Package 'WR'

November 13, 2025

Type Package

Title Win Ratio Analysis of Composite Time-to-Event Outcomes

Version 1.0

Author Lu Mao, Tuo Wang, Adrien Orué

Maintainer Lu Mao <lmao@biostat.wisc.edu>

URL https://sites.google.com/view/lmaowisc/

Description Implements various win ratio methodologies for composite endpoints of death and non-fatal events, including the (stratified) proportional win-fractions (PW) regression models (Mao and Wang, 2020 <doi:10.1111/biom.13382>), (stratified) two-sample tests with possibly recurrent nonfatal event, sample size calculation for standard win ratio test (Mao et al., 2021 <doi:10.1111/biom.13501>) and sample size computation for last-event-assisted win ratio using Monte-Carlo simulation.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

RoxygenNote 7.3.3

Imports survival, cubature, gumbel, Rcpp, ggplot2

LinkingTo Rcpp

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-11-26 21:20:08 UTC

Contents

`base` *Compute the baseline parameters needed for sample size calculation for standard win ratio test*

Description

Compute the baseline parameters ζ_0^2 and $\boldsymbol{\delta}_0$ needed for sample size calculation for standard win ratio test (see `WRSS`). The calculation is based on a Gumbel–Hougaard copula model for survival time $D^{(a)}$ and nonfatal event time $T^{(a)}$ for group a (1: treatment; 0: control):

$$P(D^{(a)} > s, T^{(a)} > t) = \exp\left(-\left[\{\exp(a\xi_1)\lambda_D s\}^{\kappa} + \{\exp(a\xi_2)\lambda_H t\}^{\kappa}\right]^{1/\kappa}\right),$$

where ξ_1 and ξ_2 are the component-wise log-hazard ratios to be used as effect size in `WRSS`. We also assume that patients are recruited uniformly over the period $[0, \tau_b]$ and followed until time τ ($\tau \geq \tau_b$), with an exponential loss-to-follow-up hazard λ_L.

Usage

```
base(lambda_D, lambda_H, kappa, tau_b, tau, lambda_L, N = 1000, seed = 12345)
```

Arguments

`lambda_D`	Baseline hazard λ_D for death.
`lambda_H`	Baseline hazard λ_H for nonfatal event.
`kappa`	Gumbel–Hougaard copula correlation parameter κ.
`tau_b`	Length of the initial (uniform) accrual period τ_b.
`tau`	Total length of follow-up τ.
`lambda_L`	Exponential hazard rate λ_L for random loss to follow-up.
`N`	Simulated sample size for monte-carlo integration.
`seed`	Seed for monte-carlo simulation.

Value

A list containing real number `zeta2` for ζ_0^2 and bivariate vector `delta` for $\boldsymbol{\delta}_0$.

References

Mao, L., Kim, K. and Miao, X. (2021). Sample size formula for general win ratio analysis. Biometrics, https://doi.org/10.1111/biom.13501.

See Also

gumbel.est, WRSS

Examples

```
# see the example for WRSS
```

---

gbc *A subset of the German Breast Cancer study data*

Description

These are a subset of the German Breast Cancer study data.

Usage

```
gbc
```

Format

A data frame with 985 rows and 12 variables:

id subject IDs

time event times (months)

status event status; 0:censoring, 1:death, 2:cancer recurrence

hormone treatment indicator: 1=Hormone therapy; 2=standard therapy

age age at diagnosis (years)

menopause menopausal Status; 1=No; 2=Yes

size tumor size

grade tumor grade, 1-3

nodes number of nodes involved

prog_recp number of progesterone receptors

estrg_recp number of estrogen receptors

References

Sauerbrei, W., Royston, P., Bojar, H., Schmoor, C. and Schumacher, M. (1999). Modelling the effects of standard prognostic factors in node-positive breast cancer. German Breast Cancer Study Group (GBSG). British Journal of Cancer, 79, 1752–1760.

Hosmer, D.W. and Lemeshow, S. and May, S. (2008) Applied Survival Analysis: Regression Modeling of Time to Event Data: Second Edition, John Wiley and Sons Inc., New York, NY

`gumbel.est` *Estimate baseline parameters in the Gumbel–Hougaard model for sample size calculation using pilot data*

Description

Estimate baseline parameters in the Gumbel–Hougaard model described in base for sample size calculation using pilot study data.

Usage

```
gumbel.est(id, time, status)
```

Arguments

`id`	A vector of unique patient identifiers.
`time`	A numeric vector of event times.
`status`	A vector of event type variable; 2 = nonfatal event, 1 = death, and 0 = censoring.

Value

A list containing `lambda_D` for λ_D, `lambda_H` for λ_H, and `kappa` for κ in the Gumbel–Hougaard model.

References

Mao, L., Kim, K. and Miao, X. (2021). Sample size formula for general win ratio analysis. Biometrics, https://doi.org/10.1111/biom.13501.

See Also

base, WRSS

Examples

```
# see the example for WRSS
```

`hfaction_cpx9` *A subset of the HF-ACTION study data on high-risk non-ischemic heart failure patients*

Description

These are data on a subgroup of 426 high-risk non-ischemic patients in the HF-ACTION study.

Usage

```
hfaction_cpx9
```

Format

A data frame with 1,448 rows and 5 variables:

patid patient ID

time event times (months)

status event status; 0:censoring, 1:death, 2:hospitalization

trt_ab treatment indicator: 1: exercise training, 0: usual care

age60 1: 60 years or older, 0: less than 60 years old

References

O'Connor, C. M., Whellan, D. J., Lee, K. L., Keteyian, S. J., Cooper, L. S., Ellis, S. J., Leifer, E. S., Kraus, W. E., Kitzman, D. W., Blumenthal, J. A. et al. (2009). Efficacy and safety of exercise training in patients with chronic heart failure: HF-ACTION randomized controlled trial. Journal of the American Medical Association, 301, 1439–1450.

---

non_ischemic *A subset of the HF-ACTION study data on non-ischemic heart failure patients with full covariate measurement.*

Description

These are a subset of the data on 451 non-ischemic patients in the HF-ACTION study will complete baseline covariates.

Usage

```
non_ischemic
```

Format

A data frame with 751 rows and 16 variables:

ID subject IDs

time event times (days)

status event status; 0:censoring, 1:death, 2:hospitalization

trt_ab treatment indicator: 1=exercise training; 0=usual care

age patient age in years

sex 1=female; 2=male

Black.vs.White 1=black; 0=otherwise

Other.vs.White 1=race other than black or white; 0=otherwise

bmi body mass index

bipllvef (biplane) left-ventricular ejection fraction

hyperten indicator for history of hypertension

COPD indicator for history of COPD

diabetes indicator for history of diabetes

acei indicator for current use of ACE inhibitors

betab indicator for current use of beta blockers

smokecurr indicator for current smoker

References

O'Connor, C. M., Whellan, D. J., Lee, K. L., Keteyian, S. J., Cooper, L. S., Ellis, S. J., Leifer, E. S., Kraus, W. E., Kitzman, D. W., Blumenthal, J. A. et al. (2009). Efficacy and safety of exercise training in patients with chronic heart failure: HF-ACTION randomized controlled trial. Journal of the American Medical Association, 301, 1439–1450.

`plot.pwreg.score` *Plot the standardized score processes*

Description

Plot the standardized score processes.

Usage

```
## S3 method for class 'pwreg.score'
plot(
  x,
  k,
  xlab = "Time",
  ylab = "Standardized score",
  lty = 1,
  frame.plot = TRUE,
  add = FALSE,
  ylim = c(-3, 3),
  xlim = NULL,
  lwd = 1,
  ...
)
```

Arguments

`x`	an object of class `pwreg.score`.
`k`	A positive integer indicating the order of covariate to be plotted. For example, `k=3` requests the standardized score process for the third covariate in the covariate matrix `Z`.
`xlab`	a title for the x axis.
`ylab`	a title for the y axis.
`lty`	the line type. Default is 1.
`frame.plot`	a logical variable indicating if a frame should be drawn in the 1D case.
`add`	a logical variable indicating whether add to current plot?
`ylim`	a vector indicating the range of y-axis. Default is (-3,3).
`xlim`	a vector indicating the range of x-axis. Default is NULL.
`lwd`	the line width, a positive number. Default is 1.
`...`	further arguments passed to or from other methods

Value

A plot of the standardized score process for object `pwreg.score`.

See Also

score.proc

Examples

```
# see the example for score.proc
```

---

print.pwreg *Print the results of the proportional win-fractions regression model*

Description

Print the results of the proportional win-fractions regression model.

Usage

```
## S3 method for class 'pwreg'
print(x, digits = 3, ...)
```

Arguments

x	an object of class pwreg.
digits	number of significant digits to display (default is 3).
...	further arguments passed to or from other methods

Value

Print the results of pwreg object

See Also

pwreg

Examples

```
# see the example for pwreg
```

print.pwreg.score *Print information on the content of the pwreg.score object*

Description

Print information on the content of the pwreg.score object

Usage

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

Arguments

x	A object of class pwreg.score.
...	further arguments passed to or from other methods.

Value

Print the results of `pwreg.score` object.

See Also

`score.proc`

Examples

```
# see the example for score.proc
```

print.WRrec *Print the results of the two-sample recurrent-event win ratio analysis*

Description

Print the results of the two-sample recurrent-event win ratio analysis.

Usage

```
## S3 method for class 'WRrec'
print(x, digits = 3, ...)
```

Arguments

x	an object of class `WRrec`.
digits	number of significant digits to display (default is 3).
...	further arguments passed to or from other methods.

Value

Print the results of `WRrec` object.

See Also

WRrec

Examples

```
# see the example for WRrec
```

`process.dat` *Reorganize recurrent-event data by treatment arm*

Description

Sorts longitudinal patient records by ID and time, then splits them into two arm-specific lists (trt = 1 and trt = 0). Within each arm, each patient is represented by a list containing a scalar ID, a vector of ordered observation times, and the patient's final (last observed) status code.

Usage

```
process.dat(ID, time, status, trt)
```

Arguments

`ID`	Vector of patient identifiers (integer, numeric, or character). All elements corresponding to the same patient must be identical.
`time`	Numeric vector of observation/event times. Must be non-decreasing within patient after sorting; ties are allowed but will be preserved as given after sorting.
`status`	Integer or numeric vector of status codes per observation. The function retains only the last observed status per patient in the output.
`trt`	Integer or logical treatment indicator with values 1 (treated) and 0 (control). Values are coerced via 'trt == 1L' to define the treated arm.

Value

A named list with two elements: - 'dat1': a list of patients in the treated arm ('trt == 1L'), - 'dat0': a list of patients in the control arm ('trt == 0L').

Each element ('dat1'/'dat0') is a named list where names correspond to patient IDs. Each patient entry is itself a list with components: - 'ID': scalar patient ID, - 'time': numeric vector of ordered observation times for that patient, - 'status': scalar final status (last observed status for that patient).

pwreg *Fit a standard proportional win-fractions (PW) regression model*

Description

Fit a standard proportional win-fractions (PW) regression model.

Usage

```
pwreg(
  ID,
  time,
  status,
  Z,
  rho = 0,
  strata = NULL,
  fixedL = TRUE,
  eps = 1e-04,
  maxiter = 50
)
```

Arguments

ID	a vector of unique subject-level identifiers.
time	a vector of event times.
status	a vector of event type labels. 0: censoring, 1:death and 2: non-fatal event.
Z	a matrix or a vector of covariates.
rho	a non-negative number as the power of the survival function used in the weight. Default (rho=0) is recommended. If there is a 'strata' argument, then 'rho' is ignored.
strata	a vector of stratifying variable if a stratified model is desired.
fixedL	logical variable indicating which variance estimator to be used. If 'TRUE', the type I variance estimator (for a small number strata) is used; otherwise the type II variance estimator (for a large number strata) is used.
eps	precision for the convergence of Newton-Raphson algorithm.
maxiter	maximum number of iterations allow for the Newton-Raphson algorithm.

Value

An object of class pwreg with the following components. beta:a vector of estimated regression coefficients. Var:estimated covariance matrix for beta. conv: boolean variable indicating whether the algorithm converged within the maximum number of iterations.

References

Mao, L. and Wang, T. (2020). A class of proportional win-fractions regression models for composite outcomes. Biometrics, 10.1111/biom.13382

Wang, T. and Mao, L. (2021+). Stratified Proportional Win-fractions Regression Analysis.

See Also

score.proc, print.pwreg

Examples

```
library(WR)
head(non_ischemic)
id_unique <-unique(non_ischemic$ID)

# Randomly sample 200 subjects from non_ischemic data
set.seed(2019)
id_sample <- sample(id_unique, 200)
non_ischemic_reduce <- non_ischemic[non_ischemic$ID %in% id_sample, ]

# Use the reduced non_ischemic data for analysis
nr <- nrow(non_ischemic_reduce)
p <- ncol(non_ischemic_reduce)-3
ID <- non_ischemic_reduce[,"ID"]
time <- non_ischemic_reduce[,"time"]
status <- non_ischemic_reduce[,"status"]
Z <- as.matrix(non_ischemic_reduce[,4:(3+p)],nr,p)
## unstratified analysis
pwreg.obj <- pwreg(time=time,status=status,Z=Z,ID=ID)
print(pwreg.obj)
## Not run:
## stratified PW by sex
sex<-Z[,3]
## take out sex from the covariate matrix
Z1<-Z[,-3]
pwreg.obj1 <- pwreg(time=time,status=status,Z=Z1,ID=ID,strata=sex)
print(pwreg.obj1)

## End(Not run)
```

`score.proc` *Computes the standardized score processes*

Description

Computes the standarized score processes for the covariates.

Usage

```
score.proc(obj, t = NULL)
```

Arguments

`obj`	an object of class pwreg.
`t`	a vector containing times. If not specified, the function will use all unique event times from the data.

Value

An object of class `pwreg.score` consisting of `t:` a vector of times; and `score:` a matrix whose rows are the standardized score processes as a function of `t`.

References

Mao, L. and Wang, T. (2020). A class of proportional win-fractions regression models for composite outcomes. Biometrics, 10.1111/biom.13382

See Also

`pwreg`, `print.pwreg`

Examples

```
library(WR)
head(non_ischemic)

# Randomly sample 200 subjects from non_ischemic data
id_unique <-unique(non_ischemic$ID)
set.seed(2019)
id_sample <- sample(id_unique, 200)
non_ischemic_reduce <- non_ischemic[non_ischemic$ID %in% id_sample, ]

# Use the reduced non_ischemic data for analysis
nr <- nrow(non_ischemic_reduce)
p <- ncol(non_ischemic_reduce)-3
ID <- non_ischemic_reduce[,"ID"]
time <- non_ischemic_reduce[,"time"]
status <- non_ischemic_reduce[,"status"]
Z <- as.matrix(non_ischemic_reduce[,4:(3+p)],nr,p)
pwreg.obj <- pwreg(time=time,status=status,Z=Z,ID=ID)
score.obj <- score.proc(pwreg.obj)
#plot the standardized score process for the first covariate
plot(score.obj, k = 1)
```

---

sz_lwr *Sample size computation for recurrent event using the last-event-assisted win ratio (LWR)*

Description

This function performs Monte Carlo simulations under both the alternative and null hypotheses to identify the smallest sample size achieving the desired power while controlling the type I error of the win ratio procedure. The data generation is based on a joint frailty model with Weibull baseline hazards for both recurrent and terminal events. This allows, in particular, for taking into account the heterogeneity and association between recurrent events and the terminal event.

Usage

```
sz_lwr(
  power,
  type1Error,
  HR_recurrent,
  HR_terminal,
  weibShapeRec,
  weibScaleRec,
  weibShapeTerm,
  weibScaleTerm,
  niType = "max",
  ni = Inf,
  theta,
  alpha,
  expoCensoringRate = NULL,
  FUP = 3,
  fupType = "fixed",
  accrualDuration = 0,
  sample_size_min = 400,
  sample_size_max = 2000,
  sample_size_step = 200,
  n_sim = 500,
  baseSeed = 42,
  output_plot_file = sprintf("power_curve_%s.png", gsub("[ :\\-]", "_",
    round(Sys.time(), 0))),
  ...
)
```

Arguments

power	Numeric value between 0 and 1, representing the desired power for the alternative scenario.
type1Error	Numeric value between 0 and 1, representing the nominal type I error level for the LWR test.
HR_recurrent	Positive value, corresponding to the hazard ratio for recurrent events under the alternative.
HR_terminal	Positive value, corresponding to the hazard ratio for terminal events under the alternative.
weibShapeRec	Positive value, corresponding to the Weibull shape for the recurrent baseline hazard.
weibScaleRec	Positive value, corresponding to the Weibull scale for the recurrent baseline hazard.
weibShapeTerm	Positive value, corresponding to the Weibull shape for the terminal event baseline hazard.
weibScaleTerm	Positive value, corresponding to the Weibull scale for the terminal event baseline hazard.
niType	Character string, describing the nature of the recurrent events. Options are: • "max": Each subject can have up to ni recurrent events. • "poisson": Each subject's maximum recurrent-event count is drawn from Poisson(ni).

- "unif": Each subject's maximum recurrent-event count is drawn uniformly between the values in `ni = c(lower, upper)`.

Defaults to `niType = "max"`.

`ni` Character string, that is associated with `niType`.

- If `niType` is "max", `ni` is the maximum number of recurrent events per subject.
- If `niType` is "poisson", `ni` is the mean of the Poisson distribution used to draw per-subject caps.
- If `niType` is "unif", `ni` is a 2D-vector corresponding to the minimum and maximum cap for each subject.

Defaults to `ni = Inf` (so that, with `niType = "max"`, we generate as many recurrent events per subject as possible).

`theta` Positive value, corresponding to the gamma frailty variance.

`alpha` Numeric value, controlling association between recurrent and terminal processes.

`expoCensoringRate`
Optional positive value, that corresponds to the Exponential parameter (i.e., rate). If provided, an additional independent exponential censoring is added to the administrative censoring. Defaults to `NULL`.

`FUP` Positive numeric value, corresponding to the follow-up duration (i.e. administrative censoring). Default is 3.

`fupType` Character string, indicating the follow-up scheme (case-insensitive). Options are:

- `"fixed"`: each subject is followed exactly for `FUP` time units after enrollment.
- `"uptoend"`: global study cutoff at time `FUP`; individual follow-up for at most `FUP` depending on their accrual time.

Default is `"fixed"`.

`accrualDuration`
Non-negative numeric value describing a uniform accrual window from time 0 to `accrualDuration`. Only meaningful when `fupType = "uptoend"`. Defaults to 0 (no "accrual").

`sample_size_min`
Minimum sample size checked. Defaults to 400.

`sample_size_max`
Maximum sample size checked. Defaults to 2000.

`sample_size_step`
Step size for the sample size grid. Defaults to 200.

`n_sim` Number of Monte Carlo replications per scenario. Defaults to 500.

`baseSeed` Integer base seed for random number generation. Defaults to 42.

`output_plot_file`
File path where the power curve is saved.

`...` Additional arguments passed to the data generation function `data_jfm()`.

Details

Under the null, both HR (rec. and terminal) are set to 1.0.

The Weibull parametrization is: $h_0(t) = \frac{shape}{scale}\left(\frac{t}{scale}\right)^{shape-1}$. If one want to generate data with an exponential baseline risk, put shape = 1 and scale = 1 / lambda, where lambda is the rate (parameter of the exponential distribution).

We recall that a gamma-joint frailty model includes a common frailty term to the individuals (ω_i) for the two rates which will take into account the heterogeneity in the data, associated with unobserved covariates. The frailty term acts differently for the two rates (ω_i for the recurrent rate and ω_i^α for the death rate). The covariates could be different for the recurrent rate and death rate. For the j^{th} recurrence $(j = 1, ..., n_i)$ and the i^{th} subject $(i = 1, ..., N)$, the joint gamma frailty model for recurrent event hazard function $r_{ij}(.)$ and death rate $\lambda_i(.)$ is:

$$\begin{cases} r_{ij}(t|\omega_i) = \omega_i r_0(t)\exp(\boldsymbol{\beta_1'} \boldsymbol{Z_i(t)}) & \text{(Recurrent)} \\ \lambda_i(t|\omega_i) = \omega_i^\alpha \lambda_0(t)\exp(\boldsymbol{\beta_2'} \boldsymbol{Z_i(t)}) & \text{(Death)} \end{cases}$$

where $r_0(t)$ (resp. $\lambda_0(t)$) is the recurrent (resp. terminal) event baseline hazard function, $\boldsymbol{\beta_1}$ (resp. $\boldsymbol{\beta_2}$) the regression coefficient vector, $\boldsymbol{Z_i(t)}$ the covariate vector. The random effects of frailties $\omega_i \sim \boldsymbol{\Gamma}(\frac{1}{\theta}, \frac{1}{\theta})$, are iid, with mean 1 and variance θ. The parameter α quantifies the strength (and direction) of association between the recurrent event process and the terminal event.

Value

A list with the following components:

`summary_table` Data frame summarizing the estimated power and type I error across sample sizes.

`computed_sample_size` Sample size achieving the desired power while controlling type I error (if any found).

`achieved_power` Estimated power at the recommended sample size (if any found).

`achieved_type_I_error` Estimated type I error at the recommended sample size (if any found).

`expected_ties` Named vector of expected tie counts at each sample size.

`ties_at_n_star` Expected tie count at the recommended sample size (if any found).

`power_curve_plot` ggplot2 object representing the power curve across sample sizes.

`output_plot_file` File path where the power curve was saved.

References

Mao, L., Kim, K. and Li, Y. (2022). On recurrent-event win ratio. Statistical Methods in Medical Research, under review.

Pocock, S., Ariti, C., Collier, T., 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, 176–182.

V. Rondeau, S. Mathoulin-Pellissier, H. Jacqmin-Gadda, V. Brouste, P. Soubeyran (2007). Joint frailty models for recurring events and death using maximum penalized likelihood estimation:application on cancer events. *Biostatistics* **8**,4, 708-721.

Examples

```
## Not run:
res <- sz_lwr(
    power = 0.80,
    type1Error = 0.05,
    weibShapeRec = 1,
    weibScaleRec = (3.6 / 12) / log(2),
    weibShapeTerm = 1,
    weibScaleTerm = (28 / 12) / log(2),
    HR_recurrent = 0.80,
    HR_terminal = 0.90,
    niType = "poisson",
    ni = 3,
    theta = 1.0,
    alpha = 1.0,
    fupType = "uptoend",
    FUP = 4,
    accrualDuration = 3,
    sample_size_min = 400,
    sample_size_max = 2000,
    sample_size_step = 200,
    n_sim = 500,
    baseSeed = 42,
    output_plot_file = sprintf(
        "power_curve_%s.png",
        gsub("[ :\\-]", "_", round(Sys.time(), 0))
    )
)

## End(Not run)
```

WRrec *Win Ratio for Recurrent Events*

Description

Perform stratified two-sample test of possibly recurrent nonfatal event and death. A total of three win ratio variants are implemented: the last-event assisted win ratio (LWR, the default), the naive win ratio (NWR), and the first-event assisted win ratio (FWR) (Mao et al., 2022). The LWR is the primary method recommended for use (Mao et al., 2022). The LWR and FWR reduce to the standard win ratio of Pocock et al. (2012).

The method also supports stratification: comparisons are performed within each stratum and then combined to produce an overall estimate.

Usage

```
WRrec(ID, time, status, trt, strata = NULL, naive = FALSE)
```

Arguments

ID	Vector (integer, character, or factor). Corresponds to the subjects identification. Repeated values indicate multiple recurrent event records for the same subject.

time	Numeric vector. Contains event or censoring times aligned with ID. Must be non-negative; assumed already on the analysis time scale.
status	Numeric vector. Contains event indicators aligned with ID. Valid values are: 0 (censoring), 1 (terminal event), and 2 (recurrent event). Each subject can have at most one final status of 0 or 1.
trt	Numeric vector. Contains the treatment group indicator aligned with ID. Valid values are binary (0/1) or a factor with exactly two levels.
strata	(Optional) Numeric vector. If supplied, comparisons are restricted to individuals with the same stratum value, aligned with ID. Default is NULL (unstratified).
naive	Logical. If TRUE, computes "FWR" and "NWR" in addition to "LWR". Default is FALSE (only "LWR" is computed).

Value

A list with the following components:

log.WR Estimated log LWR.

se Standard error of log.WR.

pval Two-sided p-value testing H0: LWR = 1.

theta Wins and losses proportions, respectively, for LWR.

desc Data frame summarizing, for each treatment arm, the number of subjects, number of recurrent events, number of deaths, and median follow-up time.

nb.pairs Total number of pairs.

If naive = TRUE, the list also contains:

log.WR.naive Estimated log NWR.

se.naive Standard error of log.WR.naive.

theta.naive Wins and losses proportions, respectively, for NWR.

pval.naive Two-sided p-value testing H0: NWR = 1.

log.WR.FI Estimated log FWR.

se.FI Standard error of log.WR.FI.

theta.FI Wins and losses proportions, respectively, for FWR.

pval.FI Two-sided p-value testing H0: FWR = 1.

Interpretation

A win ratio greater than 1 favors subjects in group trt == 1 (more wins than losses across pairwise recurrent event trajectory comparisons). The log scale is used for inference; confidence intervals can be constructed as exp(log.WR +/- z_{alpha/2} * se).

References

Mao, L., Kim, K. and Li, Y. (2022). On recurrent-event win ratio. Statistical Methods in Medical Research, under review.

Pocock, S., Ariti, C., Collier, T., 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, 176–182.

See Also

print.WRrec.

Examples

```
## load the HF-ACTION trial data
library(WR)
head(hfaction_cpx9)
dat <- hfaction_cpx9
## Comparing exercise training to usual care by LWR, FWR, and NWR
obj <- WRrec(ID=dat$patid, time=dat$time, status=dat$status,
          trt=dat$trt_ab, strata=dat$age60, naive=TRUE)
## print the results
obj
```

WRSS *Compute the sample size for standard win ratio test*

Description

Compute the sample size for standard win ratio test.

Usage

```
WRSS(xi, bparam, q = 0.5, alpha = 0.05, side = 2, power = 0.8)
```

Arguments

xi	A bivariate vector of hypothesized component-wise (treatment-to-control) log-hazard ratios under the Gumbel–Hougaard copula model described in base.
bparam	A list containing baseline parameters zeta2 for ζ_0^2 and delta for $\boldsymbol{\delta}_0$; Can directly use the output of base.
q	Proportion of patients assigned to treatment.
alpha	Type I error rate.
side	2-sided or 1-sided test.
power	Target power.

Value

A list containing n, the computed sample size.

References

Mao, L., Kim, K. and Miao, X. (2021). Sample size formula for general win ratio analysis. Biometrics, https://doi.org/10.1111/biom.13501.

See Also

gumbel.est, base

Examples

```
# The following is not run in package checking to save time.
## Not run:
## load the package and pilot dataset
library(WR)
head(hfaction_cpx9)
dat <- hfaction_cpx9
## subset to control group
pilot <- dat[dat$trt_ab == 0, ]

## get the data ready for gumbel.est()
id <- pilot$patid
## convert time from month to year
time <- pilot$time / 12
status <- pilot$status
## compute the baseline parameters for the Gumbel--Hougaard
## copula for death and hospitalization
gum <- gumbel.est(id, time, status)

## get the baseline parameters
lambda_D <- gum$lambda_D
lambda_H <- gum$lambda_H
kappa <- gum$kappa
## set up design parameters and use base()
## to calculate bparam for WRSS()
# max follow-up 4 years
tau <- 4
# 3 years of initial accrual
tau_b <- 3
# loss to follow-up rate
lambda_L <- 0.05
# compute the baseline parameters
bparam <- base(lambda_D, lambda_H, kappa, tau_b, tau, lambda_L)
bparam

## sample size with power=0.8 under hazard ratios
## 0.9 and 0.8 for death and hospitalization, respectively.
WRSS(
  xi = log(c(0.9, 0.8)), bparam = bparam, q = 0.5, alpha = 0.05,
  power = 0.8
)$n
## sample size under the same set-up but with power 0.9
WRSS(
  xi = log(c(0.9, 0.8)), bparam = bparam, q = 0.5, alpha = 0.05,
  power = 0.9
)$n

## End(Not run)
```

Index