Package 'lrstat'

August 17, 2023

Type Package

Title Power and Sample Size Calculation for Non-Proportional Hazards

Version 0.1.15

Date 2023-08-17

Author Kaifeng Lu

Maintainer Kaifeng Lu <kaifenglu@gmail.com>

Description Performs power and sample size calculation for non-proportional hazards model using the Fleming-Harrington family of weighted log-rank tests. The sequentially calculated log-rank test score statistics are assumed to have independent increments as characterized in Anastasios A. Tsiatis (1982) <doi:10.1080/01621459.1982.10477898>. The mean and variance of log-rank test score statistics are calculated based on Kaifeng Lu (2021) <doi:10.1002/pst.2069>. The boundary crossing probabilities are calculated using the recursive integration algorithm described in Christopher Jennison and Bruce W. Turnbull (2000, ISBN:0849303168).

License GPL (>= 2)

Imports Rcpp (>= 1.0.9)

LinkingTo Rcpp

Suggests knitr, rmarkdown, testthat (>= 3.0.0), mvtnorm, dplyr, tidyr

VignetteBuilder knitr

RoxygenNote 7.2.3

Encoding UTF-8

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-08-17 18:42:33 UTC

R topics documented:

`lrstat-package` *Power and Sample Size Calculation for Non-Proportional Hazards*

Description

Performs power and sample size calculation for non-proportional hazards model using the Fleming-Harrington family of weighted log-rank tests.

Details

For proportional hazards, the power is determined by the total number of events and the constant hazard ratio along with information rates and spending functions. For non-proportional hazards, the hazard ratio varies over time and the calendar time plays a key role in determining the mean and variance of the log-rank test score statistic. It requires an iterative algorithm to find the calendar time at which the targeted number of events will be reached for each interim analysis. The lrstat package uses the analytic method in Lu (2021) to find the mean and variance of the weighted log-rank test score statistic at each interim analysis. In addition, the package approximates the variance and covariance matrix of the sequentially calculated log-rank test statistics under the alternative

hypothesis with that under the null hypothesis to take advantage of the independent increments structure in Tsiatis (1982) applicable for the Fleming-Harrington family of weighted log-rank tests.

The most useful functions in the package are `lrstat`, `lrpower`, `lrsamplesize`, and `lrsim`, which calculate the mean and variance of log-rank test score statistic at a sequence of given calendar times, the power of the log-rank test, the sample size in terms of accrual duration and follow-up duration, and the log-rank test simulation, respectively. The `accrual` function calculates the number of patients accrued at given calendar times. The `caltime` function finds the calendar times to reach the targeted number of events. The `exitprob` function calculates the stagewise exit probabilities for specified boundaries with a varying mean parameter over time based on an adaptation of the recursive integration algorithm described in Chapter 19 of Jennison and Turnbull (2000).

The development of the lrstat package is heavily influenced by the rpact package. We find their function arguments to be self-explanatory. We have used the same names whenever appropriate so that users familiar with the rpact package can learn the lrstat package quickly. However, there are notable differences:

- lrstat uses direct approximation, while rpact uses the Schoenfeld method for log-rank test power and sample size calculation.
- lrstat uses `accrualDuration` to explicitly set the end of accrual period, while rpact incorporates the end of accrual period in `accrualTime`.
- lrstat considers the trial a failure at the last stage if the log-rank test cannot reject the null hypothesis up to this stage and cannot stop for futility at an earlier stage.
- the `lrsim` function uses the variance of the log-rank test score statistic as the information.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

Anastasios A. Tsiatis. Repeated significance testing for a general class of statistics used in censored survival analysis. J Am Stat Assoc. 1982;77:855-861.

Christopher Jennison, Bruce W. Turnbull. Group Sequential Methods with Applications to Clinical Trials. Chapman & Hall/CRC: Boca Raton, 2000, ISBN:0849303168

Kaifeng Lu. Sample size calculation for logrank test and prediction of number of events over time. Pharm Stat. 2021;20:229-244.

See Also

rpact, gsDesign

Examples

```
lrpower(kMax = 2, informationRates = c(0.8, 1),
        criticalValues = c(2.250, 2.025), accrualIntensity = 20,
        piecewiseSurvivalTime = c(0, 6),
        lambda1 = c(0.0533, 0.0309), lambda2 = c(0.0533, 0.0533),
        gamma1 = 0.00427, gamma2 = 0.00427,
        accrualDuration = 22, followupTime = 18)
```

accrual *Number of enrolled subjects*

Description

Obtains the number of subjects enrolled by given calendar times.

Usage

```
accrual(
  time = NA_real_,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  accrualDuration = NA_real_
)
```

Arguments

time	A vector of calendar times at which to calculate the number of enrolled subjects.
accrualTime	Accrual time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
accrualIntensity	A vector of accrual intensities. One for each accrual time interval.
accrualDuration	Duration of the enrollment period.

Value

A vector of total number of subjects enrolled by the specified calendar times.

Examples

```
# Example 1: Uniform enrollment with 20 patients per month for 12 months.

accrual(time = 3, accrualTime = 0, accrualIntensity = 20,
        accrualDuration = 12)


# Example 2: Piecewise accrual, 10 patients per month for the first
# 3 months, and 20 patients per month thereafter. Patient recruitment
# ends at 12 months for the study.

accrual(time = c(2, 9), accrualTime = c(0, 3),
        accrualIntensity = c(10, 20), accrualDuration = 12)
```

caltime *Calendar times for target number of events*

Description

Obtains the calendar times to reach the target number of subjects having an event.

Usage

```
caltime(
  nevents = NA_real_,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L
)
```

Arguments

nevents	A vector of target number of events.
allocationRatioPlanned	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
accrualTime	Accrual time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
accrualIntensity	A vector of accrual intensities. One for each accrual time interval.
piecewiseSurvivalTime	A vector that specifies the time intervals for the piecewise exponential survival distribution. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
stratumFraction	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
lambda1	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
lambda2	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.

gamma1	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
gamma2	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
accrualDuration	Duration of the enrollment period.
followupTime	Follow-up time for the last enrolled subject.
fixedFollowup	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.

Value

A vector of calendar times expected to yield the target number of events.

Examples

```
# Piecewise accrual, piecewise exponential survivals, and 5% dropout by
# the end of 1 year.

caltime(nevents = c(24, 80), allocationRatioPlanned = 1,
        accrualTime = seq(0, 9),
        accrualIntensity = c(26/9*seq(1, 9), 26),
        piecewiseSurvivalTime = c(0, 6),
        stratumFraction = c(0.2, 0.8),
        lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
        lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
        gamma1 = -log(1-0.05)/12,
        gamma2 = -log(1-0.05)/12,
        accrualDuration = 22,
        followupTime = 18, fixedFollowup = FALSE)
```

---

errorSpent *Error spending functions*

Description

Obtains the error spent at the given information fractions for the specified error spending function.

Usage

```
errorSpent(t, error, sf = "sfOF", sfpar = NA)
```

Arguments

t	A vector of information fractions.
error	Total error to spend.
sf	Spending function. One of the following: "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, and "sfHSD" for Hwang, Shi & DeCani spending function. Defaults to "sfOF".
sfpar	Parameter for the spending function. Corresponds to rho for "sfKD" and gamma for "sfHSD".

Value

A vector of errors spent up to the interim look.

Examples

```
errorSpent(t = 0.5, error = 0.025, sf = "sfOF")
errorSpent(t = c(0.5, 0.75, 1), error = 0.025, sf = "sfHSD", sfpar = -4)
```

---

exitprob *Stagewise exit probabilities*

Description

Obtains the stagewise exit probabilities for both efficacy and futility stopping.

Usage

```
exitprob(b, a = NA, theta = 0, I = NA)
```

Arguments

b	Upper boundaries on the z-test statistic scale.
a	Lower boundaries on the z-test statistic scale. Defaults to `c(rep(-6.0, kMax-1), b[kMax])` if left unspecified, where `kMax = length(b)`.
theta	Stagewise parameter of interest, e.g., `-U/V` for weighted log-rank test, where `U` is the mean and `V` is the variance of the weighted log-rank test score statistic at each stage. For proportional hazards and conventional log-rank test, use the scalar input, `theta = -log(HR)`. Defaults to 0 corresponding to the null hypothesis.
I	Stagewise cumulative information, e.g., `V`, the variance of the weighted log-rank test score statistic at each stage. For conventional log-rank test, information can be approximated by `phi*(1-phi)*D`, where `phi` is the probability of being allocated to the active arm, and `D` is the total number of events at each stage. Defaults to `seq(1, kMax)` if left unspecified.

Value

A list of stagewise exit probabilities: one vector for efficacy stopping probabilities, and the other vector for futility stopping probabilities.

Examples

```
exitprob(b = c(3.471, 2.454, 2.004), theta = -log(0.6),
         I = c(50, 100, 150)/4)

exitprob(b = c(2.963, 2.359, 2.014),
         a = c(-0.264, 0.599, 2.014),
         theta = c(0.141, 0.204, 0.289),
         I = c(81, 121, 160))
```

`fadjpbon` *Adjusted p-values for Bonferroni-based graphical approaches*

Description

Obtains the adjusted p-values for graphical approaches using weighted Bonferroni tests.

Usage

```
fadjpbon(w, G, p)
```

Arguments

`w`	The vector of initial weights for elementary hypotheses.
`G`	The initial transition matrix.
`p`	The raw p-values for elementary hypotheses.

Value

A matrix of adjusted p-values.

References

Frank Bretz, Willi Maurer, Werner Brannath and Martin Posch. A graphical approach to sequentially rejective multiple test procedures. Statistics in Medicine. 2009; 28:586-604.

Examples

```
pvalues <- matrix(c(0.01,0.005,0.015,0.022, 0.02,0.015,0.010,0.023),
                  nrow=2, ncol=4, byrow=TRUE)
w <- c(0.5,0.5,0,0)
g <- matrix(c(0,0,1,0,0,0,0,1,0,1,0,0,1,0,0,0),
            nrow=4, ncol=4, byrow=TRUE)
fadjpbon(w, g, pvalues)
```

`fadjpsim` *Adjusted p-values for Simes-based graphical approaches*

Description

Obtains the adjusted p-values for graphical approaches using weighted Simes tests.

Usage

```
fadjpsim(wgtmat, p, family)
```

Arguments

`wgtmat`	The weight matrix for intersection hypotheses.
`p`	The raw p-values for elementary hypotheses.
`family`	The matrix of family indicators for elementary hypotheses.

Value

A matrix of adjusted p-values.

References

Frank Bretz, Martin Posch, Ekkehard Glimm, Florian Klinglmueller, Willi Maurer, and Kornelius Rohmeyer. Graphical approach for multiple comparison procedures using weighted Bonferroni, Simes, or parameter tests. Biometrical Journal. 2011; 53:894-913.

Kaifeng Lu. Graphical approaches using a Bonferroni mixture of weighted Simes tests. Statistics in Medicine. 2016; 35:4041-4055.

Examples

```
pvalues <- matrix(c(0.01,0.005,0.015,0.022, 0.02,0.015,0.010,0.023),
                  nrow=2, ncol=4, byrow=TRUE)
w <- c(0.5,0.5,0,0)
g <- matrix(c(0,0,1,0,0,0,0,1,0,1,0,0,1,0,0,0),
            nrow=4, ncol=4, byrow=TRUE)
wgtmat = fwgtmat(w,g)
```

```
family = matrix(c(1,1,0,0,0,0,1,1), nrow=2, ncol=4, byrow=TRUE)
fadjpsim(wgtmat, pvalues, family)
```

`fmodmix` *Adjusted p-values for modified mixture gatekeeping procedures*

Description

Obtains the adjusted p-values for the modified gatekeeping procedures for multiplicity problems involving serial and parallel logical restrictions.

Usage

```
fmodmix(p, family, serial, parallel, gamma, test = "hommel", exhaust = 1)
```

Arguments

`p`	The raw p-values for elementary hypotheses.
`family`	The matrix of family indicators for the hypotheses.
`serial`	The matrix of serial rejection set for the hypotheses.
`parallel`	The matrix of parallel rejection set for the hypotheses.
`gamma`	The truncation parameters for each family.
`test`	The component multiple testing procedure. It is either "Holm" or "Hochberg", and it defaults to "Hochberg".
`exhaust`	Whether to use alpha-exhausting component testing procedure for the last family with active hypotheses. It defaults to TRUE.

Value

A matrix of adjusted p-values.

References

Alex Dmitrienko, George Kordzakhia, and Thomas Brechenmacher. Mixture-based gatekeeping procedures for multiplicity problems with multiple sequences of hypotheses. Journal of Biopharmaceutical Statistics. 2016; 26(4):758–780.

George Kordzakhia, Thomas Brechenmacher, Eiji Ishida, Alex Dmitrienko, Winston Wenxiang Zheng, and David Fuyuan Li. An enhanced mixture method for constructing gatekeeping procedures in clinical trials. Journal of Biopharmaceutical Statistics. 2018; 28(1):113–128.

Examples

```
p = c(0.0194, 0.0068, 0.0271, 0.0088, 0.0370, 0.0018, 0.0814, 0.0066)
family = matrix(c(1, 1, 0, 0, 0, 0, 0, 0,
                  0, 0, 1, 1, 0, 0, 0, 0,
                  0, 0, 0, 0, 1, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 1),
                nrow=4, byrow=TRUE)

serial = matrix(c(0, 0, 0, 0, 0, 0, 0, 0,
                  0, 0, 0, 0, 0, 0, 0, 0,
                  1, 0, 0, 0, 0, 0, 0, 0,
                  0, 1, 0, 0, 0, 0, 0, 0,
                  0, 0, 1, 0, 0, 0, 0, 0,
                  0, 0, 0, 1, 0, 0, 0, 0,
                  0, 0, 0, 0, 1, 0, 0, 0,
                  0, 0, 0, 0, 0, 1, 0, 0),
                nrow=8, byrow=TRUE)

parallel = matrix(0, 8, 8)
gamma = c(0.6, 0.6, 0.6, 1)
fmodmix(p, family, serial, parallel, gamma, "hommel", 1)
```

`fseqbon` *Group sequential trials using Bonferroni-based graphical approaches*

Description

Obtains the test results for group sequential trials using graphical approaches based on weighted Bonferroni tests.

Usage

```
fseqbon(
  w,
  G,
  alpha = 0.025,
  kMax,
  typeAlphaSpending = NULL,
  parameterAlphaSpending = NULL,
  incidenceMatrix = NULL,
  maxInformation = NULL,
  p,
  information,
  spendingTime = NULL
)
```

Arguments

w	The vector of initial weights for elementary hypotheses.
G	The initial transition matrix.
alpha	The significance level. Defaults to 0.025.
kMax	The maximum number of stages.
typeAlphaSpending	The vector of alpha spending functions. Each element is one of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, and "none" for no early efficacy stopping. Defaults to "sfOF" if not provided.
parameterAlphaSpending	The vector of parameter values for the alpha spending functions. Each element corresponds to the value of Delta for "WT", rho for "sfKD", or gamma for "sfHSD". Defaults to missing if not provided.
incidenceMatrix	The incidence matrix indicating whether the specific hypothesis will be tested at the given look. The number of columns of incidenceMatrix must be equal to the maximum number of study looks (kMax). If not provided, defaults to testing each hypothesis at all study looks.
maxInformation	The vector of target maximum information for each hypothesis. Defaults to a vector of 1s if not provided.
p	The matrix of raw p-values for each hypothesis by study look.
information	The matrix of observed information for each hypothesis by study look.
spendingTime	The spending time for alpha spending by study look. If not provided, it is the same as informationRates calculated from information and maxInformation.

Value

A vector to indicate the first look the specific hypothesis is rejected (0 if the hypothesis is not rejected).

References

Willi Maurer and Frank Bretz. Multiple testing in group sequential trials using graphical approaches. Statistics in Biopharmaceutical Research. 2013; 5:311-320.

Examples

```
# Case study from Maurer & Bretz (2013)

fseqbon(
  w = c(0.5, 0.5, 0, 0),
  G = matrix(c(0, 0.5, 0.5, 0,  0.5, 0, 0, 0.5,
```

```
              0, 1, 0, 0,  1, 0, 0, 0),
            nrow=4, ncol=4, byrow=TRUE),
  alpha = 0.025,
  kMax = 3,
  typeAlphaSpending = rep("sfOF", 4),
  maxInformation = rep(1, 4),
  p = matrix(c(0.0062, 0.017, 0.009, 0.13,
               0.0002, 0.0035, 0.002, 0.06),
             nrow=4, ncol=2),
  information = matrix(c(rep(1/3, 4), rep(2/3, 4)),
                       nrow=4, ncol=2))
```

`fstdmix` *Adjusted p-values for standard mixture gatekeeping procedures*

Description

Obtains the adjusted p-values for the standard gatekeeping procedures for multiplicity problems involving serial and parallel logical restrictions.

Usage

```
fstdmix(p, family, serial, parallel, gamma, test = "hommel", exhaust = 1)
```

Arguments

`p`	The raw p-values for elementary hypotheses.
`family`	The matrix of family indicators for the hypotheses.
`serial`	The matrix of serial rejection set for the hypotheses.
`parallel`	The matrix of parallel rejection set for the hypotheses.
`gamma`	The truncation parameters for each family.
`test`	The component multiple testing procedure. It is either "Holm" or "Hochberg", and it defaults to "Hochberg".
`exhaust`	Whether to use alpha-exhausting component testing procedure for the last family with active hypotheses. It defaults to TRUE.

Value

A matrix of adjusted p-values.

References

Alex Dmitrienko and Ajit C Tamhane. Mixtures of multiple testing procedures for gatekeeping applications in clinical trials. Statistics in Medicine. 2011; 30(13):1473–1488.

Examples

```
p = c(0.0194, 0.0068, 0.0271, 0.0088, 0.0370, 0.0018, 0.0814, 0.0066)
family = matrix(c(1, 1, 0, 0, 0, 0, 0, 0,
                  0, 0, 1, 1, 0, 0, 0, 0,
                  0, 0, 0, 0, 1, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 1),
                nrow=4, byrow=TRUE)

serial = matrix(c(0, 0, 0, 0, 0, 0, 0, 0,
                  0, 0, 0, 0, 0, 0, 0, 0,
                  1, 0, 0, 0, 0, 0, 0, 0,
                  0, 1, 0, 0, 0, 0, 0, 0,
                  0, 0, 1, 0, 0, 0, 0, 0,
                  0, 0, 0, 1, 0, 0, 0, 0,
                  0, 0, 0, 0, 1, 0, 0, 0,
                  0, 0, 0, 0, 0, 1, 0, 0),
                nrow=8, byrow=TRUE)

parallel = matrix(0, 8, 8)
gamma = c(0.6, 0.6, 0.6, 1)
fstdmix(p, family, serial, parallel, gamma, "hommel", 0)
```

---

fstp2seq *Adjusted p-values for stepwise testing procedures for two sequences*

Description

Obtains the adjusted p-values for the stepwise gatekeeping procedures for multiplicity problems involving two sequences of hypotheses.

Usage

```
fstp2seq(p, gamma, test = "hochberg", retest = TRUE)
```

Arguments

p	The raw p-values for elementary hypotheses.
gamma	The truncation parameters for each family.
test	The component multiple testing procedure. It is either "Holm" or "Hochberg", and it defaults to "Hochberg".
retest	Whether to allow retesting. It defaults to TRUE.

Value

A matrix of adjusted p-values.

Examples

```
p = c(0.0194, 0.0068, 0.0271, 0.0088, 0.0370, 0.0018, 0.0814, 0.0066)
gamma = c(0.6, 0.6, 0.6, 1)
fstp2seq(p, gamma, test="hochberg", retest=1)
```

---

fwgtmat *Weight matrix for all intersection hypotheses*

Description

Obtains the weight matrix for all intersection hypotheses.

Usage

```
fwgtmat(w, G)
```

Arguments

w	The vector of weights for elementary hypotheses.
G	The transition matrix.

Value

The weight matrix starting with the global null hypothesis.

Examples

```
w = c(0.5,0.5,0,0)
g = matrix(c(0,0,1,0,0,0,0,1,0,1,0,0,1,0,0,0), nrow=4, ncol=4, byrow=TRUE)
(wgtmat = fwgtmat(w,g))
```

---

getAccrualDuration *Accrual duration to enroll target number of subjects*

Description

Obtains the accrual duration to enroll the target number of subjects.

Usage

```
getAccrualDuration(
  nsubjects = NA_real_,
  accrualTime = 0L,
  accrualIntensity = NA_real_
)
```

Arguments

nsubjects	The vector of target number of subjects.
accrualTime	Accrual time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
accrualIntensity	A vector of accrual intensities. One for each accrual time interval.

Value

The vector of accrual duration.

Examples

```
getAccrualDuration(nsubjects = c(20, 150), accrualTime = c(0, 3),
                   accrualIntensity = c(10, 20))
```

---

getBound *Get efficacy boundaries for group sequential design*

Description

Obtains the efficacy stopping boundaries for a group sequential design.

Usage

```
getBound(
  k = NA_integer_,
  informationRates = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

k	Look number for the current analysis.
informationRates	Information rates up to the current look. Must be increasing and less than or equal to 1.
alpha	The significance level. Defaults to 0.025.
typeAlphaSpending	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
parameterAlphaSpending	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
userAlphaSpending	The user defined alpha spending. Cumulative alpha spent up to each stage.
spendingTime	A vector of length k for the error spending time at each analysis. Must be increasing and less than or equal to 1. Defaults to missing, in which case, it is the same as informationRates.

Value

A numeric vector of critical values up to the current look.

Examples

```
getBound(k = 2, informationRates = c(0.5,1),
         alpha = 0.025, typeAlphaSpending = "sfOF")
```

---

getDesign *Get group sequential design*

Description

Obtains the drift parameter and stopping boundaries for a generic group sequential design assuming a constant treatment effect, or obtains the power given the drift parameter and stopping boundaries.

Usage

```
getDesign(
  beta = 0.2,
  drift = NA_real_,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	Type II error. Defaults to 0.2.
`drift`	Drift parameter, i.e., `(theta-theta0)*sqrt(Imax)`. If `drift` is provided, then the input `beta` will be ignored and power will be calculated.
`kMax`	The maximum number of stages.
`informationRates`	The information rates in terms of number of events. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".

userAlphaSpending
: The user defined alpha spending. Cumulative alpha spent up to each stage.

futilityBounds
: Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., kMax-1. Defaults to rep(-6, kMax-1) if left unspecified.

typeBetaSpending
: The type of beta spending. One of the following: "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early futility stopping. Defaults to "none".

parameterBetaSpending
: The parameter value for the beta spending. Corresponds to rho for "sfKD", and gamma for "sfHSD".

userBetaSpending
: The user defined beta spending. Cumulative beta spent up to each stage.

spendingTime
: A vector of length kMax for the error spending time at each analysis. Defaults to missing, in which case, it is the same as informationRates.

Value

A list of S3 class design with three components:

* overallResults containing the overall rejection probability, overall significance level, number of stages, drift parameter, and inflation factor (relative to fixed design).

* byStageResults containing information rates, efficacy and futility boundaries on the Z-scale, probability for efficacy and futility stopping at the stage, cumulative probability for efficacy and futility stopping by the stage, cumulative alpha spent, efficacy and futility boundaries on the p-value scale, and whether efficacy and futility stopping are allowed by stage.

* settings containing input parameters such as alpha and beta spending function and parameter values, spendingTime, and calculation target.

Examples

```
# Example 1: obtain the drift parameter given power
getDesign(beta = 0.2,
          kMax = 2,
          informationRates = c(0.5,1),
          alpha = 0.025,
          typeAlphaSpending = "sfOF",
          typeBetaSpending = "sfP")


# Example 2: obtain power given the drift parameter
getDesign(drift = 3.026,
          kMax = 3,
          informationRates = c(0.5, 0.75, 1),
          alpha = 0.025,
          typeAlphaSpending = "sfOF",
```

```
            typeBetaSpending = "sfP")
```

`getDurationFromNevents` *Range of accrual duration for target number of events*

Description

Obtains a range of accrual duration to reach the target number of events.

Usage

```
getDurationFromNevents(
  nevents = NA_real_,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  followupTime = 18,
  fixedFollowup = 0L,
  npoints = 23L,
  interval = as.numeric(c(0.001, 240))
)
```

Arguments

`nevents`	The target number of events.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	Accrual time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the time intervals for the piecewise exponential survival distribution. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.

lambda1	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
lambda2	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
gamma1	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
gamma2	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
followupTime	Follow-up time for the last enrolled subject.
fixedFollowup	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
npoints	The number of accrual duration time points. Defaults to 23.
interval	The interval to search for the solution of accrualDuration. Defaults to `c(0.001, 240)`.

Value

A data frame of enrollment duration, sample size, and study duration to yield the target number of events.

Examples

```
# Piecewise accrual, piecewise exponential survivals, and 5% dropout by
# the end of 1 year.

getDurationFromNevents(
  nevents = 80, allocationRatioPlanned = 1,
  accrualTime = seq(0, 8),
  accrualIntensity = 26/9*seq(1, 9),
  piecewiseSurvivalTime = c(0, 6),
  stratumFraction = c(0.2, 0.8),
  lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
  lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
  gamma1 = -log(1-0.05)/12,
  gamma2 = -log(1-0.05)/12,
  fixedFollowup = FALSE)
```

---

getNeventsFromHazardRatio

Get the required number of events from hazard ratios

Description

Obtains the required number of events given the hazard ratios under the null and alternative hypotheses for a group sequential design.

Usage

```
getNeventsFromHazardRatio(
  beta = 0.2,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_,
  hazardRatioH0 = 1,
  hazardRatio = 0.5,
  allocationRatioPlanned = 1,
  rounding = 1L
)
```

Arguments

`beta`	Type II error. Defaults to 0.2.
`kMax`	The maximum number of stages.
`informationRates`	The information rates in terms of number of events. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".

parameterAlphaSpending
: The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".

userAlphaSpending
: The user defined alpha spending. Cumulative alpha spent up to each stage.

futilityBounds
: Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., kMax-1. Defaults to rep(-6, kMax-1) if left unspecified.

typeBetaSpending
: The type of beta spending. One of the following: "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early futility stopping. Defaults to "none".

parameterBetaSpending
: The parameter value for the beta spending. Corresponds to rho for "sfKD", and gamma for "sfHSD".

userBetaSpending
: The user defined beta spending. Cumulative beta spent up to each stage.

spendingTime
: A vector of length kMax for the error spending time at each analysis. Defaults to missing, in which case, it is the same as informationRates.

hazardRatioH0
: Hazard ratio under the null hypothesis for the active treatment versus control. Defaults to 1 for superiority test.

hazardRatio
: Hazard ratio under the alternative hypothesis for the active treatment versus control. Defaults to 0.5.

allocationRatioPlanned
: Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.

rounding
: Whether to round up the number of events. Defaults to 1 for rounding.

Value

The required number of events.

Examples

```
getNeventsFromHazardRatio(
  beta = 0.2, kMax = 2,
  informationRates = c(0.5,1),
  alpha = 0.025, typeAlphaSpending = "sfOF",
  typeBetaSpending = "sfP",
  hazardRatio = 0.673)
```

kmest *Stratified difference in milestone survival*

Description

Obtains the stratified Kaplan-Meier estimate of milestone survival probabilities and difference in milestone survival at given calendar times and milestone time.

Usage

```
kmest(
  time = NA_real_,
  milestone = NA_real_,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  numSubintervals = 300L
)
```

Arguments

`time`	A vector of calendar times at which to calculate the milestone survival.
`milestone`	The milestone time at which to calculate the Kaplan-Meier estimate of survival probability.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	Accrual time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the time intervals for the piecewise exponential survival distribution. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.

lambda1 A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.

lambda2 A vector of hazard rates for the event in each analysis time interval by stratum for the control group.

gamma1 The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.

gamma2 The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.

accrualDuration Duration of the enrollment period.

followupTime Follow-up time for the last enrolled subject.

fixedFollowup Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.

numSubintervals Number of sub-intervals to approximate the mean and variance of the weighted log-rank test score statistic. Defaults to 300. Specify a larger number for better approximation.

Value

A data frame of the number of subjects enrolled, stratified estimate of milestone survival for each treatment group, difference in milestone survival, the associated variances, and the Z test statistic at the specified calendar times.

Examples

```
# Piecewise accrual, piecewise exponential survivals, and 5% dropout by
# the end of 1 year.

kmest(time = c(22, 40),
      milestone = 18,
      allocationRatioPlanned = 1,
      accrualTime = seq(0, 9),
      accrualIntensity = c(26/9*seq(1, 9), 26),
      piecewiseSurvivalTime = c(0, 6),
      stratumFraction = c(0.2, 0.8),
      lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
      lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
      gamma1 = -log(1-0.05)/12,
      gamma2 = -log(1-0.05)/12,
      accrualDuration = 22,
      followupTime = 18, fixedFollowup = FALSE)
```

`lrpower` *Log-rank test power*

Description

Estimates the power, stopping probabilities, and expected sample size in a two-group survival design.

Usage

```
lrpower(
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  hazardRatioH0 = 1,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = 20L,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  lambda1 = 0.0309,
  lambda2 = 0.0533,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = 11.6,
  followupTime = 18,
  fixedFollowup = 0L,
  rho1 = 0,
  rho2 = 0,
  numSubintervals = 300L,
  estimateHazardRatio = 1L,
  typeOfComputation = "direct",
  spendingTime = NA_real_,
  studyDuration = NA_real_
)
```

Arguments

`kMax`	The maximum number of stages.

`informationRates`
: The information rates in terms of number of events. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.

`efficacyStopping`
: Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.

`futilityStopping`
: Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.

`criticalValues`
: Upper boundaries on the z-test statistic scale for stopping for efficacy.

`alpha`
: The significance level. Defaults to 0.025.

`typeAlphaSpending`
: The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".

`parameterAlphaSpending`
: The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".

`userAlphaSpending`
: The user defined alpha spending. Cumulative alpha spent up to each stage.

`futilityBounds`
: Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified.

`typeBetaSpending`
: The type of beta spending. One of the following: "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, and "none" for no early futility stopping. Defaults to "none".

`parameterBetaSpending`
: The parameter value for the beta spending. Corresponds to rho for "sfKD", and gamma for "sfHSD".

`hazardRatioH0`
: Hazard ratio under the null hypothesis for the active treatment versus control. Defaults to 1 for superiority test.

`allocationRatioPlanned`
: Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.

`accrualTime`
: Accrual time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).

`accrualIntensity`
: A vector of accrual intensities. One for each accrual time interval.

`piecewiseSurvivalTime`
: A vector that specifies the time intervals for the piecewise exponential survival distribution. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.

stratumFraction
: A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.

lambda1
: A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.

lambda2
: A vector of hazard rates for the event in each analysis time interval by stratum for the control group.

gamma1
: The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.

gamma2
: The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.

accrualDuration
: Duration of the enrollment period.

followupTime
: Follow-up time for the last enrolled subject.

fixedFollowup
: Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.

rho1
: The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

rho2
: The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

numSubintervals
: Number of sub-intervals to approximate the mean and variance of the weighted log-rank test score statistic. Defaults to 300. Specify a larger number for better approximation.

estimateHazardRatio
: Whether to estimate the hazard ratio from weighted Cox regression model and report the stopping boundaries on the hazard ratio scale.

typeOfComputation
: Whether to use the direct approximation method or the Schoenfeld method. Defaults to "direct". Can use "Schoenfeld" under proportional hazards and conventional log-rank test.

spendingTime
: A vector of length kMax for the error spending time at each analysis. Defaults to missing, in which case, it is the same as informationRates.

studyDuration
: Study duration for fixed follow-up design. Defaults to missing, which is to be replaced with the sum of accrualDuration and followupTime. If provided, the value is allowed to be less than the sum of accrualDuration and followupTime.

Value

A list of S3 class lrpower with 3 components:

* overallResults containing the overall rejection probability, overall significance level, maximum and expected number of events, maximum and expected number of dropouts, total and expected number of subjects, maximum and expected study duration, along with input parameters including

accrual duration, follow-up duration, whether a fixed follow-up is used, parameters for the FH weights, allocation ratio, number of stages, and hazard ratio under H0.

* `byStageResults` containing information rates, efficacy and futility boundaries on the Z-scale, probability for efficacy and futility stopping at the stage, cumulative probability for efficacy and futility stopping by the stage, cumulative alpha spent, expected number of events, number of dropouts, number of subjects, and expected study time, efficacy and futility boundaries on the HR scale and on the p-value scale, information for weighted log-rank test, hazard ratio from weighted Cox regression, and whether efficacy and futility stopping are allowed by stage.

* `settings` containing input parameters such as alpha and beta spending function and parameter values, accrual time, accrual intensity, piecewise survival time, stratum fraction, and hazard rates for survival and dropout by group.

Examples

```
# Piecewise accrual, piecewise exponential survival, and 5% dropout by
# the end of 1 year.

lrpower(kMax = 2, informationRates = c(0.8, 1),
        alpha = 0.025, typeAlphaSpending = "sfOF",
        allocationRatioPlanned = 1, accrualTime = seq(0, 9),
        accrualIntensity = c(26/9*seq(1, 9), 26),
        piecewiseSurvivalTime = c(0, 6),
        stratumFraction = c(0.2, 0.8),
        lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
        lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
        gamma1 = -log(1-0.05)/12,
        gamma2 = -log(1-0.05)/12, accrualDuration = 22,
        followupTime = 18, fixedFollowup = FALSE)
```

---

lrsamplesize *Log-rank test sample size*

Description

Obtains the needed accrual duration given power and follow-up time, the needed follow-up time given power and accrual duration, or the needed absolute accrual rates given power, accrual duration, follow-up duration, and relative accrual rates in a two-group survival design.

Usage

```
lrsamplesize(
  beta = 0.2,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
```

```
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  hazardRatioH0 = 1,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = 20L,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  lambda1 = 0.0309,
  lambda2 = 0.0533,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = NA_real_,
  followupTime = 18,
  fixedFollowup = 0L,
  rho1 = 0,
  rho2 = 0,
  numSubintervals = 300L,
  estimateHazardRatio = 1L,
  typeOfComputation = "direct",
  interval = as.numeric(c(0.001, 240)),
  spendingTime = NA_real_,
  rounding = 1L
)
```

Arguments

`beta`	Type II error. Defaults to 0.2.
`kMax`	The maximum number of stages.
`informationRates`	The information rates in terms of number of events. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.

typeAlphaSpending
: The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".

parameterAlphaSpending
: The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".

userAlphaSpending
: The user defined alpha spending. Cumulative alpha spent up to each stage.

futilityBounds
: Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified.

typeBetaSpending
: The type of beta spending. One of the following: "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early futility stopping. Defaults to "none".

parameterBetaSpending
: The parameter value for the beta spending. Corresponds to rho for "sfKD", and gamma for "sfHSD".

userBetaSpending
: The user defined beta spending. Cumulative beta spent up to each stage.

hazardRatioH0
: Hazard ratio under the null hypothesis for the active treatment versus control. Defaults to 1 for superiority test.

allocationRatioPlanned
: Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.

accrualTime
: Accrual time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).

accrualIntensity
: A vector of accrual intensities. One for each accrual time interval.

piecewiseSurvivalTime
: A vector that specifies the time intervals for the piecewise exponential survival distribution. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.

stratumFraction
: A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.

lambda1
: A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.

lambda2
: A vector of hazard rates for the event in each analysis time interval by stratum for the control group.

gamma1 The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.

gamma2 The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.

accrualDuration
Duration of the enrollment period.

followupTime Follow-up time for the last enrolled subject.

fixedFollowup Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.

rho1 The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

rho2 The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

numSubintervals
Number of sub-intervals to approximate the mean and variance of the weighted log-rank test score statistic. Defaults to 300. Specify a larger number for better approximation.

estimateHazardRatio
Whether to estimate the hazard ratio from weighted Cox regression model and report the stopping boundaries on the hazard ratio scale.

typeOfComputation
Whether to use the direct approximation method or the Schoenfeld method. Defaults to "direct". Can use "Schoenfeld" under proportional hazards and conventional log-rank test.

interval The interval to search for the solution of accrualDuration, followupDuration, or the proportionality constant of accrualIntensity. Defaults to `c(0.001, 240)`. Adjustment may be needed for non-monotone relationship with study power.

spendingTime A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.

rounding Whether to round up sample size and events. Defaults to 1 for sample size rounding.

Value

A list of S3 class `lrpower`.

Examples

```
# Piecewise accrual, piecewise exponential survival, and 5% dropout by
# the end of 1 year.

# Example 1: Obtains accrual duration given power and follow-up duration

lrsamplesize(beta = 0.2, kMax = 2,
             informationRates = c(0.8, 1),
```

```
             alpha = 0.025, typeAlphaSpending = "sfOF",
             accrualTime = seq(0, 9),
             accrualIntensity = c(26/9*seq(1, 9), 26),
             piecewiseSurvivalTime = c(0, 6),
             stratumFraction = c(0.2, 0.8),
             lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
             lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
             gamma1 = -log(1-0.05)/12,
             gamma2 = -log(1-0.05)/12,
             accrualDuration = NA,
             followupTime = 18, fixedFollowup = FALSE)


# Example 2: Obtains follow-up duration given power and accrual duration

lrsamplesize(beta = 0.2, kMax = 2,
             informationRates = c(0.8, 1),
             alpha = 0.025, typeAlphaSpending = "sfOF",
             accrualTime = seq(0, 9),
             accrualIntensity = c(26/9*seq(1, 9), 26),
             piecewiseSurvivalTime = c(0, 6),
             stratumFraction = c(0.2, 0.8),
             lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
             lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
             gamma1 = -log(1-0.05)/12,
             gamma2 = -log(1-0.05)/12,
             accrualDuration = 22,
             followupTime = NA, fixedFollowup = FALSE)


# Example 3: Obtains absolute accrual intensity given power,
# accrual duration, follow-up duration, and relative accrual intensity

lrsamplesize(beta = 0.2, kMax = 2,
             informationRates = c(0.8, 1),
             alpha = 0.025, typeAlphaSpending = "sfOF",
             accrualTime = seq(0, 9),
             accrualIntensity = c(26/9*seq(1, 9), 26),
             piecewiseSurvivalTime = c(0, 6),
             stratumFraction = c(0.2, 0.8),
             lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
             lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
             gamma1 = -log(1-0.05)/12,
             gamma2 = -log(1-0.05)/12,
             accrualDuration = 22,
             followupTime = 18, fixedFollowup = FALSE)


# Example 4: Non-inferiority trial with fixed follow-up and
# superiority alternative

lrsamplesize(beta = 0.1,
             kMax = 3,
```

```
alpha = 0.025,
typeAlphaSpending = "sfOF",
hazardRatioH0 = 1.1,
accrualTime = c(0, 6),
accrualIntensity = c(1000, 1500),
lambda1 = log(2)/48*0.95,
lambda2 = log(2)/48,
gamma1 = -log(1-0.08)/12,
gamma2 = -log(1-0.08)/12,
accrualDuration = NA,
followupTime = 18,
fixedFollowup = 1,
typeOfComputation = "Schoenfeld")
```

`lrsim` *Log-rank test simulation*

Description

Performs simulation for two-arm group sequential trials based on weighted log-rank test.

Usage

```
lrsim(
  kMax = NA_integer_,
  informationTime = NA_real_,
  criticalValues = NA_real_,
  futilityBounds = NA_real_,
  hazardRatioH0 = 1,
  allocation1 = 1L,
  allocation2 = 1L,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  rho1 = 0,
  rho2 = 0,
  plannedEvents = NA_integer_,
```

```
  plannedTime = NA_real_,
  maxNumberOfIterations = 1000L,
  maxNumberOfRawDatasetsPerStage = 0L,
  seed = NA_integer_
)
```

Arguments

kMax	The maximum number of stages.
informationTime	Information time in terms of variance of weighted log-rank test score statistic under the null hypothesis. Same as informationRates in terms of number of events for the conventional log-rank test. Use caltime and lrstat to derive the information time for weighted log-rank tests. Fixed prior to the trial. If left unspecified, it defaults to plannedEvents / plannedEvents[kMax] when plannedEvents is provided and to plannedTime / plannedTime[kMax] otherwise.
criticalValues	Upper boundaries on the z-test statistic scale for stopping for efficacy.
futilityBounds	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., kMax-1. Defaults to rep(-6, kMax-1) if left unspecified.
hazardRatioH0	Hazard ratio under the null hypothesis for the active treatment versus control. Defaults to 1 for superiority test.
allocation1	Number of subjects in the active treatment group in a randomization block. Defaults to 1 for equal randomization.
allocation2	Number of subjects in the control group in a randomization block. Defaults to 1 for equal randomization.
accrualTime	Accrual time intervals. Must start with 0, e.g., c(0, 3) breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
accrualIntensity	A vector of accrual intensities. One for each accrual time interval.
piecewiseSurvivalTime	A vector that specifies the time intervals for the piecewise exponential survival distribution. Must start with 0, e.g., c(0, 6) breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
stratumFraction	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
lambda1	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
lambda2	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
gamma1	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
gamma2	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.

accrualDuration
: Duration of the enrollment period.

followupTime
: Follow-up time for the last enrolled subject.

fixedFollowup
: Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.

rho1
: The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

rho2
: The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

plannedEvents
: The planned cumulative total number of events at each stage.

plannedTime
: The calendar times for the analyses. To use calendar time to plan the analyses, plannedEvents should be missing.

maxNumberOfIterations
: The number of simulation iterations. Defaults to 1000.

maxNumberOfRawDatasetsPerStage
: The number of raw datasets per stage to extract. Defaults to 1.

seed
: The seed to reproduce the simulation results. The computer clock will be used if left unspecified,

Value

A list of S3 class lrsim with 3 components:

* overview is a list containing incremental and cumulative efficacy and futility stopping probabilities by stage, expected number of events, number of dropouts, number of subjects, analysis time by stage, overall rejection probability, overall expected number of events, number of dropouts, number of subjects, study duration, hazard ratio under H0, and whether the analyses are planned based on the number of events or calendar time.

* sumdata is a data frame of summary data by stage for each iteration, containing at which stage the trial stops, whether the target number of events is achieved, the analysis time, number of accrued subjects overall and by treatment group, number of events overall and by treatment group, number of dropouts overall and by treatment group, numerator and variance of weighted log-rank score statistic, log-rank test Z-statistic, and whether the trial stops for efficacy or futility at the stage.

* rawdata (exists if maxNumberOfRawDatasetsPerStage is a positive integer) is a data frame for subject-level data for selected replications, containing the subject number, arrival time, stratum, treatment group, survival time, dropout time, observation time when the trial stops, time under observation, and event and dropout indicators.

Examples

```
# Example 1: analyses based on number of events

sim1 = lrsim(kMax = 2, informationTime = c(0.5, 1),
             criticalValues = c(2.797, 1.977),
             accrualIntensity = 11,
             lambda1 = 0.018, lambda2 = 0.030,
             accrualDuration = 12,
             plannedEvents = c(60, 120),
```

```
              maxNumberOfIterations = 1000,
              maxNumberOfRawDatasetsPerStage = 1,
              seed = 314159)

# summary statistics
sim1

# summary for each simulated data set
head(sim1$sumdata)

# raw data for selected replication
head(sim1$rawdata)


# Example 2: analyses based on calendar time have similar power

sim2 = lrsim(kMax = 2, informationTime = c(0.5, 1),
              criticalValues = c(2.797, 1.977),
              accrualIntensity = 11,
              lambda1 = 0.018, lambda2 = 0.030,
              accrualDuration = 12,
              plannedTime = c(31.9, 113.2),
              maxNumberOfIterations = 1000,
              maxNumberOfRawDatasetsPerStage = 1,
              seed = 314159)

# summary statistics
sim2

# summary for each simulated data set
head(sim2$sumdata)
```

`lrsim2e` *Log-rank test simulation for two endpoints and two arms*

Description

Performs simulation for two-endpoint two-arm group sequential trials based on weighted log-rank test. The first `kMaxe1` looks are driven by the total number of PFS events in two arms combined, and the subsequent looks are driven by the total number of OS events in two arms combined. Alternatively, the analyses can be planned to occur at specified calendar times.

Usage

```
lrsim2e(
  kMax = NA_integer_,
  kMaxe1 = NA_integer_,
  hazardRatioH0e1 = 1,
  hazardRatioH0e2 = 1,
```

```
  allocation1 = 1L,
  allocation2 = 1L,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  rho = 0,
  lambda1e1 = NA_real_,
  lambda2e1 = NA_real_,
  lambda1e2 = NA_real_,
  lambda2e2 = NA_real_,
  gamma1e1 = 0L,
  gamma2e1 = 0L,
  gamma1e2 = 0L,
  gamma2e2 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  rho1 = 0,
  rho2 = 0,
  plannedEvents = NA_integer_,
  plannedTime = NA_real_,
  maxNumberOfIterations = 1000L,
  maxNumberOfRawDatasetsPerStage = 0L,
  seed = NA_integer_
)
```

Arguments

kMax	The maximum number of stages.
kMaxe1	Number of stages with timing determined by PFS events. Ranges from 0 (none) to kMax.
hazardRatioH0e1	Hazard ratio under the null hypothesis for the active treatment vs control for endpoint 1 (PFS). Defaults to 1 for superiority test.
hazardRatioH0e2	Hazard ratio under the null hypothesis for the active treatment vs control for endpoint 2 (OS). Defaults to 1 for superiority test.
allocation1	Number of subjects in the treatment group in a randomization block. Defaults to 1 for equal randomization.
allocation2	Number of subjects in the control group in a randomization block. Defaults to 1 for equal randomization.
accrualTime	Accrual time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
accrualIntensity	A vector of accrual intensities. One for each accrual time interval.

piecewiseSurvivalTime
: A vector that specifies the time intervals for the piecewise exponential survival distribution. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.

stratumFraction
: A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.

rho
: The correlation coefficient for the standard bivariate normal random variables used to generate time to disease progression and time to death using the inverse CDF method.

lambda1e1
: A vector of hazard rates for the event in each analysis time interval by stratum for the treatment group and endpoint 1 (PFS).

lambda2e1
: A vector of hazard rates for the event in each analysis time interval by stratum for the control group and endpoint 1 (PFS).

lambda1e2
: A vector of hazard rates for the event in each analysis time interval by stratum for the treatment group and endpoint 2 (OS).

lambda2e2
: A vector of hazard rates for the event in each analysis time interval by stratum for the control group and endpoint 2 (OS).

gamma1e1
: The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the treatment group and endpoint 1 (PFS).

gamma2e1
: The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group and endpoint 1 (PFS).

gamma1e2
: The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the treatment group and endpoint 2 (OS).

gamma2e2
: The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group and endpoint 2 (OS).

accrualDuration
: Duration of the enrollment period.

followupTime
: Follow-up time for the last enrolled subject.

fixedFollowup
: Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.

rho1
: The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

rho2
: The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

plannedEvents
: The planned cumulative total number of PFS events at Look 1 to Look kMaxe1 and the planned cumulative total number of OS events at Look kMaxe1+1 to Look kMax.

plannedTime The calendar times for the analyses. To use calendar time to plan the analyses, plannedEvents should be missing.

maxNumberOfIterations
The number of simulation iterations. Defaults to 1000.

maxNumberOfRawDatasetsPerStage
The number of raw datasets per stage to extract. Defaults to 1.

seed The seed to reproduce the simulation results. The computer clock will be used if left unspecified,

Value

A list with 2 components:

* sumdata is a data frame of summary data by stage for each iteration, containing the analysis time, number of accrued subjects overall and by treatment group, and number of events overall and by treatment group, number of dropouts overall and by treatment group, and log-rank test statistic by endpoint.

* rawdata (exists if maxNumberOfRawDatasetsPerStage is a positive integer) is a data frame for subject-level data for selected replications, containing the stage number, subject number, arrival time, stratum, treatment group, observation time, and survival time, dropout time, time under observation, and event and dropout indicators for each endpoint.

Examples

```
sim1 = lrsim2e(
  kMax = 3,
  kMaxe1 = 2,
  allocation1 = 2,
  allocation2 = 1,
  accrualTime = c(0, 8),
  accrualIntensity = c(10, 28),
  piecewiseSurvivalTime = 0,
  rho = 0,
  lambda1e1 = log(2)/12*0.60,
  lambda2e1 = log(2)/12,
  lambda1e2 = log(2)/30*0.65,
  lambda2e2 = log(2)/30,
  accrualDuration = 20.143,
  plannedEvents = c(186, 259, 183),
  maxNumberOfIterations = 1000,
  maxNumberOfRawDatasetsPerStage = 1,
  seed = 314159)

head(sim1$sumdata)
head(sim1$rawdata)
```

`lrsim2e3a` *Log-rank test simulation for two endpoints and three arms*

Description

Performs simulation for two-endpoint three-arm group sequential trials based on weighted log-rank test. The first `kMaxe1` looks are driven by the total number of PFS events in Arm A and Arm C combined, and the subsequent looks are driven by the total number of OS events in Arm A and Arm C combined. Alternatively, the analyses can be planned to occur at specified calendar times.

Usage

```
lrsim2e3a(
  kMax = NA_integer_,
  kMaxe1 = NA_integer_,
  hazardRatioH013e1 = 1,
  hazardRatioH023e1 = 1,
  hazardRatioH012e1 = 1,
  hazardRatioH013e2 = 1,
  hazardRatioH023e2 = 1,
  hazardRatioH012e2 = 1,
  allocation1 = 1L,
  allocation2 = 1L,
  allocation3 = 1L,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  rho = 0,
  lambda1e1 = NA_real_,
  lambda2e1 = NA_real_,
  lambda3e1 = NA_real_,
  lambda1e2 = NA_real_,
  lambda2e2 = NA_real_,
  lambda3e2 = NA_real_,
  gamma1e1 = 0L,
  gamma2e1 = 0L,
  gamma3e1 = 0L,
  gamma1e2 = 0L,
  gamma2e2 = 0L,
  gamma3e2 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  rho1 = 0,
  rho2 = 0,
  plannedEvents = NA_integer_,
```

```
  plannedTime = NA_real_,
  maxNumberOfIterations = 1000L,
  maxNumberOfRawDatasetsPerStage = 0L,
  seed = NA_integer_
)
```

Arguments

kMax	The maximum number of stages.
kMaxe1	Number of stages with timing determined by PFS events. Ranges from 0 (none) to kMax.
hazardRatioH013e1	Hazard ratio under the null hypothesis for arm 1 vs arm 3 for endpoint 1 (PFS). Defaults to 1 for superiority test.
hazardRatioH023e1	Hazard ratio under the null hypothesis for arm 2 vs arm 3 for endpoint 1 (PFS). Defaults to 1 for superiority test.
hazardRatioH012e1	Hazard ratio under the null hypothesis for arm 1 vs arm 2 for endpoint 1 (PFS). Defaults to 1 for superiority test.
hazardRatioH013e2	Hazard ratio under the null hypothesis for arm 1 vs arm 3 for endpoint 2 (OS). Defaults to 1 for superiority test.
hazardRatioH023e2	Hazard ratio under the null hypothesis for arm 2 vs arm 3 for endpoint 2 (OS). Defaults to 1 for superiority test.
hazardRatioH012e2	Hazard ratio under the null hypothesis for arm 1 vs arm 2 for endpoint 2 (OS). Defaults to 1 for superiority test.
allocation1	Number of subjects in Arm A in a randomization block. Defaults to 1 for equal randomization.
allocation2	Number of subjects in Arm B in a randomization block. Defaults to 1 for equal randomization.
allocation3	Number of subjects in Arm C in a randomization block. Defaults to 1 for equal randomization.
accrualTime	Accrual time intervals. Must start with 0, e.g., c(0, 3) breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
accrualIntensity	A vector of accrual intensities. One for each accrual time interval.
piecewiseSurvivalTime	A vector that specifies the time intervals for the piecewise exponential survival distribution. Must start with 0, e.g., c(0, 6) breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
stratumFraction	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.

rho The correlation coefficient for the standard bivariate normal random variables used to generate time to disease progression and time to death using the inverse CDF method.

lambda1e1 A vector of hazard rates for the event in each analysis time interval by stratum for arm 1 and endpoint 1 (PFS).

lambda2e1 A vector of hazard rates for the event in each analysis time interval by stratum for arm 2 and endpoint 1 (PFS).

lambda3e1 A vector of hazard rates for the event in each analysis time interval by stratum for arm 3 and endpoint 1 (PFS).

lambda1e2 A vector of hazard rates for the event in each analysis time interval by stratum for arm 1 and endpoint 2 (OS).

lambda2e2 A vector of hazard rates for the event in each analysis time interval by stratum for arm 2 and endpoint 2 (OS).

lambda3e2 A vector of hazard rates for the event in each analysis time interval by stratum for arm 3 and endpoint 2 (OS).

gamma1e1 The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 1 and endpoint 1 (PFS).

gamma2e1 The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 2 and endpoint 1 (PFS).

gamma3e1 The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 3 and endpoint 1 (PFS).

gamma1e2 The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 1 and endpoint 2 (OS).

gamma2e2 The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 2 and endpoint 2 (OS).

gamma3e2 The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 3 and endpoint 2 (OS).

accrualDuration

Duration of the enrollment period.

followupTime Follow-up time for the last enrolled subject.

fixedFollowup Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.

rho1 The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

rho2 The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

plannedEvents The planned cumulative total number of PFS events at Look 1 to Look kMaxe1 for Arms A and C combined and the planned cumulative total number of OS events at Look kMaxe1+1 to Look kMax for Arms A and C combined.

plannedTime The calendar times for the analyses. To use calendar time to plan the analyses, plannedEvents should be missing.

maxNumberOfIterations
The number of simulation iterations. Defaults to 1000.

maxNumberOfRawDatasetsPerStage
The number of raw datasets per stage to extract. Defaults to 1.

seed The seed to reproduce the simulation results. The computer clock will be used if left unspecified,

Value

A list with 2 components:

* sumdata is a data frame of summary data by stage for each iteration, containing the analysis time, number of accrued subjects overall and by treatment group, number of events overall and by treatment group, number of dropouts overall and by treatment group, and log-rank test statistic for each comparison by endpoint.

* rawdata (exists if maxNumberOfRawDatasetsPerStage is a positive integer) is a data frame for subject-level data for selected replications, containing the stage number, subject number, arrival time, stratum, treatment group, observation time, survival time, dropout time, time under observation, and event and dropout indicators for each endpoint.

Examples

```
sim1 = lrsim2e3a(
  kMax = 3,
  kMaxe1 = 2,
  allocation1 = 2,
  allocation2 = 2,
  allocation3 = 1,
  accrualTime = c(0, 8),
  accrualIntensity = c(10, 28),
  piecewiseSurvivalTime = 0,
  rho = 0,
  lambda1e1 = log(2)/12*0.60,
  lambda2e1 = log(2)/12*0.70,
  lambda3e1 = log(2)/12,
  lambda1e2 = log(2)/30*0.65,
  lambda2e2 = log(2)/30*0.75,
  lambda3e2 = log(2)/30,
  accrualDuration = 30.143,
  plannedEvents = c(186, 259, 183),
  maxNumberOfIterations = 1000,
  maxNumberOfRawDatasetsPerStage = 1,
  seed = 314159)
```

```
head(sim1$sumdata)
head(sim1$rawdata)
```

---

lrsim3a *Log-rank test simulation for three arms*

Description

Performs simulation for three-arm group sequential trials based on weighted log-rank test. The looks are driven by the total number of events in Arm A and Arm C combined. Alternatively, the analyses can be planned to occur at specified calendar times.

Usage

```
lrsim3a(
  kMax = NA_integer_,
  hazardRatioH013 = 1,
  hazardRatioH023 = 1,
  hazardRatioH012 = 1,
  allocation1 = 1L,
  allocation2 = 1L,
  allocation3 = 1L,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  lambda3 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  gamma3 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  rho1 = 0,
  rho2 = 0,
  plannedEvents = NA_integer_,
  plannedTime = NA_real_,
  maxNumberOfIterations = 1000L,
  maxNumberOfRawDatasetsPerStage = 0L,
  seed = NA_integer_
)
```

Arguments

kMax	The maximum number of stages.
hazardRatioH013	Hazard ratio under the null hypothesis for arm 1 versus arm 3. Defaults to 1 for superiority test.
hazardRatioH023	Hazard ratio under the null hypothesis for arm 2 versus arm 3. Defaults to 1 for superiority test.
hazardRatioH012	Hazard ratio under the null hypothesis for arm 1 versus arm 2. Defaults to 1 for superiority test.
allocation1	Number of subjects in Arm A in a randomization block. Defaults to 1 for equal randomization.
allocation2	Number of subjects in Arm B in a randomization block. Defaults to 1 for equal randomization.
allocation3	Number of subjects in Arm C in a randomization block. Defaults to 1 for equal randomization.
accrualTime	Accrual time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
accrualIntensity	A vector of accrual intensities. One for each accrual time interval.
piecewiseSurvivalTime	A vector that specifies the time intervals for the piecewise exponential survival distribution. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
stratumFraction	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
lambda1	A vector of hazard rates for the event in each analysis time interval by stratum for arm 1.
lambda2	A vector of hazard rates for the event in each analysis time interval by stratum for arm 2.
lambda3	A vector of hazard rates for the event in each analysis time interval by stratum for arm 3.
gamma1	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 1.
gamma2	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 2.
gamma3	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 3.
accrualDuration	Duration of the enrollment period.

followupTime — Follow-up time for the last enrolled subject.

fixedFollowup — Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.

rho1 — The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

rho2 — The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

plannedEvents — The planned cumulative total number of events at Look 1 to Look kMax for Arms A and C combined.

plannedTime — The calendar times for the analyses. To use calendar time to plan the analyses, plannedEvents should be missing.

maxNumberOfIterations — The number of simulation iterations. Defaults to 1000.

maxNumberOfRawDatasetsPerStage — The number of raw datasets per stage to extract. Defaults to 1.

seed — The seed to reproduce the simulation results. The computer clock will be used if left unspecified,

Value

A list with 2 components:

* sumdata is a data frame of summary data by stage for each iteration, containing the analysis time, number of accrued subjects overall and by treatment group, number of events overall and by treatment group, number of dropouts overall and by treatment group, and log-rank test statistic for each comparison.

* rawdata (exists if maxNumberOfRawDatasetsPerStage is a positive integer) is a data frame for subject-level data for selected replications, containing the stage number, subject number, arrival time, stratum, treatment group, observation time, survival time, dropout time, time under observation, and event and dropout indicators.

Examples

```
sim1 = lrsim3a(
  kMax = 3,
  allocation1 = 2,
  allocation2 = 2,
  allocation3 = 1,
  accrualTime = c(0, 8),
  accrualIntensity = c(10, 28),
  piecewiseSurvivalTime = 0,
  lambda1 = log(2)/12*0.60,
  lambda2 = log(2)/12*0.70,
  lambda3 = log(2)/12,
  accrualDuration = 30.143,
  plannedEvents = c(186, 259, 295),
  maxNumberOfIterations = 1000,
  maxNumberOfRawDatasetsPerStage = 1,
  seed = 314159)
```

```
head(sim1$sumdata)
head(sim1$rawdata)
```

`lrstat` *Number of subjects having an event and log-rank statistics*

Description

Obtains the number of subjects accrued, number of events and number of dropouts in each group, mean and variance of weighted log-rank score statistic, estimated hazard ratio from weighted Cox regression and variance of log hazard ratio estimate at given calendar times.

Usage

```
lrstat(
  time = NA_real_,
  hazardRatioH0 = 1,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  rho1 = 0,
  rho2 = 0,
  numSubintervals = 300L,
  predictEventOnly = 0L
)
```

Arguments

`time`	A vector of calendar times at which to calculate the number of events and the mean and variance of log-rank test score statistic.
`hazardRatioH0`	Hazard ratio under the null hypothesis for the active treatment versus control. Defaults to 1 for superiority test.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.

accrualTime Accrual time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).

accrualIntensity
A vector of accrual intensities. One for each accrual time interval.

piecewiseSurvivalTime
A vector that specifies the time intervals for the piecewise exponential survival distribution. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.

stratumFraction
A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.

lambda1 A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.

lambda2 A vector of hazard rates for the event in each analysis time interval by stratum for the control group.

gamma1 The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.

gamma2 The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.

accrualDuration
Duration of the enrollment period.

followupTime Follow-up time for the last enrolled subject.

fixedFollowup Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.

rho1 The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

rho2 The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

numSubintervals
Number of sub-intervals to approximate the mean and variance of the weighted log-rank test score statistic. Defaults to 300. Specify a larger number for better approximation.

predictEventOnly
Whether to predict the number of events only. Defaults to 0 for obtaining log-rank score statistic mean and variance. Set `predictEventOnly = 1` for predicting the number of events only. Set `predictEventOnly = 2` for predicting the number of events, calculating the mean and variance of log-rank score statistic, and calculating the estimated hazard ratio and variance of log hazard ratio.

Value

A data frame of the number of subjects enrolled, the number of subjects having an event overall and in each group, the number of subjects who drop out overall and in each group, the mean and variance of weighted log-rank score statistic, the estimated hazard ratio from weighted Cox regression, and variance of the log hazard ratio estimate at the specified calendar times.

Examples

```
# Piecewise accrual, piecewise exponential survivals, and 5% dropout by
# the end of 1 year.

lrstat(time = c(22, 40), allocationRatioPlanned = 1,
       accrualTime = seq(0, 9),
       accrualIntensity = c(26/9*seq(1, 9), 26),
       piecewiseSurvivalTime = c(0, 6),
       stratumFraction = c(0.2, 0.8),
       lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
       lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
       gamma1 = -log(1-0.05)/12,
       gamma2 = -log(1-0.05)/12,
       accrualDuration = 22,
       followupTime = 18, fixedFollowup = FALSE)
```

---

qtpwexp *Quantile function of truncated piecewise exponential distribution*

Description

Obtains the quantile of a piecewise expoenential distribution given that it exceeds a specified lower bound.

Usage

```
qtpwexp(
  probability,
  piecewiseSurvivalTime = 0,
  lambda = 0.0578,
  lowerBound = 0
)
```

Arguments

probability	The scalar probability corresponding to the quantile.
piecewiseSurvivalTime	A vector that specifies the time intervals for the piecewise exponential survival distribution. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
lambda	A vector of hazard rates for the event. One for each analysis time interval.
lowerBound	The left truncation time point for the survival time. Defaults to 0 for no truncation.

Value

The quantile x such that P(X > x | X > lowerBound) = 1 - probability.

Examples

```
qtpwexp(probability = c(0.3, 0.5), piecewiseSurvivalTime = c(0, 6, 9, 15),
        lambda = c(0.025, 0.04, 0.015, 0.007))
```

`repeatedPValue` *Repeated p-values for group sequential design*

Description

Obtains the repeated p-values for a group sequential design.

Usage

```
repeatedPValue(
  kMax,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA,
  maxInformation = 1,
  p,
  information,
  spendingTime = NULL
)
```

Arguments

`kMax`	The maximum number of stages.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`maxInformation`	The target maximum information. Defaults to 1, in which case, `information` represents `informationRates`.
`p`	The raw p-values at look 1 to look `k`. It can be a matrix with `k` columns for `k <= kMax`.
`information`	The observed information by look. It can be a matrix with `k` columns.
`spendingTime`	The error spending time at each analysis, must be increasing and less than or equal to 1. Defaults to NULL, in which case, it is the same as `informationRates` derived from `information` and `maxInformation`. It can be a matrix with `k` columns.

Value

The repeated p-values at look 1 to look k.

Examples

```
# Example 1: informationRates different from spendingTime
repeatedPValue(kMax = 3, typeAlphaSpending = "sfOF",
               maxInformation = 800,
               p = c(0.2, 0.15, 0.1),
               information = c(529, 700, 800),
               spendingTime = c(0.6271186, 0.8305085, 1))

# Example 2: Maurer & Bretz (2013), current look is not the last look
repeatedPValue(kMax = 3, typeAlphaSpending = "sfOF",
               p = matrix(c(0.0062, 0.017, 0.009, 0.13,
                            0.0002, 0.0035, 0.002, 0.06),
                          nrow=4, ncol=2),
               information = c(1/3, 2/3))
```

---

updateGraph *Update graph for graphical approaches*

Description

Updates the weights and transition matrix for graphical approaches.

Usage

```
updateGraph(w, G, I, j)
```

Arguments

w	The current vector of weights for elementary hypotheses.
G	The current transition matrix.
I	The set of indices for yet to be rejected hypotheses.
j	The hypothesis to remove from index set I.

Value

A list containing the new vector of weights and the new transition matrix for the graph, and the new set of indices of yet to be rejected hypotheses.

Examples

```
updateGraph(w = c(0.5, 0.5, 0, 0),
            G = matrix(c(0, 0.5, 0.5, 0,  0.5, 0, 0, 0.5,
                         0, 1, 0, 0,  1, 0, 0, 0),
                       nrow=4, ncol=4, byrow=TRUE),
            I = c(1, 2, 3, 4),
            j = 1)
```

Index