Package: winratiosim (via r-universe)

July 6, 2026

Type Package

Title Simulate Power for Hierarchical Win Ratio Endpoints

Version 1.0.0

Author Se Yoon Lee [aut, cre]

Maintainer Se Yoon Lee <`seyoonlee.stat.math@gmail.com`>

Description Provides simulation tools for power analysis in two-arm clinical trials with hierarchical win ratio endpoints. The package simulates time-to-event, recurrent event, and continuous outcomes, applies prioritized pairwise win/loss scoring, and summarizes win ratio and Finkelstein-Schoenfeld test operating characteristics.

License GPL-2

Encoding UTF-8

RoxygenNote 7.3.2

Imports parallel

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL `https://github.com/yain22/winratiosim`

BugReports `https://github.com/yain22/winratiosim/issues`

Depends R (>= 4.0.0)

NeedsCompilation no

Repository https://cran.r-universe.dev

Date/Publication 2026-07-06 14:10:02 UTC

RemoteUrl https://github.com/cran/winratiosim

RemoteRef HEAD

RemoteSha 5da374bbc571dfee724c8425bf56b9eeef6271ff

Contents

`binom.conf.exact` *Exact Binomial Confidence Interval*

Description

Computes an exact two-sided Clopper-Pearson confidence interval for a binomial proportion by inverting the binomial test.

Usage

```
binom.conf.exact(x, n, alpha = 0.05)
```

Arguments

`x`	Integer. Number of observed successes.
`n`	Integer. Total number of trials.
`alpha`	Numeric. Significance level for the confidence interval. The default is `0.05`, corresponding to a 95 percent interval.

Value

A named numeric vector with three elements:

PointEst Observed proportion, `x / n`.

Lower Lower confidence limit.

Upper Upper confidence limit.

Examples

```
binom.conf.exact(x = 8, n = 10)
binom.conf.exact(x = 50, n = 100, alpha = 0.01)
```

`Scoring_Conti` *Score Continuous Pairwise Comparisons*

Description

Assigns win, loss, tie, or unresolved scores to subject pairs based on a continuous endpoint. This function is typically used after higher-priority layers have left a pair unresolved.

Usage

```
Scoring_Conti(dataset, higher_better, var1, var2)
```

Arguments

`dataset`	A data frame containing pairwise subject comparisons. The data frame must contain columns named `score`, `WR_cat`, `usubjid1`, and `usubjid2`.
`higher_better`	Character. Use `"Yes"` when higher values are better and `"No"` when lower values are better.
`var1`	Character. Name of the continuous endpoint column for subject 1.
`var2`	Character. Name of the continuous endpoint column for subject 2.

Value

A data frame matching `dataset`, with updated `score` and `WR_cat` columns. Scores are 1 when subject 1 wins, -1 when subject 2 wins, 0 for exact or near-exact ties, and `NA` when either value is missing.

Examples

```
pairs <- data.frame(
  usubjid1 = c(1, 1, 2),
  usubjid2 = c(3, 4, 4),
  kccq1 = c(15, 10, NA),
  kccq2 = c(10, 10, 12),
  score = NA_real_,
  WR_cat = ""
)

Scoring_Conti(pairs, higher_better = "Yes", var1 = "kccq1", var2 = "kccq2")
```

`Scoring_TTE` *Score Time-to-Event Pairwise Comparisons*

Description

Assigns win, loss, or unresolved scores to subject pairs based on a time-to-event endpoint. This function is typically used for the first, highest-priority layer in a hierarchical win ratio analysis.

Usage

```
Scoring_TTE(dataset, var1, var2, censor1, censor2)
```

Arguments

`dataset`	A data frame containing pairwise subject comparisons. The data frame must contain columns named `score`, `WR_cat`, `usubjid1`, and `usubjid2`.
`var1`	Character. Name of the time-to-event column for subject 1.
`var2`	Character. Name of the time-to-event column for subject 2.
`censor1`	Character. Name of the event indicator column for subject 1, coded as 1 for event and 0 for censored.
`censor2`	Character. Name of the event indicator column for subject 2, coded as 1 for event and 0 for censored.

Value

A data frame matching `dataset`, with updated `score` and `WR_cat` columns. Scores are 1 when subject 1 wins, -1 when subject 2 wins, and `NA` when the comparison remains tied or unresolved because of censoring.

Examples

```
pairs <- data.frame(
  usubjid1 = c(1, 1),
  usubjid2 = c(2, 3),
  deathdays1 = c(360, 120),
  deathdays2 = c(100, 200),
  death1 = c(0, 1),
  death2 = c(1, 1),
  score = NA_real_,
  WR_cat = ""
)

Scoring_TTE(pairs, "deathdays1", "deathdays2", "death1", "death2")
```

`SimData_per_group` *Simulate Individual-Level Trial Data for One Arm*

Description

Generates individual-level simulated data for a treatment or control arm in a hierarchical win ratio trial. The simulation includes frailty-adjusted time to death, recurrent event counts, censoring times, and a continuous quality-of-life change score.

Usage

```
SimData_per_group(
  treatment,
  ngroup,
  alpha.JFM,
  theta.JFM,
  lambda,
  ann.icr,
  censorrate,
  xbase,
  xfinal,
  sd.delta.x
)
```

Arguments

`treatment`	Integer. Treatment group indicator, usually 1 for the active treatment arm and 0 for the control arm.
`ngroup`	Integer. Number of subjects to simulate in this arm.
`alpha.JFM`	Numeric. Alpha parameter for the joint frailty model.
`theta.JFM`	Numeric. Frailty variance parameter for the joint frailty model. Must be positive.
`lambda`	Numeric. Annual mortality probability. Must be in `[0, 1)`.
`ann.icr`	Numeric. Annual incidence rate of recurrent events.
`censorrate`	Numeric. Annual censoring probability. Must be in `[0, 1)`.
`xbase`	Numeric. Baseline value of the continuous outcome.
`xfinal`	Numeric. Expected final value of the continuous outcome among subjects followed through 360 days.
`sd.delta.x`	Numeric. Standard deviation of the change in the continuous outcome.

Value

A named list. If `treatment = 1`, the list contains `surv_1`; otherwise, it contains `surv_0`. The data frame has one row per subject and includes subject ID, treatment indicator, death time, censoring time, death indicator, recurrent event count, and continuous outcome value.

Examples

```
set.seed(1)
sim <- SimData_per_group(
  treatment = 1, ngroup = 5,
  alpha.JFM = 0, theta.JFM = 1,
  lambda = 0.13, ann.icr = 0.32,
  censorrate = 0.2, xbase = 45, xfinal = 52.5,
  sd.delta.x = 20
)
str(sim$surv_1)
```

`winratiosim` *Simulate Hierarchical Win Ratio Trials*

Description

Simulates replicated two-arm clinical trials and analyzes each trial using a three-layer hierarchical win ratio framework: time to death, annualized recurrent event count, and a continuous quality-of-life score.

Usage

```
winratiosim(
  nsim,
  N,
  Randomization.ratio,
  alpha.JFM,
  theta.JFM,
  lambda_trt,
  lambda_ctl,
  ann.icr_trt,
  ann.icr_ctl,
  xbase_trt,
  xfinal_trt,
  xbase_ctl,
  xfinal_ctl,
  sd.delta.x_trt,
  sd.delta.x_ctl,
  censorrate_trt,
  censorrate_ctl,
  nc = 1,
  seed = NULL
)
```

Arguments

`nsim`	Integer. Number of simulated trials.

`N`	Integer. Total number of subjects in each simulated trial.
`Randomization.ratio`	Numeric vector of length 2 giving the treatment and control allocation ratio, for example `c(1, 1)`.
`alpha.JFM`	Numeric. Alpha parameter for the joint frailty model.
`theta.JFM`	Numeric. Frailty variance parameter for the joint frailty model. Must be positive.
`lambda_trt`, `lambda_ctl`	Numeric. Annual mortality probabilities for the treatment and control arms.
`ann.icr_trt`, `ann.icr_ctl`	Numeric. Annual recurrent event incidence rates for the treatment and control arms.
`xbase_trt`, `xfinal_trt`	Numeric. Baseline and expected final continuous outcome values in the treatment arm.
`xbase_ctl`, `xfinal_ctl`	Numeric. Baseline and expected final continuous outcome values in the control arm.
`sd.delta.x_trt`, `sd.delta.x_ctl`	Numeric. Standard deviations for the continuous outcome change in the treatment and control arms.
`censorrate_trt`, `censorrate_ctl`	Numeric. Annual censoring probabilities for the treatment and control arms.
`nc`	Integer. Number of worker processes to use. The default is 1.
`seed`	Optional integer seed. If supplied, results are reproducible across different values of `nc`.

Value

A named list with the following elements:

df_FS.analysis.summary Finkelstein-Schoenfeld analysis summary for each simulation.

df_WR.analysis.summary Win ratio analysis summary for each simulation.

df_sample.size.summary Sample sizes used in each simulated trial.

df_Total_probability Win, tie, loss, and total probabilities for each simulation.

df_Total_count Win, tie, loss, and total counts for each simulation.

References

Lee, S. Y. (2025). A note on the sample size formula for a win ratio endpoint. *Statistics in Medicine*, 44, e70165. doi:10.1002/sim.70165

Examples

```
result <- winratiosim(
  nsim = 1,
  N = 20,
  Randomization.ratio = c(1, 1),
  alpha.JFM = 0,
  theta.JFM = 1,
  lambda_trt = 0.13,
  lambda_ctl = 0.15,
  ann.icr_trt = 0.32,
  ann.icr_ctl = 0.55,
  xbase_trt = 45,
  xfinal_trt = 52.5,
  xbase_ctl = 45,
  xfinal_ctl = 45,
  sd.delta.x_trt = 20,
  sd.delta.x_ctl = 20,
  censorrate_trt = 0.2,
  censorrate_ctl = 0.2,
  nc = 1,
  seed = 2025
)
result$df_WR.analysis.summary
```

`WR_analysis` *Perform Hierarchical Win Ratio Analysis*

Description

Analyzes treatment-control pairwise comparisons across three prioritized outcome layers. The function computes layer-specific win, tie, and loss counts; sample sizes; Finkelstein-Schoenfeld statistics; and win ratio statistics based on permutation and large-sample variance formulas.

Usage

```
WR_analysis(dataset1, dataset2, dataset3)
```

Arguments

`dataset1`	Data frame containing pairwise scores for the first, highest-priority layer.
`dataset2`	Data frame containing pairwise scores through the second layer.
`dataset3`	Data frame containing pairwise scores through the third layer.

Value

A named list with four elements:

win.losses.count.summary Counts and proportions of treatment wins, ties, and treatment losses by layer and overall.

sample.size.summary Treatment, control, total, and pairwise comparison counts.

FS.analysis.summary Finkelstein-Schoenfeld statistic, variance, z-score, and one-sided p-value.

WR.analysis.summary Win ratio, log win ratio, variance estimates, confidence limits, and one-sided p-value.

References

Finkelstein, D. M., and Schoenfeld, D. A. (1999). Combining mortality and longitudinal measures in clinical trials. *Statistics in Medicine*, 18(11), 1341-1354.

Pocock, S. J., Ariti, C. A., Collier, T. J., and Wang, D. (2012). The win ratio: a new approach to the analysis of composite endpoints in clinical trials based on clinical priorities. *European Heart Journal*, 33(2), 176-182.

Yu, R. X., and Ganju, J. (2022). Sample size formula for a win ratio endpoint. *Statistics in Medicine*, 41(6), 950-963.

Examples

```
subjects <- data.frame(
  usubjid = c(1, 2, 1001, 1002),
  treatment = c(1, 1, 0, 0)
)
dataset1 <- merge(subjects, subjects, by = NULL)
names(dataset1) <- c("usubjid1", "treatment1", "usubjid2", "treatment2")
dataset1$score <- NA_real_
wr_rows <- dataset1$treatment1 == 1 & dataset1$treatment2 == 0
dataset1$score[wr_rows] <- c(1, 1, -1, -1)

dataset2 <- dataset1
dataset3 <- dataset1
WR_analysis(dataset1, dataset2, dataset3)$sample.size.summary
```

Index