Package ‘nphPower’

July 22, 2025

Title Sample Size Calculation under Non-Proportional Hazards

Version 1.1.0

Description Performs combination tests and sample size calculation for
fixed design with survival endpoints using combination tests under either
proportional hazards or non-proportional hazards. The combination tests
include maximum weighted log-rank test and projection test. The sample
size calculation procedure is very flexible, allowing for user-defined
hazard ratio function and considering various trial conditions like
staggered entry, drop-out etc. The sample size calculation also applies to
various cure models such as proportional hazards cure model, cure model with
(random) delayed treatments effects. Trial simulation function is also provided
to facilitate the empirical power calculation. The references for
projection test and maximum weighted logrank test include Brendel et al. (2014)
<doi:10.1111/sjos.12059> and Cheng and He (2021) <doi:10.48550/arXiv.2110.03833>. The
references for sample size calculation under proportional hazard include
Schoenfeld (1981) <doi:10.1093/biomet/68.1.316> and Freedman (1982) <doi:10.1002/sim.4780010204>.
The references for calculation under non-proportional hazards include
Lakatos (1988) <doi:10.2307/2531910> and Cheng and He (2023) <doi:10.1002/bimj.202100403>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Imports survival, stats, mvtnorm, MASS, zoo

Suggests rmarkdown, knitr

URL `https://github.com/hcheng99/nphPower`

Depends R (>= 2.10)

NeedsCompilation no

Author Huan Cheng [aut, cre] (ORCID: <`https://orcid.org/0000-0002-2552-7656`>)

Maintainer Huan Cheng <`hcheng1118@hotmail.com`>

Repository CRAN

Date/Publication 2024-03-28 08:20:02 UTC

Contents

`cal_event` *Event Rate Calculation*

Description

Calculate the event rate given the hazards and drop-out distribution parameters

Usage

```
cal_event(ratio, lambda1, lambda0, entry, fup, l_shape, l_scale)
```

Arguments

`ratio`	allocation ratio
`lambda1`	hazard rate for treatment group
`lambda0`	hazard rate for control group
`entry`	enrollment period time
`fup`	follow-up period time
`l_shape`	shape parameter of weibull distribution for drop-out
`l_scale`	scale parameter of weibull distribution for drop-out

Details

The event rate is calculated based on the following assumptions: 1) patients are uniformly enrolled within `entry` time; 2) survival times for treatment and control are from exponential distribution; 3) the drop-out times for treatment and control follow the weibull distribution. The final rate is obtained via numeric integration:

$$P = \int_{t_{fup}}^{t_{enrl}+t_{fup}} \Big\{ \int_0^t r(u) exp\big[- \int_0^u [r(x) + l(x)] dx \big] d(u) \Big\} \frac{1}{t_{enrl}} dt$$

where $r(x)$ is the hazard of event and $l(x)$ is the hazard of drop-out; t_{enrl} is the entry time and t_{fup} is the follow-up duration.

Value

a list of components:

`ep1`	event rate for treatment group
`ep0`	event rate for control group
`ep`	mean event rate weighted by the randomization ratio

Examples

```
# median survival time for treatment and control: 16 months vs 12  months
# entry time: 12 months ; follow-up time: 18 months
# the shape parameter for weibull drop-out : 0.5
# median time for drop-out : 48 =>
# scale parameter: 48/log(2)^(1/0.5)=100
  RR <- 1; l1 <- log(2)/16; l0 <- log(2)/12
  t_enrl <- 12; t_fup <- 18

  cal_event(1,l1,l0,t_enrl,t_fup,0.5,100)
```

`cureHR` *Control hazard and hazard ratio generation function*

Description

Generate control hazard and hazard ratio function used for sample size calculation for cure model

Usage

```
cureHR(pi0, pi1 = NULL, k0, lmd0, theta, HRType, tchg = NULL)
```

Arguments

`pi0`	cure rate for the control group
`pi1`	cure rate for the treatment group, Default: NULL
`k0`	shape parameter of the Weibull distribution for the control group
`lmd0`	rate parameter of the Weibull distribution for the control group
`theta`	hazard ratio function
`HRType`	hazard ratio function type. `susceptible` indicates the hazard ratio function applies to the susceptible only; `overall` indicates the hazard ratio function applies to the overall population; `delayed` indicates a cure model with delayed treatment effects. See details.
`tchg`	delayed timepoint for `HRType` = `delayed`, Default: NULL

Details

DETAILS The control group has a survival function of $S_o0 = \pi_0 + (1 - \pi_0)S_0$,where π_0 is the cure rate and S_0 is the survival function for the susceptible population. For `HRType` = `susceptible`, the user also needs to provide the cure rate for the experimental group. The provided hazard ratio applies to the susceptible population only. The returned hazard ratio function is the overall one. For `HRType=delayed`, the returned hazard ratio is derived based on the paper of Wei and Wu (2020) .

Value

a list of components including

`ctrl_hr`	a hazard function for the control group
`hr`	a hazard ratio function

References

Wei, J. and Wu, J., 2020. Cancer immunotherapy trial design with cure rate and delayed treatment effect. Statistics in medicine, 39(6), pp.698-708.

See Also

`integrate`

Examples

```
p0 <- 0.2; p1 <- 0.3; param <- c(1, log(2)/12);
theta_eg <-function(t){t^0*0.7}
fit <- cureHR(p0, p1, param[1], param[2],theta_eg, HRType="susceptible")
# with delayed effects
theta_eg2 <- function(t){(t<=9)+(t>9)*0.7}
fit2 <- cureHR(p0, p1, param[1], param[2],theta_eg2, HRType="delayed", tchg=9)
```

`evalfup` *Visualization of the Relationship between Follow-up and Sample Size*

Description

`evalfup` function displays the graph showing the relationship between the follow-up time and the total sample size/event number required to achieve the the same power

Usage

```
evalfup(
  object,
  lower.time,
  upper.time,
  size,
  increment = 0.5,
  xlabel = "Follow-up Time",
  ylabel = "Total Sample Size/Event Number",
  title = "Relationship between Follow-up and \n Total Sample Size"
)
```

Arguments

`object`	returned object by function `pwr2n.NPH`
`lower.time`	a numeric value specifying the shortest duration time
`upper.time`	a numeric value specifying the longest duration time
`size`	an integer specifying the planned total sample size
`increment`	a numeric value specifying an increment number used for creating a sequence of duration times in plotting, Default: 0.5
`xlabel`	a text for labeling the x axis in the plot, Default: 'Follow-up Time'
`ylabel`	a text for labeling the y axis in the plot, Default: 'Total Sample Size'
`title`	a text for title in the plot: 'Relationship between Follow-up and Total Sample Size'

Details

The `evalfun` function helps to evaluate the relationship between sample size/event number and follow-up duration. It retrieves the trial design information from the `object` returned by `pwr2n.NPH` function. A sequence of follow-up times starting from `lower.time` and ending with `upper.time` are generated. The number of subjects and number of events required for achieving the specified power in `object` are calculated at each time point. An interpolation function `approx` from **stats** is applied to smooth the curves. In case of proportional hazards, the follow-up duration has little impact on the event number except for variations from numeric approximations, while in case of nonproportional hazards, the follow-up time imposes an important impact on both the total sample size and event number.

Value

a graph showing the relationship and a list of components:

`approx.time`	approximate follow-up time corresponding to specified sample size to reach the same target power
`original`	a list with elements of `x` and `y`. Vector `x` contains the follow-up duration and vector `y` contains the corresponding sample size
`interp`	a list containing the interpolated `x` and `y` included in `original`
`Esize`	a vector of events number corresponding to `x` in `original`

Examples

```
# The following code takes more than 5 seconds to run.

# define design parameters
  t_enrl <- 12; t_fup <- 18; lmd0 <- log(2)/12
# define hazard ratio function
  f_hr_delay <- function(x){(x<=6)+(x>6)*0.75}
# define control hazard
  f_haz0 <- function(x){lmd0*x^0}
# perform sample size calculation using logrank test
# generate weight for test
  wlr <- gen.wgt(method="LR")
  snph1 <- pwr2n.NPH(entry = t_enrl, fup = t_fup, Wlist = wlr,
                     k = 100, ratio = 2, CtrlHaz = f_haz0, hazR = f_hr_delay)

# suppose the follow-up duration that are taken into consideration ranges
# from 12 to 24. The planned number of patients to recruit 2200.
# draw the graph
  efun <- evalfup(snph1,lower.time = 12, upper.time = 24, size = 2200,
                title = NULL)
```

`gen.wgt` *Weight Function Generation*

Description

Generate commonly used weight functions for `MaxLRtest` function or `pwr2n.NPH` function

Usage

```
gen.wgt(method = c("LR"), param, theta = 0.5)
```

Arguments

`method`	a vector of text specifying the method(s). The method(s) must be one or some of c(`"LR", "FH", "Wilcoxon", "Tarone","Maxcombo","Maxcross"`). Default: c("LR")
`param`	a vector of length 2. If `FH` method is selected, ρ and γ parameters must be provided, Default: 1
`theta`	a value within (0,1). If method `Maxcross` is selected, `theta` should be specified. See details. Default: 0.5

Details

The weight function for Fleming-Harrington (FH) test is $S(t)^{\rho}(1 - S(t)^{\gamma})$. If `FH` test is specified, both ρ and γ should be provided. The weight for Tarone and Ware test is $y(t)^{1/2}$, where $y(t)$ is number of subjects at risk. The weight for Wilcoxon test is $y(t)$. See Klein (2003) for more details about all those tests. Both Maxcombo test and test proposed by Cheng and He (2021) need four weight functions. Cheng's method is more sensitive in detecting crossing hazards. A nuisance parameter `theta` is required to be specified. Parameter `theta` represents the Cumulative Density Function (CDF) at the crossing time point. If the hazards crossing occurs when few events occur yet, a small value should be chosen. The default value is 0.5.

Function `MaxLRtest` supports different base functions including pooled Kaplan-Meier (K-M) version of CDF functions rather than K-M survival functions. Therefore, if a F(0,1) test is requested, the returned function is `function(x) {x}`, where x denotes the estimated CDF for `KM` base. All the supported base functions are increasing over time.

Value

a list of weight function(s).

References

Klein, J. P., & Moeschberger, M. L. (2003). Survival analysis: techniques for censored and truncated data (Vol. 1230). New York: Springer.

Cheng, H., & He, J. (2021). A Maximum Weighted Logrank Test in Detecting Crossing Hazards. arXiv preprint arXiv:2110.03833.

See Also

`MaxLRtest`, `pwr2n.NPH`

Examples

```
#logrank test
gen.wgt(method="LR")
# FH and logrank test
  fn <- gen.wgt(method=c("FH","LR"), param = c(1,1))
# maximum weighted logrank test proposed by Cheng, including weight
# for detecting crossing hazards
  wcross <- gen.wgt(method="Maxcross", theta = c(0.2))
```

`lung` *Lung cancer data set*

Description

Survival in patients with lung cancer presented in Appendix of Kalbfleisch and Prentice (1980)

Usage

```
lung
```

Format

An object of class `data.frame` with 137 rows and 10 columns.

Details

Therapy Type of treatment: standard or test

Cell Cell type

SurvTime Failure or censoring time

DiagTime Months till randomization

Age Age in years

Prior Prior treatment?:0=no,10=yes

Treatment Treatment indicator: 0=standard,1=test

censor Censor indicator: 1=censor,0=event

References

Kalbfleisch, J. D., and Prentice, R. L. (1980). The Statistical Analysis of Failure Time Data. New York: John Wiley & Sons.

`MaxLRtest` *Maximum Weighted Logrank Test*

Description

`MaxLRtest` performs the maximum weighted logrank test if multiple weight functions are provided. It is the regular weighted logrank test, if a single weight function is specified,

Usage

```
MaxLRtest(
  dat,
  Wlist,
  base = c("KM"),
  alpha = 0.05,
  alternative = c("two.sided")
)
```

Arguments

`dat`	a dataframe or matrix. The first three columns of the data set are survival time, event status indicator and group label. The status indicator, normally 0=alive, 1=dead/event. Other choices are TRUE/FALSE (TRUE=death) or 1/2 (2=death). The group label can be either numeric values like 0=control, 1=treatment or text like C=control, T=treatment.
`Wlist`	a list with components of weight functions
`base`	a text must be one of c("`KM`", "`Combined`", "`N`"), Default: c("KM")
`alpha`	a number indicating type I error rate, Default: 0.05
`alternative`	a text must be one of c("`two.sided`", "`less`", "`greater`"), indicating the alternative hypothesis, Default: c("two.sided")

Details

`MaxLRtest` function performs logrank, weighted logrank test such as Fleming-Harrington test and maximum weighted logrank test depending on the type and number of weight functions. Let $w(x_t)$ denote the weight applied at event time point t, where x_t is the base function. There are three options for `base`. If `KM` is used, $x_t = 1 - S_t$, where S_t is pooled Kaplan-Meier estimate of survival rate at time point t. A FH(1,0) test needs a weight function $1 - x_t$. If `Combined` base is selected, $x_t = 1 - S_t^*$, where $S_t^* = w_1 S_t^1 + w_0 S_t^0$, the weighted average of KM estimate of survival rate for treatment (S_t^1) and control group (S_t^0). It is considered more robust in case of unbalanced data. For option `N`, $x_t = 1 - \frac{Y_t}{N}$, where Y_t is the subjects at risk at time t and N is the total number of subjects.The Wilcoxon and tarone test should use this base. The base x_t in all three cases is an increasing function of time t. Function `gen.wgt` helps to generate the commonly used weight functions.

Let Λ_1 and Λ_0 denote the cumulative hazard for treatment and control group. The alternative of a two-sided test is $H_a : \Lambda_1 \neq \Lambda_0$. The `"less"` alternative corresponds to $H_a : \Lambda_1 < \Lambda_0$ and the `"greater"` alternative is $H_a : \Lambda_1 > \Lambda_0$.

A p-value is obtained from a multivariate normal distribution if multiple weights are provided. The function `pmvnorm` from R package **mvtnorm** is used. Because the algorithm is slightly seed-dependent,the p-value and critical value is the average of 10 runs.

Value

a list of components including

`stat`	a numeric value indicating the test statistic. It is logrank or weighted logrank test statistic if one weight function is specified. Otherwise, it gives the maximum weighted logrank test statistic, which takes the maximum of absolute values of all the statistics.
`stat.mat`	a matrix with the first column showing weighted logrank test statistics and other columns displaying the variance and covariance between statistics
`critV`	a numeric value indicating the critical value corresponding to the nominal level - `alpha`
`details`	a dataframe showing the intermediate variables used in the calculation.
`p.value`	a numeric value indicating the p-value of the test

See Also

`pwr2n.NPH`, `gen.wgt`

Examples

```
data(lung)
#Only keep variables for analysis
tmpd <- with(lung, data.frame(time=SurvTime,stat=1-censor,grp=Treatment))
#logrank test
wlr <- gen.wgt(method = "LR")
t1 <- MaxLRtest(tmpd, Wlist = wlr, base = c("KM") )
t1$stat ;t1$p.value


# maxcombo test
wmax <- gen.wgt(method="Maxcombo")
t2 <- MaxLRtest(tmpd, Wlist = wmax, base = c("KM") )
t2$stat ;t2$p.value
#visualize the weight functions
plot(t2)
```

`n2pwr.NPH` *Power Calculation with Combination Test*

Description

`n2pwr.NPH` calculates the power given either the number of events or number of subjects using combination test

Usage

```
n2pwr.NPH(
  method = "MaxLR",
  entry = 1,
  fup = 1,
```

```
  maxfup = entry + fup,
  CtrlHaz,
  hazR,
  transP1,
  transP0,
  Wlist,
  entry_pdf0 = function(x) {
     (1/entry) * (x >= 0 & x <= entry)
 },
  entry_pdf1 = entry_pdf0,
  eventN = NULL,
  totalN = NULL,
  ratio = 1,
  alpha = 0.05,
  alternative = c("two.sided"),
  k = 100,
  nreps = 10
)
```

Arguments

`method`	a text specifying the calculation method, either `"MaxLR"` or `"Projection"`. Maximum weighted logrank test is used if `"MaxLR"` is specified; otherwise, projection test is used.
`entry`	a numeric value indicating the enrollment time, Default: 1
`fup`	a numeric value indicating the minimum follow-up time for subjects. , Default: 1
`maxfup`	maximum follow-up time
`CtrlHaz`	a function, specifying the hazard function for control group.
`hazR`	a function, specifying the hazard ratio function between treatment and control group
`transP1`	a numeric vector of length 2, consisting of the transition probability from receiving treatment to drop-out (drop-out rate) and from receiving treatment to receiving control (drop-in rate) per time unit.
`transP0`	a numeric vector of length 2, consisting of the transition probability from receiving control to drop-out (drop-out rate) and from receiving control to receiving treatment (drop-in rate) per time unit.
`Wlist`	a list, consisting of weight functions applied to the test. The element of the list must be functions. Default is a list of one constant function, corresponding to the logrank test.
`entry_pdf0`	a function, indicating the probability density function (pdf) of enrollment/entry time for control group. The default assumes a uniform distribution corresponding to the constant enrollment rate.
`entry_pdf1`	a pdf function of enrollment/entry time for treatment
`eventN`	the number of events

`totalN`	the number of subjects
`ratio`	allocation ratio, Default: 1
`alpha`	type i error, Default: 0.05
`alternative`	alternative hypothesis - one of c(`"two.sided"`, `"less"`, `"greater"`), Default: `"two.sided"`
`k`	an integer, indicating number of sub-intervals per time unit, Default: 100
`nreps`	number of replicates used for calculating quantitle using multivariate normal

Details

Function `npwr.NPH` calculates the asymptotic power given number of events or number of subjects using maximum weighted logrank test or projection type test. If only `eventN` is provided, the asymptotic power is based on provided number of events. If only `totalN` is given, the pooled event probability ($eprob$) is calculated according input design parameters including entry time, follow-up time and hazard functions, etc. The number of events is calculated as `totalN`*$eprob$, which is given in returned vector `outN`. Similarly, if only `eventN` is given, the total sample size is given as `eventN`/$eprob$. However, if both `eventN` and `totalN` are provided, we only use `eventN` for calculation. Check function `pwr2n.NPH` for more calculation details.

Value

a list of components:

`power`	asymptotic power
`inN`	a vector consisting of the input of `eventN` and `totalN`
`outN`	a vector including the output of number of events and total sample. See details.
`prob_event`	event probability at the end of trial
`L_trans`	a list, consisting of transition matrix at each interval
`pdat`	a data frame including all the intermediate variables in the calculation.
`studytime`	a vector of length 2, including the entry and follow-up time as input
`RandomizationRatio`	as input

See Also

`pwr2n.NPH`

Examples

```
# entry time
t_enrl <- 12
# follow-up time
t_fup <- 18
# baseline hazard
lmd0 <- -log(0.2)/10
# delayed treatment effects
f_hr_delay <- function(x){(x<=6)+(x>6)*0.75}
```

```
# maxcombo test
maxc <- gen.wgt(method="Maxcombo")
pwr1 <- n2pwr.NPH(entry   = t_enrl
                 ,fup      = t_fup
                 ,CtrlHaz = function(x){x^0*lmd0}
                 ,hazR = f_hr_delay
                 ,transP1 = c(0,0)
                 ,transP0 = c(0,0)
                 ,Wlist = maxc
                 ,eventN = 50 # targeted number of events
)
```

`plot.MaxLR` *Graphical Display of Weight Functions*

Description

Display weight functions used in the function `MaxLRtest`

Usage

```
## S3 method for class 'MaxLR'
plot(x, ...)
```

Arguments

`x`	object of `MaxLRtest` function
`...`	additional graphical arguments passed to the plot function

Value

Plots are produced on the current graphics device

See Also

`MaxLRtest`

Examples

```
# See examples in the help file of function MaxLRtest
```

`plot.NPHpwr` *Graphical Display of Design Parameters in Sample Size Calculation*

Description

Displays graphs of survival, hazards, drop-out and censor over time as specified in the calculation.

Usage

```
## S3 method for class 'NPHpwr'
plot(x, type = c("hazard", "survival", "dropout", "event", "censor"), ...)
```

Arguments

`x`	object of the `pwr2n.NPH` function
`type`	a vector of string, specifying the graphs to display. The options include "`hazard`", "`survival`", "`dropout`", "`event`", and "`censor`". If `type` is not provided, all the available graphs are generated.
`...`	additional graphical arguments passed to the plot function

Details

The `type` argument provides five options to visualize the trial in design. Option `survival` shows the survival probabilities of treatment and control group over time. Option `hazard` provides the hazard rates and hazard ratio over time. Option `dropout` shows the proportion of drop-out subjects across the trial duration. Option `censor` shows the proportion of censored subjects over time.

Value

plots are produced on the current graphics device

See Also

`pwr2n.NPH`

Examples

```
# generate weight function
wlr <- gen.wgt(method = "LR" )
t_enrl <- 12; t_fup <- 18; lmd0 <- log(2)/12
# delayed treatment effects, the crossign point is at 6.
f_hr_delay <- function(x){(x<=6)+(x>6)*0.75}
f_haz0 <- function(x){lmd0*x^0}
snph1 <- pwr2n.NPH(entry = t_enrl, fup = t_fup, Wlist = wlr,
                   k = 100, ratio = 2, CtrlHaz = f_haz0,
                   hazR = f_hr_delay)
# display the hazards plot only
plot(snph1, type="hazard")
```

```
# display all plots
plot(snph1)
```

`plotHazSurv` *Graphic Display of Hazard and Survival Function*

Description

Plot the hazard and survival function of the of control group (from weibull or loglogistic distribution) and treatment group (derived from an arbitrary hazard ratio function)

Usage

```
plotHazSurv(
  bsl_dist = c("weibull", "loglogistic"),
  param = c(1.2, 0.03),
  fun_list,
  end,
  tit = c("Hazard Function", "Survival Function"),
  pos = c(1, 2),
  hlegend.loc = "bottomleft",
  slegend.loc = "topright"
)
```

Arguments

`bsl_dist`	a text must be one of (`"weibull"`, `"loglogistic"`) distribution, specified for the control group
`param`	a vector of length 2, specifying the shape and rate (1/scale) parameter of the `bsl_dist` distribution, Default: c(1.2, 0.03)
`fun_list`	a list of hazard ratio functions comparing treatment group and control group
`end`	a value specifying the duration of the curve
`tit`	a vector specifying the titles of each graph, Default: c("Hazard Function", "Survival Function")
`pos`	a graphic parameter in the form of c(nr,nc). Subsequent figures will be drawn in an nr-by-nc array, Default: c(1, 2)
`hlegend.loc`	a text indicating the position of legend for the hazard plot. Default: "bottomleft"
`slegend.loc`	a text indicating the position of legend for the survival plot. Default: "topright"

Value

graphics of hazard and survival functions

Examples

```
# proportional hazards
plotHazSurv(
 bsl_dist=c("weibull")
 ,param=c(1.2,1/30)
 ,fun_list=list(function(x){x^0*0.7})
 ,40
 ,tit= c("Hazard Function","Survival Function")
 ,pos=c(1,2)
)
# crossing hazards
plotHazSurv(
 bsl_dist=c("weibull")
 ,param=c(1.2,1/30)
 ,fun_list=list(function(x){1.3*(x<10)+(x>=10)*0.7})
 ,40
 ,tit= c("Hazard Function","Survival Function")
 ,pos=c(1,2)
)
```

`projection.test` *Projection test*

Description

Perform projection test as proposed by Brendel (2014)

Usage

```
projection.test(dat, Wlist, base, alpha = 0.05)
```

Arguments

`dat`	a dataframe or matrix, of which the first three columns are survival time, event status indicator and group label. The status indicator, normally 0=alive, 1=dead/event. Other choices are TRUE/FALSE (TRUE=death) or 1/2 (2=death). The group label can be either numeric values like 0=control, 1=treatment or text like C=control, T=treatment.
`Wlist`	a list object with components of weight functions
`base`	a text must be one of c("KM","`Combined`","`N`"), Default: c("KM")
`alpha`	a number indicating type I error rate, Default: 0.05

Details

The base functions are the same as those described in function `MaxLRtest`. The method detail can be found in Brendel (2014) paper. The main idea is to map the multiple weighted logrank statistics into a chi-square distribution. The degree freedom of the chi-square is the rank of the generalized inverse of covariance matrix. Only two-sided test is supported in the current function.

Value

a list of components including

`chisq`	a numeric value indicating the chi-square statistic
`df.chis`	a numeric value indicating the degree freedom of the test
`pvalue`	a numeric value giving the p-value of the test
`details`	a data frame consisting of statistics from multiple weight functions and the variance-covariance matrix

References

Brendel, M., Janssen, A., Mayer, C. D., & Pauly, M. (2014). Weighted logrank permutation tests for randomly right censored life science data. Scandinavian Journal of Statistics, 41(3), 742-761.

See Also

`MaxLRtest`

Examples

```
# load and prepare data
data(lung)
tmpd <- with(lung, data.frame(time=SurvTime,stat=1-censor,grp=Treatment))
# two weight functions are defined.
# one is constant weight; the other emphasize diverging hazards
timef1 <- function(x){1}
timef2 <- function(x){(x)}
test1 <- projection.test(tmpd,list(timef1,timef2),base="KM")
test1$chisq; test1$pvalue; test1$df.chisq
```

`pwr2n.LR` *Sample Size Calculation under Proportional Hazards*

Description

`pwr2n.LR` calculates the total number of events and total number of subjects required given the provided design parameters based on either schoenfeld or freedman formula.

Usage

```
pwr2n.LR(
  method = c("schoenfeld", "freedman"),
  lambda0,
  lambda1,
  ratio = 1,
  entry = 0,
  fup,
```

```
  alpha = 0.05,
  beta = 0.1,
  alternative = c("two.sided"),
  Lparam = NULL,
  summary = TRUE
)
```

Arguments

`method`	calculation formula, Default: c("schoenfeld", "freedman")
`lambda0`	hazard rate for the control group
`lambda1`	hazard rate for the treatment group
`ratio`	randomization ratio between treatment and control. For example, ratio=2 if randomization ratio is 2:1 to treatment and control group. Default:1
`entry`	enrollment time. A constant enrollment rate is assumed, Default: 0
`fup`	follow-up time.
`alpha`	type I error rate, Default: 0.05
`beta`	type II error rate. For example,if the target power is 80%, beta is 0.2. Default: 0.1
`alternative`	a value must be one of ("two.sided", "one.sided"), indicating whether a two-sided or one-sided test to use. Default: c("two.sided")
`Lparam`	a vector of shape and scale parameters for the drop-out Weibull distribution, See Details below. Default: NULL
`summary`	a logical controlling whether a brief summary is printed or not , Default: TRUE

Details

Both Schoenfeld's formula and Freedman's formula are included in the function `pwr2n.LR`. The total event number is determined by α, β and hazard ratio, i.e., λ_1/λ_0. Other design parameters such as enrollment period affects the event probability and thus the total sample size. A fixed duration design is assumed in the calculation. All patients are enrolled at a constant rate within `entry` time and have at least `fup` time of follow-up. So the total study duration is `entry+fup`. If drop-out is expected, a Weibull distribution with shape parameter -α and scale parameter - β is considered. The CDF of Weibull is $F(x) = 1 - exp(-(x/\beta)^{\alpha})$, where α is the shape parameter and β is the scale parameter. The event rate is calculated through numeric integration. See more details in `cal_event`.

Value

a list of components including

`eventN`	a numeric value giving the total number of events
`totalN`	a numeric value giving the total number of subjects
`summary`	a list containing the input parameters and output results

References

Schoenfeld, D. (1981) The asymptotic properties of nonparametric tests for comparing survival distributions. Biometrika, 68, 316–319.

Freedman, L. S. (1982) Tables of the number of patients required in clinical trials using the logrank test. Statistics in medicine, 1, 121–129.

See Also

`pwr2n.NPH`, `evalfup`, `cal_event`

Examples

```
# define design parameters
l0 <- log(2)/14; HR <- 0.8; RR <- 2; entry <- 12; fup <- 12;
eg1 <- pwr2n.LR( method    = c("schoenfeld")
                 ,l0
                 ,l0*HR
                 ,ratio=RR
                 ,entry
                 ,fup
                 ,alpha     = 0.05
                 ,beta      = 0.1
)
# event number, total subjects, event probability
c(eg1$eventN,eg1$totalN,eg1$eventN/eg1$totalN)

# example 2: drop-out from an exponential with median time is 30
eg2 <- pwr2n.LR( method    = c("schoenfeld")
                 ,l0
                 ,l0*HR
                 ,ratio=RR
                 ,entry
                 ,fup
                 ,alpha     = 0.05
                 ,beta      = 0.1
                 ,Lparam = c(1,30/log(2))
)
# event number, total subjects, event probability
c(eg2$eventN,eg2$totalN,eg2$eventN/eg2$totalN)
```

`pwr2n.NPH` *Sample Size Calculation with Combination Test*

Description

`pwr2n.NPH` calculates the number of events and subjects required to achieve pre-specified power in the setup of two groups. The method extends the calculation in the framework of the Markov model by Lakatos, allowing for using the maximum weighted logrank tests or projection test with an arbitrary number of weight functions. For maximum weighted logrank type test, if only one weight function is provided, the test is essentially the classic (weighted) logrank test.

Usage

```
pwr2n.NPH(
  method = "MaxLR",
  entry = 1,
  fup = 1,
  maxfup = entry + fup,
  CtrlHaz,
  hazR,
  transP1 = c(0, 0),
  transP0 = c(0, 0),
  Wlist = list(function(x) {
     x^0
 }),
  entry_pdf0 = function(x) {
     (1/entry) * (x >= 0 & x <= entry)
 },
  entry_pdf1 = entry_pdf0,
  ratio = 1,
  alpha = 0.05,
  beta = 0.1,
  alternative = c("two.sided"),
  criteria = 500,
  k = 100,
  m = 0,
  nreps = 10,
  varianceType = c("equal"),
  weightBase = "C",
  summary = TRUE
)
```

Arguments

`method`	a text specifying the calculation method, either `"MaxLR"` or `"Projection"`. Maximum weighted logrank test is used if `"MaxLR"` is specified; otherwise, projection test is used.
`entry`	a numeric value indicating the enrollment time, Default: 1
`fup`	a numeric value indicating the minimum follow-up time for subjects. , Default: 1
`maxfup`	maximum follow-up time
`CtrlHaz`	a function, specifying the hazard function for control group.
`hazR`	a function, specifying the hazard ratio function between treatment and control group
`transP1`	a numeric vector of length 2, consisting of the transition probability from receiving treatment to drop-out (drop-out rate) and from receiving treatment to receiving control (drop-in rate) per time unit.

`transP0`	a numeric vector of length 2, consisting of the transition probability from receiving control to drop-out (drop-out rate) and from receiving control to receiving treatment (drop-in rate) per time unit.
`Wlist`	a list, consisting of weight functions applied to the test. The element of the list must be functions. Default is a list of one constant function, corresponding to the logrank test.
`entry_pdf0`	a function, indicating the probability density function (pdf) of enrollment time for control group. The default assumes a uniform distribution corresponding to the constant enrollment rate.
`entry_pdf1`	a pdf of enrollment time for treatment group. See `entry_pdf0`, Default: assume same pdf as control group.
`ratio`	an integer, indicating the randomization ratio between treatment and control group, Default: 1
`alpha`	type I error rate, Default: 0.05
`beta`	type II error rate, Default: 0.1
`alternative`	a character string specifying the alternative hypothesis, must be one of `"two.sided"`, `"greater"`,`"less"`. See details. For `"Projection"` method, only `"two-sided"` alternative is supported. Default: `c("two.sided")`
`criteria`	an integer indicating the maximum iteration allowed in obtaining the number of events. See details , Default: 500
`k`	an integer, indicating number of sub-intervals per time unit, Default: 100
`m`	a value within 0 and 1.
`nreps`	an integer, indicating number of iterations in calculating the quantile of multivariate normal. See Details.
`varianceType`	Default: `equal`. Indicates different variance assumptions for the sample size calculation. It is not applicable for the maximum weighted logrank test. See details.
`weightBase`	A character, either `"F"` or `"T"`. `F` indicates a CDF is the base for the weight function used in the weighted logrank or maximum weighted logrank test. `T` indicates time is the base for weight function. Default: `F`
`summary`	a logical value, controlling whether to print the summary of calculation, Default: TRUE

Details

The detailed methods can be found in the reference papers. The number of subjects is determined by several factors, including the control hazard function, hazard ratio function, entry time distribution, follow-up time, etc. Under proportional hazard assumption, the number of events is mainly determined by the hazard ratio besides type i/ii error rates. However, under nonproportional hazards, all the above design parameters may have an impact on the number of events. The study design assumes `entry` time units of enrollment and at least `fup` time units of follow-up. If enrollment time `entry` is set to zero, all subjects are enrolled simultaneously, so there is no staggered entry. Otherwise, if `entry` is greater than 0, administrative censoring is considered. The user-defined enrollment time function, hazard function for the control group and hazard ratio function can be either discrete or continuous. Various non-proportional hazards types are accommodated. See examples below. If multiple weight functions are provided in `Wlist`, a maximum weighted logrank test or combination test is implemented. An iterative procedure is used to obtain the event number based on the multivariate normal distribution. Package **mvtnorm** is used to calculate the quantiles. Because the algorithm is slightly seed dependent, the quantiles are mean values of ten replicates by default. The number of replicates is controlled by argument `ninter`.

The `"alternative"` option supports both two-sided and one-sided test. Let Λ_1 and Λ_0 denote the cumulative hazard of treatment and control group. The `less` option tests $H_0 : \Lambda_1 > \Lambda_0$ against $H_a : \Lambda_1 <= \Lambda_0$. The `greater` option tests $H_0 : \Lambda_1 < \Lambda_0$ against $H_a : \Lambda_1 >= \Lambda_0$.

When `varianceType` is `equal`, the sample size for a two sided test is $(z_{1-\alpha/2} + z_{1-\beta})^2\tilde{\sigma}^2/\mu_w^2$, where $\tilde{\sigma}^2$ is the variance estimate under alternative. when `varianceType` is not `equal`. The formula is $(z_{1-\alpha/2}\sigma_w + z_{1-\beta}\tilde{\sigma})^2/\mu_w^2$. Please use `equal` variance type for the maximum weighted logrank test.

Value

An object of class `"NPHpwr"` with corresponding `plot` function. The object is a list containing the following components:

`eventN`	total number of events
`totalN`	total number of subjects
`pwr`	actual power given the number of events
`prob_event`	event probability at the end of trial
`prob1`	event probability for the treatment group
`prob0`	event probability for the control group
`L_trans`	a list, consisting of transition matrix at each interval
`pdat`	a dataframe including all the intermediate variables in the calculation. see Details.
`studytime`	a vector of length 2, including the entry and follow-up time as input
`RandomizationRatio`	as input
`eventlist`	a vector containing the number of events using each weight function alone
`inputfun`	a list containing all the input functions specified by users

References

Brendel, M., Janssen, A., Mayer, C. D., & Pauly, M. (2014). Weighted logrank permutation tests for randomly right censored life science data. Scandinavian Journal of Statistics, 41(3), 742-761.

Cheng, H., & He, J. (2021). A Maximum Weighted Logrank Test in Detecting Crossing Hazards. arXiv preprint arXiv:2110.03833.

Cheng H, He J. Sample size calculation for the combination test under nonproportional hazards. Biom J. 2023 Apr;65(4):e2100403. doi: 10.1002/bimj.202100403. Epub 2023 Feb 15. PMID: 36789566

See Also

`pwr2n.LR` `gen.wgt`, `evalfup`

Examples

```
#------------------------------------------------------------
## Delayed treatment effects using maxcombo test
## generate a list of weight functions for maxcombot test
wmax <- gen.wgt(method = "Maxcombo" )
t_enrl <- 12; t_fup <- 18; lmd0 <- log(2)/12
## delayed treatment effects
f_hr_delay <- function(x){(x<=6)+(x>6)*0.75}
f_haz0 <- function(x){lmd0*x^0}
##  The following code takes more than 5 seconds to run

snph1 <- pwr2n.NPH(entry = t_enrl, fup = t_fup, Wlist = wmax,
                   k = 100, ratio = 2, CtrlHaz = f_haz0, hazR = f_hr_delay)

#-------------------------------------------------------------
# same setting using projection test
snph2 <- pwr2n.NPH(method = "Projection", entry = t_enrl,
 fup = t_fup, Wlist = wmax, k = 10, ratio = 2, CtrlHaz = f_haz0,
 hazR = f_hr_delay)

#-------------------------------------------------------------
#proportional hazards with weibull survival for control group
#logrank test
wlr <- gen.wgt(method = "LR" )
b0 <- 3
th0 <- 10/(-log(0.2))^(1/b0)
#Weibull hazard function
f_hz_weibull <- function(x){b0/th0^b0*x^(b0-1)}
#hazard ratio function
f_hr <- function(x){0.5*x^0}
# define entry and follow-up time
t_enrl <- 5; t_fup <- 5
exph1 <- pwr2n.NPH(entry = t_enrl, fup = t_fup, k = 100,
Wlist = wlr,  CtrlHaz = f_hz_weibull, hazR = f_hr, summary = FALSE)
summary(exph1)
```

`simu.trial` *Simulate Survival Trial Data*

Description

`simu.trial` simulates survival data allowing flexible input of design parameters. It supports both event-driven design and fixed study duration design.

Usage

```
simu.trial(
  type = c("event", "time"),
  trial_param,
  bsl_dist = c("weibull", "loglogistic", "mix-weibull"),
  bsl_param,
  drop_param0,
  drop_param1 = drop_param0,
  entry_pdf0 = function(x) {
     (1/trial_param[2]) * (x >= 0 & x <= trial_param[2])
 },
  entry_pdf1 = entry_pdf0,
  enrollmentType = NULL,
  entryP = list(10000, 1),
  HR_fun,
  ratio,
  cureModel = NULL,
  cureRate1 = NULL,
  HR_data = NULL,
  upInt = 100,
  summary = TRUE
)
```

Arguments

`type`	indicates whether event-driven trial ("`event`) or fixed study duration trial ("`time`"), Option: c("`event`", "`time`")
`trial_param`	a vector of length 3 with components for required subject size, enrollment time and required number of events ("`event`" type trial)/follow-up time ("`time`" type trial)
`bsl_dist`	indicates the survival distribution for control group, option: `c`("`weibull`", "`loglogistic`",`mix-weibull`)
`bsl_param`	a vector of length 2 with the shape and rate (scale) parameter for the weibull or loglogistic survival distribution of control group. A vector of length 3 with shape, rate and cure rate for the mix-weibull distribution. See details.
`drop_param0`	a vector of length 2 with shape and scale parameter for the weibull distribution of drop-out time for control group
`drop_param1`	a vector of length 2 with shape and scale parameter for the weibull distribution of drop-out time for treatment group
`entry_pdf0`	a function describing the pdf of the entry time for control. Default: uniform enrollment
`entry_pdf1`	a function describing the pdf of the entry time for treatment.
`enrollmentType`	default value is NULL, indicating a entry time follows specified distribution. Specify "`piecewise uniform`", indicating entry time follows piecewise uniform
`entryP`	if `enrollmentType` is piecewise uniform. `entryP` should be provided with a list containing the enrollment rate at each interval

`HR_fun`	a function describing the hazard ratio function between treatment and control group
`ratio`	allocation ratio between treatment and control group. For example, `ratio`=2 if 2:1 allocation is used.
`cureModel`	specifies the cure model. "`PHCM`" and "`PHCRM`".
`cureRate1`	specifies the cure rate for the susceptible population in the experimental group if the cure model is `PHCM`.
`HR_data`	a matrix consisting of covariates values
`upInt`	a value indicating the upper bound used in the `uniroot` function. See details. Default: 100
`summary`	a logical indicating whether basic information summary is printed to the console or not, Default: TRUE

Details

The loglogistic distribution for the event time has the survival function $S(x) = b^a/(b^a + x^a)$ and hazard function $\lambda(x) = a/b(x/b)^{a-1}/(1 + (t/b)^a)$, where a is the shape parameter and b is the scale parameter. The weibull distribution for event time and drop-out time has the survival function $S(x) = exp(-(xb)^a)$ and hazard function $\lambda(x) = ab(xb)^{a-1}$, where a is the shape parameter and b is the rate parameter. The median of weibull distribution is $(ln(2)^{1/a}/b)$. If drop out or loss to follow-up are do not need to be considered, a very small rate parameter b can be chosen such that the median time is greatly larger than the study duration. The default entry time is uniformly distributed within the enrollment period by default. Other options are allowed through providing the density function.

The `simu.trial` function simulates survival times for control and treatment groups separately. The control survival times are drawn from standard parametric distribution (Weibull, Loglogistic). Let $\lambda_1(t)$ and $\lambda_0(t)$ denote the hazard function for treatment and control. It is assumed that $\lambda_1(t)/\lambda_0(t) = g(t)$, where $g(t)$ is the user-defined function, describing the change of hazard ratio over time. In case of proportional hazards, $g(t)$ is a constant. The survival function for treatment group is $S_1(t) = exp(-\int_0^t g(s)\lambda_0(s)ds)$. The Survival time T is given by $T = S_1^{(} - 1)(U)$, where U is drawn from uniform (0,1). More details can be found in Bender, et al. (2005). Function `uniroot` from `stats` package is used to generate the random variable. The argument `upInt` in `simu.trial` function corresponds to the upper end point of the search interval and it can be adjusted by user if the default value 100 is not appropriate. More details can be found in help file of `uniroot` function.

Value

A list containing the following components

`data`: a dataframe (simulated dataset) with columns:

id identifier number from 1:n, n is the total sample size

group group variable with 1 indicating treatment and 0 indicating control

aval observed survival time, the earliest time among event time, drop-out time and end of study time

cnsr censoring indicator with 1 indicating censor and 0 indicating event

cnsr.desc description of the `cnsr` status, including three options- drop-out, event and end of study. Both drop-out and end of study are considered as censor.

event event indicator with 1 indicating event and 0 indicating censor

entry.time time when the patient is enrolled in the study

a list of summary information of the trial including

`type` a character indicating input design type - `event` or `time`

`entrytime` a number indicating input enrollment period

`maxob` a number indicating the maximum study duration. For `time` type of design, the value is equal to the enrollment period plus the follow-up period. For `event` type of design, the value is the calendar time of the last event.

References

Bender, R., Augustin, T., & Blettner, M. (2005). Generating survival times to simulate Cox proportional hazards models. Statistics in medicine, 24(11), 1713-1723.

See Also

`Weibull`, `integrate`, `Logistic`, `Uniform`, `optimize`, `uniroot`

Examples

```
# total sample size
N <- 300
# target event
E <- 100
# allocation ratio
RR <- 2
# enrollment time
entry <- 12
# follow-up time
fup <- 18
# shape and scale parameter of weibull for entry time
b_weibull <- c(1,log(2)/18)
# shape and scale parameter of weibull for drop-out time
drop_weibull <- c(1,log(2)/30)
# hazard ratio function (constant)
HRf <- function(x){0.8*x^0}

### event-driven trial
set.seed(123445)
data1 <- simu.trial(type="event",trial_param=c(N,entry,E),bsl_dist="weibull",
                    bsl_param=b_weibull,drop_param0=drop_weibull,HR_fun=HRf,
                    ratio=RR)

### fixed study duration
set.seed(123445)
data2 <- simu.trial(type="time",trial_param=c(N,entry,fup),bsl_dist="weibull",
                    bsl_param=b_weibull,drop_param0=drop_weibull,HR_fun=HRf,
                    ratio=RR)
```

`summary.NPHpwr` *Summary of the* `pwr2n.NPH` *function*

Description

Summarize and print the results of `pwr2n.NPH` function

Usage

```
## S3 method for class 'NPHpwr'
summary(object, ...)
```

Arguments

`object`	object of the `pwr2n.NPH` function
`...`	additional arguments passed to the summary function

Value

No return value. Summary results are printed to Console.

See Also

`pwr2n.NPH`

Index