"equivalence":
  - For sample size: p, p0, delta, alpha, beta
  - For power: p, p0, sdA, delta, alpha, n

Examples

```
# Sample size for `"2-side"` test
one_prop_size(p = 0.5, p0 = 0.3,
              alpha = 0.05, beta = 0.2, test_type = "2-side")

# Power of `"2-side"` test
one_prop_size(p = 0.5, p0 = 0.3,
              alpha = 0.05, n = 50, test_type = "2-side")

# Sample size for `"1-side"` test
one_prop_size(p = 0.05, p0 = 0.02,
              alpha = 0.05, beta = 0.2, test_type = "1-side")

# Power of `"1-sided"` test
one_prop_size(p = 0.05, p0 = 0.02,
              alpha = 0.05, n = 191, test_type = "1-side")

# Sample size for `"non-inferiority"` test
one_prop_size(p = 0.5, p0 = 0.3, delta = -0.1,
```

```
              alpha = 0.05, beta = 0.2, test_type = "non-inferiority")

# Power of `"non-inferiority"` test
one_prop_size(p = 0.5, p0 = 0.3, delta = -0.1,
              alpha = 0.05, n = 18, test_type = "non-inferiority")

# Sample size for `"equivalence"` test
one_prop_size(p = 0.6, p0 = 0.6, delta = 0.2,
              alpha = 0.05, beta = 0.2, test_type = "equivalence")

# Power of `"equivalence"` test
one_prop_size(p = 0.6, p0 = 0.6, delta = 0.2,
              alpha = 0.05, n = 52, test_type = "equivalence")
```

`or_size` *Sample Size or Power for Odds Ratio Test*

Description

Calculates sample size or power for odds ratio test.

Usage

```
or_size(
  pA,
  pB,
  delta = NULL,
  kappa = 1,
  alpha,
  beta = NULL,
  nB = NULL,
  test_type = "equality"
)
```

Arguments

`pA`	Numeric. True proportion of group A.
`pB`	Numeric. True proportion of group B.
`delta`	Numeric (optional). Margin for `"non-inferiority"` or `"equivalence"` test. Required for `"non-inferiority"` or `"equivalence"` test.
`kappa`	Numeric. Ratio of sample sizes (nA/nB). Default is 1.
`alpha`	Numeric. Type I error rate.
`beta`	Numeric (optional). Type II error rate. Required for sample size calculation.
`nB`	Integer (optional). Sample size for group B. Required for power calculation.
`test_type`	Character. `"equality"`, `"non-inferiority"`, or `"equivalence"`. Default is `"2-side"`.

Value

Numeric. Returns sample size (if beta is given), or power (if nB is given).

Note

Only one of beta (for sample size calculation) or nB (for power calculation) should be specified.

Required arguments by test_type:

- "equality":
  - For sample size: pA, pB, alpha, beta
  - For power: pA, pB, alpha, nB
- "non-inferiority"/"equivalence":
  - For sample size: pA, pB, delta, alpha, beta
  - For power: pA, pB, delta, alpha, nB

Examples

```
# Sample size for `"equality"` test
or_size(pA = 0.4, pB = 0.25, kappa = 1,
        alpha = 0.05, beta = 0.2, test_type = "equality")

# Power of `"equality"` test
or_size(pA = 0.4, pB = 0.25, kappa = 1,
        alpha = 0.05, nB = 156, test_type = "equality")

# Sample size for `"non-inferiority"` test
or_size(pA = 0.4, pB = 0.25, delta = 0.2, kappa = 1,
        alpha = 0.05, beta = 0.2, test_type = "non-inferiority")

# Power of `"non-inferiority"` test
or_size(pA = 0.4, pB = 0.25, delta = 0.2, kappa = 1,
        alpha = 0.05, nB = 242, test_type = "non-inferiority")

# Sample size for `"equivalence"` test
or_size(pA = 0.25, pB = 0.25, delta = 0.5, kappa = 1,
        alpha = 0.05, beta = 0.2, test_type = "equivalence")

# Power of `"equivalence"` test
or_size(pA = 0.25, pB = 0.25, delta = 0.5, kappa = 1,
        alpha = 0.05, nB = 366, test_type = "equivalence")
```

---

pair_prop_size *Sample Size or Power for Paired-Sample Proportion Test*

Description

Calculates sample size or power for a paired-sample proportion test.

Usage

```
pair_prop_size(p01, p10, alpha, beta = NULL, n = NULL, test_type = "2-side")
```

Arguments

p01	Numeric. Proportion of discordant pairs with (before = 1, after = 0).
p10	Numeric. Proportion of discordant pairs with (before = 0, after = 1).
alpha	Numeric. Type I error rate.
beta	Numeric (optional). Type II error rate. Required for sample size calculation.
n	Integer (optional). Sample size. Required for power calculation.
test_type	Character. "2-side" or "1-side". Default is "2-side".

Value

Numeric. Returns sample size (if beta is given), or power (if n is given).

Note

Only one of beta (for sample size calculation) or n (for power calculation) should be specified.

Required arguments:

- For sample size: p01, p10, alpha, beta
- For power: p01, p10, alpha, n

Examples

```
# Sample size for `"2-side"` test
pair_prop_size(p01 = 0.45, p10 = 0.05,
               alpha = 0.1, beta = 0.1, test_type = "2-side")

# Power of `"2-side"` test
pair_prop_size(p01 = 0.45, p10 = 0.05,
               alpha = 0.1, n = 23, test_type = "2-side")

# Sample size for `"1-side"` test
pair_prop_size(p01 = 0.45, p10 = 0.05,
               alpha = 0.05, beta = 0.1, test_type = "1-side")

# Power of `"1-side"` test
pair_prop_size(p01 = 0.45, p10 = 0.05,
               alpha = 0.05, n = 23, test_type = "1-side")
```

`rashnuBasic` *Interactive Sample Size Calculator for Survival Studies (Shiny App)*

Description

Launches a Shiny web application that calculates required sample sizes and expected event numbers for different types of survival analysis designs:

- Two-group Non-Inferiority
- Two-group Superiority (Lakatos method)
- One-sample survival test (with transformation methods)

Usage

```
rashnuBasic()
```

Details

Users can specify survival probabilities, accrual and follow-up durations, allocation ratios, non-inferiority margins, transformation methods, and test types. The app dynamically adjusts input UI based on the selected design and displays results in a data table format.

Test Types:

- `"ni"` - Non-Inferiority (two-group exponential survival comparison)
- `"sup"` - Superiority (Lakatos method with logrank/Gehan/Tarone-Ware weighting)
- `"one"` - One-sample survival test with multiple transformation options

Included References:

- Jung SH, Chow SC. Journal of Biopharmaceutical Statistics, 2012.
- Lakatos E. Biometrics, 1988.
- Lakatos & Lan. Statistics in Medicine, 1992.
- Fleming & Harrington. Counting Processes and Survival Analysis, 1991.
- Borgan 0, Andersen PK et al. Springer-Verlag, 1993.
- Nagashima et al. Pharmaceutical Statistics, 2020.

Value

Launches a Shiny app in the default browser.

Note

Requires associated functions `twoSurvSampleSizeNI()`, `lakatosSampleSize()`, and `oneSurvSampleSize()` to be defined in the environment. Assumes a CSS file is available at `"www/style.css"` for custom styling.

Examples

```
if (interactive()) {
  rashnuBasic()
}
```

`sccs_size` *Sample Size or Power for Self-Controlled Case Series (SCCS)*

Description

Calculates sample size or power for self-controlled case series studies.

Usage

```
sccs_size(p, r, alpha, beta = NULL, n = NULL)
```

Arguments

`p`	Numeric. True relative incidence (risk period vs baseline).
`r`	Numeric. Proportion of observation time that is risk period.
`alpha`	Numeric. Type I error rate.
`beta`	Numeric (optional). Type II error rate. Required for sample size calculation.
`n`	Integer (optional). Sample size. Required for power calculation.

Value

Numeric. Returns sample size (if `beta` is given), or power (if `n` is given).

Note

Only one of `beta` (for sample size calculation) or `n` (for power calculation) should be specified.

Required arguments:

- For sample size: `p`, `r`, `alpha`, `beta`
- For power: `p`, `r`, `alpha`, `n`

Examples

```
# Sample size
sccs_size(p = 3, r = 42/365,
          alpha = 0.05, beta = 0.2)

# Power
sccs_size(p = 3, r = 42/365,
          alpha = 0.05, n = 54)
```

`twoSurvSampleSizeNI` *Sample Size Calculation for Two-Group Non-Inferiority Survival Study*

Description

Calculates the required sample size and expected event numbers for a non-inferiority trial with two survival curves, using piecewise integration of hazard functions under exponential survival assumptions.

Usage

```
twoSurvSampleSizeNI(
  syear,
  yrsurv1,
  yrsurv2,
  alloc,
  accrualTime,
  followTime,
  alpha,
  power,
  margin
)
```

Arguments

`syear`	Survival time horizon (e.g., median survival time) in years.
`yrsurv1`	Survival probability of the standard group at `syear`.
`yrsurv2`	Survival probability of the test group at `syear`.
`alloc`	Allocation ratio (Test / Standard), e.g., 1 means equal allocation.
`accrualTime`	Duration of patient accrual period.
`followTime`	Follow-up period after last patient is accrued.
`alpha`	One-sided significance level (e.g., 0.025).
`power`	Desired statistical power (e.g., 0.8).
`margin`	Non-inferiority margin for hazard ratio (HR).

Value

A list containing:

Sample_size_of_standard_group Required sample size in the standard group.

Sample_size_of_test_group Required sample size in the test group.

Total_sample_size Total sample size.

Expected_event_numbers_of_standard_group Expected number of events in the standard group.

Expected_event_numbers_of_test_group Expected number of events in the test group.

Total_expected_event_numbers Total number of expected events across both groups.

References

Jung SH, Chow SC. (2012). On sample size calculation for comparing survival curves under general hypothesis testing. *Journal of Biopharmaceutical Statistics*, 22(3), 485–495.

Web calculator (Non-Inferiority): https://nshi.jp/en/js/twosurvyrni/

Examples

```
twoSurvSampleSizeNI(
  syear = 2,
  yrsurv1 = 0.7,
  yrsurv2 = 0.65,
  alloc = 1,
  accrualTime = 1,
  followTime = 1,
  alpha = 0.025,
  power = 0.8,
  margin = 1.3
)
```

---

two_mean_size *Sample Size or Power for Two-Sample Mean Test*

Description

Calculates sample size or power for a two-sample mean test.

Usage

```
two_mean_size(
  muA,
  muB,
  delta = NULL,
  kappa = 1,
  sd = NULL,
  sdA = NULL,
  sdB = NULL,
  alpha,
  beta = NULL,
  nA = NULL,
  nB = NULL,
  test_type = "2-side"
)
```

Arguments

muA	Numeric. True mean of group A.
muB	Numeric. True mean of group B.
delta	Numeric (optional). Margin for "non-inferiority" or "equivalence test". Required for "non-inferiority" or "equivalence" test.
kappa	Numeric. Ratio of sample sizes (nA/nB). Default is 1.
sd	Numeric (optional). Standard deviation. Required for "2-side", "non-inferiority" or "equivalence" test.
sdA	Numeric (optional). Standard deviation of group A. Required for "1-side" test.
sdB	Numeric (optional). Standard deviation of group B. Required for "1-side" test.
alpha	Numeric. Type I error rate.
beta	Numeric (optional). Type II error rate. Required for sample size calculation.
nA	Integer (optional). Sample size for group A. Required for power calculation of "1-side" test.
nB	Integer (optional). Sample size for group B. Required for power calculation of "2-side", "non-inferiority" or "equivalence" test.
test_type	Character. "2-side", "1-side", "non-inferiority", or "equivalence". Default is "2-side".

Value

Numeric. Returns sample size (if beta is given), or power (if nA/nB is given).

Note

Only one of beta (for sample size calculation) or nA/nB (for power calculation) should be specified.

Required arguments by test_type:

- "2-side":
  - For sample size: muA, muB, sd, alpha, beta
  - For power: muA, muB, sd, alpha, nB
- "1-side":
  - For sample size: muA, muB, sdA, sdB, alpha, beta
  - For power: muA, muB, sdA, sdB, alpha, nA
- "non-inferiority"/"equivalence":
  - For sample size: muA, muB, delta, sd, alpha, beta
  - For power: muA, muB, delta, sd, alpha, nB

Examples

```
# Sample size for `"2-side"` test
two_mean_size(muA = 5, muB = 10, kappa = 1, sd = 10,
              alpha = 0.05, beta = 0.2, test_type = "2-side")

# Power of `"2-side"` test
two_mean_size(muA = 5, muB = 10, kappa = 1, sd = 10,
              alpha = 0.05, nB = 63, test_type = "2-side")

# Sample size for `"1-side"` test
two_mean_size(muA = 132.86, muB = 127.44, kappa = 2, sdA = 15.34, sdB = 18.23,
              alpha = 0.05, beta = 0.2, test_type = "1-side")

# Power of `"1-sided"` test
two_mean_size(muA = 132.86, muB = 127.44, kappa = 2, sdA = 15.34, sdB = 18.23,
              alpha = 0.05, nA = 85, test_type = "1-side")

# Sample size for `"non-inferiority"` test
two_mean_size(muA = 5, muB = 5, delta = 5, kappa = 1, sd = 10,
              alpha = 0.05, beta = 0.2, test_type = "non-inferiority")

# Power of `"non-inferiority"` test
two_mean_size(muA = 5, muB = 5, delta = 5, kappa = 1, sd = 10,
              alpha = 0.05, nB = 50, test_type = "non-inferiority")

# Sample size for `"equivalence"` test
two_mean_size(muA = 5, muB = 4, delta = 5, kappa = 1, sd = 10,
              alpha = 0.05, beta = 0.2, test_type = "equivalence")

# Power of `"equivalence"` test
two_mean_size(muA = 5, muB = 4, delta = 5, kappa = 1, sd = 10,
              alpha = 0.05, nB = 108, test_type = "equivalence")
```

---

two_prop_size *Sample Size or Power for Two-Sample Proportion Test*

Description

Calculates sample size or power for a two-sample proportion test.

Usage

```
two_prop_size(
  pA,
  pB,
  delta = NULL,
  kappa = 1,
  alpha,
```

```
  beta = NULL,
  nB = NULL,
  test_type = "2-side"
)
```

Arguments

pA	Numeric. True proportion of group A.
pB	Numeric. True proportion of group B.
delta	Numeric (optional). Margin for "non-inferiority" or "equivalence" test. Required for "non-inferiority" or "equivalence" test.
kappa	Numeric. Ratio of sample sizes (nA/nB). Default is 1.
alpha	Numeric. Type I error rate.
beta	Numeric (optional). Type II error rate. Required for sample size calculation.
nB	Integer (optional). Sample size for group B. Required for power calculation.
test_type	Character. "2-side", "1-side", "non-inferiority", or "equivalence". Default is "2-side".

Value

Numeric. Returns sample size (if beta is given), or power (if nB is given).

Note

Only one of beta (for sample size calculation) or nA/nB (for power calculation) should be specified.

Required arguments by test_type:

- "2-side"/"1-side":
  - For sample size: pA, pB, alpha, beta
  - For power: pA, pB, alpha, nB
- "non-inferiority"/"equivalence":
  - For sample size: pA, pB, delta, alpha, beta
  - For power: pA, pB, delta, alpha, nB

Examples

```
# Sample size for `"2-side"` test
two_prop_size(pA = 0.65, pB = 0.85, kappa = 1,
              alpha = 0.05, beta = 0.2, test_type = "2-side")

# Power of `"2-side"` test
two_prop_size(pA = 0.65, pB = 0.85, kappa = 1,
              alpha = 0.05, nB = 70, test_type = "2-side")

# Sample size for `"1-side"` test
two_prop_size(pA = 0.65, pB = 0.85, kappa = 1,
              alpha = 0.05, beta = 0.2, test_type = "1-side")
```

```
# Power of `"1-sided"` test
two_prop_size(pA = 0.65, pB = 0.85, kappa = 1,
              alpha = 0.05, nB = 55, test_type = "1-side")

# Sample size for `"non-inferiority"` test
two_prop_size(pA = 0.85, pB = 0.65, delta = -0.1, kappa = 1,
              alpha = 0.05, beta = 0.2, test_type = "non-inferiority")

# Power of `"non-inferiority"` test
two_prop_size(pA = 0.85, pB = 0.65, delta = -0.1, kappa = 1,
              alpha = 0.05, nB = 25, test_type = "non-inferiority")

# Sample size for `"equivalence"` test
two_prop_size(pA = 0.65, pB = 0.85, delta = 0.05, kappa = 1,
              alpha = 0.05, beta = 0.2, test_type = "equivalence")

# Power of `"equivalence"` test
two_prop_size(pA = 0.65, pB = 0.85, delta = 0.05, kappa = 1,
              alpha = 0.05, nB = 136, test_type = "equivalence")
```

Index