Package 'lrstat'

December 22, 2023

Type Package

Title Power and Sample Size Calculation for Non-Proportional Hazards

Version 0.2.2

Date 2023-12-18

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), mvtnorm (>= 1.1-3)

LinkingTo Rcpp

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

VignetteBuilder knitr

RoxygenNote 7.2.3

Encoding UTF-8

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-12-22 04:30:03 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.

In addition to the log-rank test power and sample size calculations, the lrstat package can also be used for the following tasks:

- design generic group sequential trials for continuous, binary, or other endpoints.
- design adaptive group sequential trials for changes in sample size, error spending function, number and spacing or future looks.
- calculate the terminating and repeated confidence intervals for standard and adaptive group sequential trials.
- calculate the conditional power for non-proportional hazards with or without design changes.
- perform multiplicity adjustment based on graphical approaches using weighted Bonferroni tests, Bonferroni mixture of weighted Simes test, and Bonferroni mixture of Dunnett test as well as group sequential trials with multiple hypotheses.
- perform multiplicity adjustment using stepwise gatekeeping procedures for two sequences of hypotheses and the standard or modified mixture gatekeeping procedures in the general case.

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	A vector that specifies the starting time of piecewise Poisson enrollment 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.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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)
```

---

adaptDesign *Adaptive design at an interim look*

Description

Obtains the conditional power for specified incremental information given the interim results, parameter value, and data-dependent changes in the error spending function and the number and spacing of interim looks. Conversely, obtain the incremental information needed to attain a specified conditional power given the interim results, parameter value, and data-dependent changes in the error spending function and the number and spacing of interim looks.

Usage

```
adaptDesign(
  betaNew = NA_real_,
  INew = NA_real_,
  L = NA_integer_,
  zL = NA_real_,
  theta = NA_real_,
  IMax = NA_real_,
  kMax = NA_integer_,
  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_,
  spendingTime = NA_real_,
  MullerSchafer = 0L,
  kNew = NA_integer_,
  informationRatesNew = NA_real_,
  efficacyStoppingNew = NA_integer_,
  futilityStoppingNew = NA_integer_,
  typeAlphaSpendingNew = "sfOF",
  parameterAlphaSpendingNew = NA_real_,
  typeBetaSpendingNew = "none",
  parameterBetaSpendingNew = NA_real_,
  userBetaSpendingNew = NA_real_,
  spendingTimeNew = NA_real_
)
```

Arguments

betaNew	The type II error for the secondary trial.
INew	The maximum information of the secondary trial. Either betaNew or INew should be provided while the other one should be missing.
L	The interim adaptation look of the primary trial.
zL	The z-test statistic at the interim adaptation look of the primary trial.
theta	The parameter value.
IMax	The maximum information of the primary trial. Must be provided if futilityBounds is missing and typeBetaSpending is not equal to "none".
kMax	The maximum number of stages of the primary trial.
informationRates	The information rates of the primary trial.

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

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

`criticalValues`
: The upper boundaries on the z-test statistic scale for efficacy stopping for the primary trial.

`alpha`
: The significance level of the primary trial. Defaults to 0.025.

`typeAlphaSpending`
: The type of alpha spending for the primary trial. 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 of alpha spending for the primary trial. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".

`userAlphaSpending`
: The user defined alpha spending for the primary trial. Cumulative alpha spent up to each stage.

`futilityBounds`
: The lower boundaries on the z-test statistic scale for futility stopping for the primary trial. Defaults to `rep(-6, kMax-1)` if left unspecified.

`typeBetaSpending`
: The type of beta spending for the primary trial. 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 of beta spending for the primary trial. Corresponds to rho for "sfKD", and gamma for "sfHSD".

`spendingTime`
: The error spending time of the primary trial. Defaults to missing, in which case, it is the same as `informationRates`.

`MullerSchafer`
: Whether to use the Muller and Schafer (2001) method for trial adaptation.

`kNew`
: The number of looks of the secondary trial.

`informationRatesNew`
: The spacing of looks of the secondary trial.

`efficacyStoppingNew`
: The indicators of whether efficacy stopping is allowed at each look of the secondary trial. Defaults to true if left unspecified.

`futilityStoppingNew`
: The indicators of whether futility stopping is allowed at each look of the secondary trial. Defaults to true if left unspecified.

`typeAlphaSpendingNew`
: The type of alpha spending for the secondary trial. 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".

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

`typeBetaSpendingNew`
: The type of beta spending for the secondary trial. 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".

`parameterBetaSpendingNew`
: The parameter value of beta spending for the secondary trial. Corresponds to rho for "sfKD", and gamma for "sfHSD".

`userBetaSpendingNew`
: The user defined cumulative beta spending. Cumulative beta spent up to each stage of the secondary trial.

`spendingTimeNew`
: The error spending time of the secondary trial. Defaults to missing, in which case, it is the same as `informationRatesNew`.

Value

An `adaptDesign` object with two list components:

- `primaryTrial`: A list of selected information for the primary trial, including `L`, `zL`, `theta`, `kMax`, `informationRates`, `efficacyBounds`, `futilityBounds`, and `MullerSchafer`.
- `secondaryTrial`: A `design` object for the secondary trial.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

Lu Chi, H. M. James Hung, and Sue-Jane Wang. Modification of sample size in group sequential clinical trials. Biometrics 1999;55:853-857.

Hans-Helge Muller and Helmut Schafer. Adaptive group sequential designs for clinical trials: Combining the advantages of adaptive and of classical group sequential approaches. Biometrics 2001;57:886-891.

See Also

`getDesign`

Examples

```
# original group sequential design with 90% power to detect delta = 6
delta = 6
sigma = 17
n = 282
(des1 = getDesign(IMax = n/(4*sigma^2), theta = delta, kMax = 3,
                  alpha = 0.05, typeAlphaSpending = "sfHSD",
                  parameterAlphaSpending = -4))

# interim look results
L = 1
n1 = n/3
delta1 = 4.5
sigma1 = 20
zL = delta1/sqrt(4/n1*sigma1^2)

t = des1$byStageResults$informationRates

# conditional power with sample size increase
(des2 = adaptDesign(
  betaNew = NA, INew = 420/(4*sigma1^2),
  L, zL, theta = delta1,
  IMax = n/(4*sigma1^2), kMax = 3, informationRates = t,
  alpha = 0.05, typeAlphaSpending = "sfHSD",
  parameterAlphaSpending = -4))

# Muller & Schafer (2001) method to design the secondary trial:
# 3-look gamma(-2) spending with 84% power at delta = 4.5 and sigma = 20
(des2 = adaptDesign(
  betaNew = 0.16, INew = NA,
  L, zL, theta = delta1,
  IMax = n/(4*sigma1^2), kMax = 3, informationRates = t,
  alpha = 0.05, typeAlphaSpending = "sfHSD",
  parameterAlphaSpending = -4,
  MullerSchafer = TRUE,
  kNew = 3, typeAlphaSpendingNew = "sfHSD",
  parameterAlphaSpendingNew = -2))

# incremental sample size for sigma = 20
(nNew = 4*sigma1^2*des2$secondaryTrial$overallResults$maxInformation)
```

`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	A vector that specifies the starting time of piecewise Poisson enrollment 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 starting time of piecewise exponential survival time intervals. 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.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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:

- `exitProbUpper`: The vector of efficacy stopping probabilities
- `exitProbLower`: The vector of futility stopping probabilities.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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)
```

`fadjpdun` *Adjusted p-values for Dunnett-based graphical approaches*

Description

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

Usage

```
fadjpdun(wgtmat, p, family = NULL, corr = NULL)
```

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.
`corr`	The correlation matrix that should be used for the parametric test. Can contain NAs for unknown correlations between families.

Value

A matrix of adjusted p-values.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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.

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)
corr = matrix(c(1,0.5,NA,NA, 0.5,1,NA,NA,
                NA,NA,1,0.5, NA,NA,0.5,1),
              nrow = 4, byrow = TRUE)
fadjpdun(wgtmat, pvalues, family, corr)
```

`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 = NULL)
```

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.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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 = NULL,
  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.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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).

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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 = NULL,
  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.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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)
```

ftrunc *Adjusted p-values for Holm, Hochberg, or Hommel procedures*

Description

Obtains the adjusted p-values for possibly truncated Holm, Hochberg, or Hommel procedures.

Usage

```
ftrunc(p, test = "hommel", gamma = 1)
```

Arguments

p	The raw p-values for elementary hypotheses.
test	The test to use, e.g., "holm", "hochberg", or "hommel" (default).
gamma	The value of the truncation parameter. Defaults to 1 for the regular Holm, Hochberg, or Hommel procedure.

Value

A matrix of adjusted p-values.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

Alex Dmitrienko, Ajit C. Tamhane, and Brian L. Wiens. General multistage gatekeeping procedures. Biometrical Journal. 2008; 5:667-677.

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)
ftrunc(pvalues, "hochberg")
```

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.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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))
```

getAccrualDurationFromN *Accrual duration to enroll target number of subjects*

Description

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

Usage

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

Arguments

nsubjects	The vector of target number of subjects.
accrualTime	A vector that specifies the starting time of piecewise Poisson enrollment 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

A vector of accrual durations.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

Examples

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

---

getADCI *Confidence interval after adaptation*

Description

Obtains the p-value, median unbiased point estimate, and confidence interval after the end of an adaptive trial.

Usage

```
getADCI(
  L = NA_integer_,
  zL = NA_real_,
  IMax = NA_real_,
  kMax = NA_integer_,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.25,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  spendingTime = NA_real_,
  L2 = NA_integer_,
  zL2 = NA_real_,
  INew = NA_real_,
```

```
  MullerSchafer = 0L,
  informationRatesNew = NA_real_,
  efficacyStoppingNew = NA_integer_,
  typeAlphaSpendingNew = "sfOF",
  parameterAlphaSpendingNew = NA_real_,
  spendingTimeNew = NA_real_
)
```

Arguments

L	The interim adaptation look of the primary trial.
zL	The z-test statistic at the interim adaptation look of the primary trial.
IMax	The maximum information of the primary trial.
kMax	The maximum number of stages of the primary trial.
informationRates	The information rates of the primary trial.
efficacyStopping	Indicators of whether efficacy stopping is allowed at each stage of the primary trial. Defaults to true if left unspecified.
criticalValues	The upper boundaries on the z-test statistic scale for efficacy stopping for the primary trial.
alpha	The significance level of the primary trial. Defaults to 0.025.
typeAlphaSpending	The type of alpha spending for the primary trial. 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".
parameterAlphaSpending	The parameter value of alpha spending for the primary trial. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
spendingTime	The error spending time of the primary trial. Defaults to missing, in which case, it is the same as informationRates.
L2	The termination look of the secondary trial.
zL2	The z-test statistic at the termination look of the secondary trial.
INew	The maximum information of the secondary trial.
MullerSchafer	Whether to use the Muller and Schafer (2001) method for trial adaptation.
informationRatesNew	The spacing of looks of the secondary trial up to look L2.
efficacyStoppingNew	The indicators of whether efficacy stopping is allowed at each look of the secondary trial up to look L2. Defaults to true if left unspecified.

typeAlphaSpendingNew
: The type of alpha spending for the secondary trial. 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".

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

spendingTimeNew
: The error spending time of the secondary trial up to look L2. Defaults to missing, in which case, it is the same as informationRatesNew.

Value

A list with the following components:

- pvalue: p-value for rejecting the null hypothesis.
- thetahat: Median unbiased point estimate of the parameter.
- cilevel: Confidence interval level.
- lower: Lower bound of confidence interval.
- upper: Upper bound of confidence interval.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

Ping Gao, Lingyun Liu and Cyrus Mehta. Exact inference for adaptive group sequential designs. Stat Med. 2013;32(23):3991-4005.

See Also

adaptDesign

Examples

```
# original group sequential design with 90% power to detect delta = 6
delta = 6
sigma = 17
n = 282
(des1 = getDesign(IMax = n/(4*sigma^2), theta = delta, kMax = 3,
                  alpha = 0.05, typeAlphaSpending = "sfHSD",
                  parameterAlphaSpending = -4))

# interim look results
```

```
L = 1
n1 = n/3
delta1 = 4.5
sigma1 = 20
zL = delta1/sqrt(4/n1*sigma1^2)

t = des1$byStageResults$informationRates

# Muller & Schafer (2001) method to design the secondary trial:
des2 = adaptDesign(
  betaNew = 0.2, L = L, zL = zL, theta = 5,
  kMax = 3, informationRates = t,
  alpha = 0.05, typeAlphaSpending = "sfHSD",
  parameterAlphaSpending = -4,
  MullerSchafer = TRUE,
  kNew = 3, typeAlphaSpendingNew = "sfHSD",
  parameterAlphaSpendingNew = -2)

n2 = ceiling(des2$secondaryTrial$overallResults$maxInformation*4*20^2)
ns = round(n2*(1:3)/3)
 (des2 = adaptDesign(
   INew = n2/(4*20^2), L = L, zL = zL, theta = 5,
   kMax = 3, informationRates = t,
   alpha = 0.05, typeAlphaSpending = "sfHSD",
   parameterAlphaSpending = -4,
   MullerSchafer = TRUE,
   kNew = 3, informationRatesNew = ns/n2,
   typeAlphaSpendingNew = "sfHSD",
   parameterAlphaSpendingNew = -2))

# termination at the second look of the secondary trial
L2 = 2
delta2 = 6.86
sigma2 = 21.77
zL2 = delta2/sqrt(4/197*sigma2^2)

t2 = des2$secondaryTrial$byStageResults$informationRates[1:L2]

# confidence interval
getADCI(L = L, zL = zL,
        IMax = n/(4*sigma1^2), kMax = 3,
        informationRates = t,
        alpha = 0.05, typeAlphaSpending = "sfHSD",
        parameterAlphaSpending = -4,
        L2 = L2, zL2 = zL2,
        INew = n2/(4*sigma2^2),
        MullerSchafer = TRUE,
        informationRatesNew = t2,
        typeAlphaSpendingNew = "sfHSD",
        parameterAlphaSpendingNew = -2)
```

getADRCI *Repeated confidence interval after adaptation*

Description

Obtains the repeated p-value, conservative point estimate, and repeated confidence interval for an adaptive group sequential trial.

Usage

```
getADRCI(
  L = NA_integer_,
  zL = NA_real_,
  IMax = NA_real_,
  kMax = NA_integer_,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.25,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  spendingTime = NA_real_,
  L2 = NA_integer_,
  zL2 = NA_real_,
  INew = NA_real_,
  MullerSchafer = 0L,
  informationRatesNew = NA_real_,
  efficacyStoppingNew = NA_integer_,
  typeAlphaSpendingNew = "sfOF",
  parameterAlphaSpendingNew = NA_real_,
  spendingTimeNew = NA_real_
)
```

Arguments

L	The interim adaptation look of the primary trial.
zL	The z-test statistic at the interim adaptation look of the primary trial.
IMax	The maximum information of the primary trial.
kMax	The maximum number of stages of the primary trial.
informationRates	The information rates of the primary trial.
efficacyStopping	Indicators of whether efficacy stopping is allowed at each stage of the primary trial. Defaults to true if left unspecified.
criticalValues	The upper boundaries on the z-test statistic scale for efficacy stopping for the primary trial.

alpha The significance level of the primary trial. Defaults to 0.025.

typeAlphaSpending
: The type of alpha spending for the primary trial. 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".

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

spendingTime The error spending time of the primary trial. Defaults to missing, in which case, it is the same as informationRates.

L2 The look of interest in the secondary trial.

zL2 The z-test statistic at the look of the secondary trial.

INew The maximum information of the secondary trial.

MullerSchafer Whether to use the Muller and Schafer (2001) method for trial adaptation.

informationRatesNew
: The spacing of looks of the secondary trial.

efficacyStoppingNew
: The indicators of whether efficacy stopping is allowed at each look of the secondary trial up to look L2. Defaults to true if left unspecified.

typeAlphaSpendingNew
: The type of alpha spending for the secondary trial. 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".

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

spendingTimeNew
: The error spending time of the secondary trial. up to look L2. Defaults to missing, in which case, it is the same as informationRatesNew.

Value

A list with the following components:

- pvalue: Repeated p-value for rejecting the null hypothesis.
- thetahat: Point estimate of the parameter.
- cilevel: Confidence interval level.
- lower: Lower bound of repeated confidence interval.
- upper: Upper bound of repeated confidence interval.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

Cyrus R. Mehta, Peter Bauer, Martin Posch and Werner Brannath. Repeated confidence intervals for adaptive group sequential trials. Stat Med. 2007;26:5422–5433.

See Also

adaptDesign

Examples

```
# original group sequential design with 90% power to detect delta = 6
delta = 6
sigma = 17
n = 282
(des1 = getDesign(IMax = n/(4*sigma^2), theta = delta, kMax = 3,
                  alpha = 0.05, typeAlphaSpending = "sfHSD",
                  parameterAlphaSpending = -4))

# interim look results
L = 1
n1 = n/3
delta1 = 4.5
sigma1 = 20
zL = delta1/sqrt(4/n1*sigma1^2)

t = des1$byStageResults$informationRates

# Muller & Schafer (2001) method to design the secondary trial:
des2 = adaptDesign(
  betaNew = 0.2, L = L, zL = zL, theta = 5,
  kMax = 3, informationRates = t,
  alpha = 0.05, typeAlphaSpending = "sfHSD",
  parameterAlphaSpending = -4,
  MullerSchafer = TRUE,
  kNew = 3, typeAlphaSpendingNew = "sfHSD",
  parameterAlphaSpendingNew = -2)

n2 = ceiling(des2$secondaryTrial$overallResults$maxInformation*4*20^2)
ns = round(n2*(1:3)/3)
(des2 = adaptDesign(
  INew = n2/(4*20^2), L = L, zL = zL, theta = 5,
  kMax = 3, informationRates = t,
  alpha = 0.05, typeAlphaSpending = "sfHSD",
  parameterAlphaSpending = -4,
  MullerSchafer = TRUE,
  kNew = 3, informationRatesNew = ns/n2,
  typeAlphaSpendingNew = "sfHSD",
```

```
  parameterAlphaSpendingNew = -2))

# termination at the second look of the secondary trial
L2 = 2
delta2 = 6.86
sigma2 = 21.77
zL2 = delta2/sqrt(4/197*sigma2^2)

t2 = des2$secondaryTrial$byStageResults$informationRates[1:L2]

# repeated confidence interval
getADRCI(L = L, zL = zL,
         IMax = n/(4*sigma1^2), kMax = 3,
         informationRates = t,
         alpha = 0.05, typeAlphaSpending = "sfHSD",
         parameterAlphaSpending = -4,
         L2 = L2, zL2 = zL2,
         INew = n2/(4*sigma2^2),
         MullerSchafer = TRUE,
         informationRatesNew = t2,
         typeAlphaSpendingNew = "sfHSD",
         parameterAlphaSpendingNew = -2)
```

---

getBound *Get efficacy boundaries for group sequential design*

Description

Obtains the efficacy stopping boundaries for a group sequential design.

Usage

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

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`.

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

Details

If `typeAlphaSpending` is "OF", "P", or "WT", then the boundaries will be based on equally spaced looks.

Value

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

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

Examples

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

---

getCI *Confidence interval after trial termination*

Description

Obtains the p-value, median unbiased point estimate, and confidence interval after the end of a group sequential trial.

Usage

```
getCI(
  L = NA_integer_,
  zL = NA_real_,
  IMax = NA_real_,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

L	The termination look.
zL	The z-test statistic at the termination look.
IMax	The maximum information of the trial.
informationRates	The information rates up to look L.
efficacyStopping	Indicators of whether efficacy stopping is allowed at each stage up to look L. Defaults to true if left unspecified.
criticalValues	The upper boundaries on the z-test statistic scale for efficacy stopping up to look L.
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, and "none" for no early efficacy stopping. Defaults to "sfOF".
parameterAlphaSpending	The parameter value of alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
spendingTime	The error spending time up to look L. Defaults to missing, in which case, it is the same as informationRates.

Value

A list with the following components:

- pvalue: p-value for rejecting the null hypothesis.
- thetahat: Median unbiased point estimate of the parameter.

- `cilevel`: Confidence interval level.
- `lower`: Lower bound of confidence interval.
- `upper`: Upper bound of confidence interval.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

Anastasios A. Tsiatis, Gary L. Rosner and Cyrus R. Mehta. Exact confidence intervals following a group sequential test. Biometrics 1984;40:797-803.

Examples

```
# group sequential design with 90% power to detect delta = 6
delta = 6
sigma = 17
n = 282
(des1 = getDesign(IMax = n/(4*sigma^2), theta = delta, kMax = 3,
                  alpha = 0.05, typeAlphaSpending = "sfHSD",
                  parameterAlphaSpending = -4))

# crossed the boundary at the second look
L = 2
n1 = n*2/3
delta1 = 7
sigma1 = 20
zL = delta1/sqrt(4/n1*sigma1^2)

# confidence interval
getCI(L = L, zL = zL, IMax = n/(4*sigma1^2),
      informationRates = c(1/3, 2/3), alpha = 0.05,
      typeAlphaSpending = "sfHSD", parameterAlphaSpending = -4)
```

`getCP` *Conditional power allowing for varying parameter values*

Description

Obtains the conditional power for specified incremental information given the interim results, parameter values, and data-dependent changes in the error spending function and the number and spacing of interim looks.

Usage

```
getCP(
  INew = NA_real_,
  L = NA_integer_,
  zL = NA_real_,
  theta = NA_real_,
  IMax = NA_real_,
  kMax = NA_integer_,
  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_,
  spendingTime = NA_real_,
  MullerSchafer = 0L,
  kNew = NA_integer_,
  informationRatesNew = NA_real_,
  efficacyStoppingNew = NA_integer_,
  futilityStoppingNew = NA_integer_,
  typeAlphaSpendingNew = "sfOF",
  parameterAlphaSpendingNew = NA_real_,
  typeBetaSpendingNew = "none",
  parameterBetaSpendingNew = NA_real_,
  spendingTimeNew = NA_real_
)
```

Arguments

INew	The maximum information of the secondary trial.
L	The interim adaptation look of the primary trial.
zL	The z-test statistic at the interim adaptation look of the primary trial.
theta	A scalar or a vector of parameter values of length kMax + kMax - L if MullerSchafer = FALSE or length kMax + kNew if MullerSchafer = TRUE.
IMax	The maximum information of the primary trial.
kMax	The maximum number of stages of the primary trial.
informationRates	The information rates of the primary trial.
efficacyStopping	Indicators of whether efficacy stopping is allowed at each stage of the primary trial. Defaults to true if left unspecified.

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

`criticalValues`
: The upper boundaries on the z-test statistic scale for efficacy stopping for the primary trial.

`alpha`
: The significance level of the primary trial. Defaults to 0.025.

`typeAlphaSpending`
: The type of alpha spending for the primary trial. 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 of alpha spending for the primary trial. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".

`userAlphaSpending`
: The user defined alpha spending for the primary trial. Cumulative alpha spent up to each stage.

`futilityBounds`
: The lower boundaries on the z-test statistic scale for futility stopping for the primary trial. Defaults to `rep(-6, kMax-1)` if left unspecified.

`typeBetaSpending`
: The type of beta spending for the primary trial. 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 of beta spending for the primary trial. Corresponds to rho for "sfKD", and gamma for "sfHSD".

`spendingTime`
: The error spending time of the primary trial. Defaults to missing, in which case, it is the same as `informationRates`.

`MullerSchafer`
: Whether to use the Muller and Schafer (2001) method for trial adaptation.

`kNew`
: The number of looks of the secondary trial.

`informationRatesNew`
: The spacing of looks of the secondary trial.

`efficacyStoppingNew`
: The indicators of whether efficacy stopping is allowed at each look of the secondary trial. Defaults to true if left unspecified.

`futilityStoppingNew`
: The indicators of whether futility stopping is allowed at each look of the secondary trial. Defaults to true if left unspecified.

`typeAlphaSpendingNew`
: The type of alpha spending for the secondary trial. 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".

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

`typeBetaSpendingNew`
: The type of beta spending for the secondary trial. 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".

`parameterBetaSpendingNew`
: The parameter value of beta spending for the secondary trial. Corresponds to rho for "sfKD", and gamma for "sfHSD".

`spendingTimeNew`
: The error spending time of the secondary trial. Defaults to missing, in which case, it is the same as `informationRatesNew`.

Value

The conditional power given the interim results, parameter values, and data-dependent design changes.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

Cyrus R. Mehta and Stuart J. Pocock. Adaptive increase in sample size when interim results are promising: A practical guide with examples. Stat Med. 2011;30:3267–3284.

See Also

`getDesign`

Examples

```
# Conditional power calculation with delayed treatment effect

# Two interim analyses have occurred with 179 and 266 events, respectively
# The observed hazard ratio at the second interim look is 0.81

trialsdt = as.Date("2020-03-04")                      # trial start date
iadt = c(as.Date("2022-02-01"), as.Date("2022-11-01")) # interim dates
mo1 = as.numeric(iadt - trialsdt + 1)/30.4375         # interim months
```

```
# Assume a piecewise Poisson enrollment process with a 8-month ramp-up and
# 521 patients were enrolled after 17.94 months
N = 521                   # total number of patients
Ta = 17.94                # enrollment duration
Ta1 = 8                   # assumed end of enrollment ramp-up
enrate = N / (Ta - Ta1/2) # enrollment rate after ramp-up

# Assume a median survival of 16.7 months for the control group, a 5-month
# delay in treatment effect, and a hazard ratio of 0.7 after the delay
lam1 = log(2)/16.7  # control group hazard of exponential distribution
t1 = 5              # months of delay in treatment effect
hr = 0.7            # hazard ratio after delay
lam2 = hr*lam1      # treatment group hazard after delay

# Assume an annual dropout rate of 5%
gam = -log(1-0.05)/12  # hazard for dropout

# The original target number of events was 298 and the new target is 335
mo2 <- caltime(
  nevents = c(298, 335),
  allocationRatioPlanned = 1,
  accrualTime = seq(0, Ta1),
  accrualIntensity = enrate*seq(1, Ta1+1)/(Ta1+1),
  piecewiseSurvivalTime = c(0, t1),
  lambda1 = c(lam1, lam2),
  lambda2 = c(lam1, lam1),
  gamma1 = gam,
  gamma2 = gam,
  accrualDuration = Ta,
  followupTime = 1000)

# expected number of events and average hazard ratios
(lr1 <- lrstat(
  time = c(mo1, mo2),
  accrualTime = seq(0, Ta1),
  accrualIntensity = enrate*seq(1, Ta1+1)/(Ta1+1),
  piecewiseSurvivalTime = c(0, t1),
  lambda1 = c(lam1, lam2),
  lambda2 = c(lam1, lam1),
  gamma1 = gam,
  gamma2 = gam,
  accrualDuration = Ta,
  followupTime = 1000,
  predictTarget = 3))


hr2 = 0.81                    # observed hazard ratio at interim 2
z2 = (-log(hr2))*sqrt(266/4)  # corresponding z-test statistic value

# expected mean of -log(HR) at the original looks and the new final look
theta = -log(lr1$HR[c(1,2,3,4)])

# conditional power with sample size increase
```

```
getCP(INew = (335 - 266)/4,
      L = 2, zL = z2, theta = theta,
      IMax = 298/4, kMax = 3,
      informationRates = c(179, 266, 298)/298,
      alpha = 0.025, typeAlphaSpending = "sfOF")
```

`getDesign` *Get group sequential design*

Description

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

Usage

```
getDesign(
  beta = NA_real_,
  IMax = NA_real_,
  theta = 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`	The type II error.
`IMax`	The maximum information. Either `beta` or `IMax` should be provided while the other one should be missing.
`theta`	The parameter value.
`kMax`	The maximum number of stages.

`informationRates`
: The information rates. 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

An S3 class `design` object with three components:

- `overallResults`: A data frame containing the following variables:
  - `overallReject`: The overall rejection probability.
  - `alpha`: The overall significance level.

  - kMax: The number of stages.
  - theta: The parameter value.
  - maxInformation: The maximum information.
  - expectedInformationH1: The expected information under H1.
  - expectedInformationH0: The expected information under H0.
  - drift: The drift parameter, equal to theta*sqrt(maxInformation).
  - inflationFactor: The inflation factor (relative to the fixed design).

- byStageResults: A data frame containing the following variables:
  - informationRates: The information rates.
  - efficacyBounds: The efficacy boundaries on the Z-scale.
  - futilityBounds: The futility boundaries on the Z-scale.
  - rejectPerStage: The probability for efficacy stopping.
  - futilityPerStage: The probability for futility stopping.
  - cumulativeRejection: The cumulative probability for efficacy stopping.
  - cumulativeFutility: The cumulative probability for futility stopping.
  - cumulativeAlphaSpent: The cumulative alpha spent.
  - efficacyTheta: The efficacy boundaries on the parameter scale.
  - futilityTheta: The futility boundaries on the parameter scale.
  - efficacyP: The efficacy boundaries on the p-value scale.
  - futilityP: The futility boundaries on the p-value scale.
  - information: The cumulative information.
  - efficacyStopping: Whether to allow efficacy stopping.
  - futilityStopping: Whether to allow futility stopping.

- settings: A list containing the following input parameters:
  - typeAlphaSpending: The type of alpha spending.
  - parameterAlphaSpending: The parameter value for alpha spending.
  - userAlphaSpending: The user defined alpha spending.
  - typeBetaSpending: The type of beta spending.
  - parameterBetaSpending: The parameter value for beta spending.
  - userBetaSpending: The user defined beta spending.
  - spendingTime: The error spending time at each analysis.
  - calculationTarget: The calculation target, beta or IMax.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

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

Examples

```
# Example 1: obtain the maximum information given power
getDesign(beta = 0.2, theta = -log(0.7),
          kMax = 2, informationRates = c(0.5,1),
          alpha = 0.025, typeAlphaSpending = "sfOF",
          typeBetaSpending = "sfP")


# Example 2: obtain power given the maximum information
getDesign(IMax = 72.5, theta = -log(0.7),
          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`
: A vector that specifies the starting time of piecewise Poisson enrollment 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 starting time of piecewise exponential survival time intervals. 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 the following variables:

- `nevents`: The target number of events.
- `fixedFollowup`: Whether a fixed follow-up design is used.
- `accrualDuration`: The accrual duration.
- `subjects`: The total number of subjects.
- `studyDuration`: The study duration.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

Examples

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

---

getRCI *Repeated confidence interval for group sequential design*

Description

Obtains the repeated confidence interval for a group sequential trial.

Usage

```
getRCI(
  L = NA_integer_,
  zL = NA_real_,
  IMax = NA_real_,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

L	The look of interest.
zL	The z-test statistic at the look.
IMax	The maximum information of the trial.
informationRates	The information rates up to look L.
efficacyStopping	Indicators of whether efficacy stopping is allowed at each stage up to look L. Defaults to true if left unspecified.
criticalValues	The upper boundaries on the z-test statistic scale for efficacy stopping up to look L.
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, and "none" for no early efficacy stopping. Defaults to "sfOF".
parameterAlphaSpending	The parameter value of alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
spendingTime	The error spending time up to look L. Defaults to missing, in which case, it is the same as informationRates.

Value

A list with the following components:

- pvalue: Repeated p-value for rejecting the null hypothesis.
- thetahat: Point estimate of the parameter.
- cilevel: Confidence interval level.
- lower: Lower bound of repeated confidence interval.
- upper: Upper bound of repeated confidence interval.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

Christopher Jennison and Bruce W. Turnbull. Interim analyses: the repeated confidence interval approach (with discussion). J R Stat Soc Series B. 1989;51:305-361.

Examples

```
# group sequential design with 90% power to detect delta = 6
delta = 6
sigma = 17
n = 282
(des1 = getDesign(IMax = n/(4*sigma^2), theta = delta, kMax = 3,
                  alpha = 0.05, typeAlphaSpending = "sfHSD",
                  parameterAlphaSpending = -4))

# results at the second look
L = 2
n1 = n*2/3
delta1 = 7
sigma1 = 20
zL = delta1/sqrt(4/n1*sigma1^2)

# repeated confidence interval
getRCI(L = L, zL = zL, IMax = n/(4*sigma1^2),
       informationRates = c(1/3, 2/3), alpha = 0.05,
       typeAlphaSpending = "sfHSD", parameterAlphaSpending = -4)
```

---

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	A vector that specifies the starting time of piecewise Poisson enrollment 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 starting time of piecewise exponential survival time intervals. 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 containing the following variables:

- `time`: The calendar time at which to calculate the milestone survival.

- subjects: The enrolled number of subjects.
- milestone: The milestone time relative to randomization.
- surv1: The milestone survival probability for the treatment group.
- surv2: The milestone survival probability for the control group.
- vsurv1: The variance for surv1.
- vsurv2: The variance for surv2.
- survdiff: The difference in milestone survival probabilities, i.e., surv1 - surv2.
- vsurvdiff: The variance for survdiff.
- survdiffZ: The Z-statistic value, i.e., survdiff/sqrt(vsurvdiff).

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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 for the conventional log-rank test and in terms of the actual information for weighted log-rank tests. 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 A vector that specifies the starting time of piecewise Poisson enrollment 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 starting time of piecewise exponential survival time intervals. 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
The type of computation, either "direct" for the direct approximation method, or "schoenfeld" for 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

An S3 class lrpower object with 4 components:

- overallResults: A data frame containing the following variables:
  - overallReject: The overall rejection probability.
  - alpha: The overall significance level.
  - numberOfEvents: The total number of events.
  - numberOfDropouts: The total number of dropouts.
  - numbeOfSubjects: The total number of subjects.
  - studyDuration: The total study duration.
  - expectedNumberOfEvents: The expected number of events.
  - expectedNumberOfDropouts: The expected number of dropouts.

  - expectedNumberOfSubjects: The expected number of subjects.
  - expectedStudyDuration: The expected study duration.
  - accrualDuration: The accrual duration.
  - followupTime: The follow-up duration.
  - fixedFollowup: Whether a fixed follow-up design is used.
  - rho1: The first parameter of the Fleming-Harrington family of weighted log-rank test.
  - rho2: The second parameter of the Fleming-Harrington family of weighted log-rank test.
  - allocationRatioPlanned: Allocation ratio for the active treatment versus control.
  - kMax: The number of stages.
  - hazardRatioH0: The hazard ratio under the null hypothesis.
  - etimateHazardRatio: Whether to estimate the hazard ratio.
  - typeOfComputation: The type of computation, either "direct" for the direct approximation method, or "schoenfeld" for the Schoenfeld method.

- byStageResults: A data frame containing the following variables:
  - informationRates: The information rates.
  - efficacyBounds: The efficacy boundaries on the Z-scale.
  - futilityBounds: The futility boundaries on the Z-scale.
  - rejectPerStage: The probability for efficacy stopping.
  - futilityPerStage: The probability for futility stopping.
  - cumulativeRejection: The cumulative probability for efficacy stopping.
  - cumulativeFutility: The cumulative probability for futility stopping.
  - cumulativeAlphaSpent: The cumulative alpha spent.
  - numberOfEvents: The number of events.
  - numberOfDropouts: The number of dropouts.
  - numberOfSubjects: The number of subjects.
  - analysisTime: The average time since trial start.
  - efficacyHR: The efficacy boundaries on the hazard ratio scale.
  - futilityHR: The futility boundaries on the hazard ratio scale.
  - efficacyP: The efficacy boundaries on the p-value scale.
  - futilityP: The futility boundaries on the p-value scale.
  - information: The cumulative information.
  - HR: The average hazard ratio.
  - efficacyStopping: Whether to allow efficacy stopping.
  - futilityStopping: Whether to allow futility stopping.

- settings: A list containing the following input parameters: typeAlphaSpending, parameterAlphaSpending, userAlphaSpending, typeBetaSpending, parameterBetaSpending, userBetaSpending, accrualTime, accuralIntensity, piecewiseSurvivalTime, stratumFraction, lambda1, lambda2, gamma1, gamma2, and spendingTime.
- byTreatmentCounts: A list containing the following counts by treatment group:
  - numberOfEvents1: The number of events by stage for the treatment group.
  - numberOfDropouts1: The number of dropouts by stage for the treatment group.
  - numberOfSubjects1: The number of subjects by stage for the treatment group.

- numberOfEvents2: The number of events by stage for the control group.
- numberOfDropouts2: The number of dropouts by stage for the control group.
- numberOfSubjects2: The number of subjects by stage for the control group.
- expectedNumberOfEvents1: The expected number of events for the treatment group.
- expectedNumberOfDropouts1: The expected number of dropouts for the treatment group.
- expectedNumberOfSubjects1: The expected number of subjects for the treatment group.
- expectedNumberOfEvents2: The expected number of events for control group.
- expectedNumberOfDropouts2: The expected number of dropouts for the control group.
- expectedNumberOfSubjects2: The expected number of subjects for the control group.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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 = NA_real_,
  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 for the conventional log-rank test and in terms of the actual information for weighted log-rank tests. 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 A vector that specifies the starting time of piecewise Poisson enrollment 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 starting time of piecewise exponential survival time intervals. 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
The type of computation, either "direct" for the direct approximation method, or "schoenfeld" for 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 two components:

- `resultsUnderH1`: An S3 class `lrpower` object under the alternative hypothesis.
- `resultsUnderH0`: An S3 class `lrpower` object under the null hypothesis.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

See Also

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)
```

`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_,
  informationRates = 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.
informationRates	The information rates in terms of number of events for the conventional log-rank test and in terms of the actual information 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	A vector that specifies the starting time of piecewise Poisson enrollment 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 starting time of piecewise exponential survival time intervals. 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

An S3 class lrsim object with 3 components:

- overview: A list containing the following information:
  - rejectPerStage: The efficacy stopping probability by stage.
  - futilityPerStage: The futility stopping probability by stage.
  - cumulativeRejection: Cumulative efficacy stopping probability by stage.
  - cumulativeFutility: The cumulative futility stopping probability by stage.
  - numberOfEvents: The average number of events by stage.
  - numberOfDropouts: The average number of dropouts by stage.
  - numberOfSubjects: The average number of subjects by stage.
  - analysisTime: The average analysis time by stage.
  - overallReject: The overall rejection probability.
  - expectedNumberOfEvents: The expected number of events for the overall study.
  - expectedNumberOfDropouts: The expected number of dropouts for the overall study.
  - expectedNumberOfSubjects: The expected number of subjects for the overall study.
  - expectedStudyDuration: The expected study duration.
  - hazardRatioH0: Hazard ratio under the null hypothesis for the active treatment versus control.
  - useEvents: whether the analyses are planned based on the number of events or calendar time.
  - accrualDuration: Duration of the enrollment period.
  - fixedFollowup: Whether a fixed follow-up design is used.
  - 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.
  - kMax: The maximum number of stages.
- sumdata: A data frame of summary data by iteration and stage:
  - iterationNumber: The iteration number.
  - stopStage: The stage at which the trial stops.

  - eventsNotAchieved: Whether the target number of events is not achieved for the iteration.
  - stageNumber: The stage number, covering all stages even if the trial stops at an interim look.
  - analysisTime: The time for the stage since trial start.
  - accruals1: The number of subjects enrolled at the stage for the treatment group.
  - accruals2: The number of subjects enrolled at the stage for the control group.
  - totalAccruals: The total number of subjects enrolled at the stage.
  - events1: The number of events at the stage for the treatment group.
  - events2: The number of events at the stage for the control group.
  - totalEvents: The total number of events at the stage.
  - dropouts1: The number of dropouts at the stage for the treatment group.
  - dropouts2: The number of dropouts at the stage for the control group.
  - totalDropouts: The total number of dropouts at the stage.
  - uscore: The numerator of the log-rank test statistic.
  - vscore: The variance of the log-rank test statistic.
  - logRankStatistic: The log-rank test Z-statistic.
  - rejectPerStage: Whether to reject the null hypothesis at the stage.
  - futilityPerStage: Whether to stop the trial for futility at the stage.

- rawdata (exists if maxNumberOfRawDatasetsPerStage is a positive integer): A data frame for subject-level data for selected replications, containing the following variables:
  - iterationNumber: The iteration number.
  - stopStage: The stage at which the trial stops.
  - analysisTime: The time for the stage since trial start.
  - subjectId: The subject ID.
  - arrivalTime: The enrollment time for the subject.
  - stratum: The stratum for the subject.
  - treatmentGroup: The treatment group (1 or 2) for the subject.
  - survivalTime: The underlying survival time for the subject.
  - dropoutTime: The underlying dropout time for the subject.
  - timeUnderObservation: The time under observation since since randomization.
  - event: Whether the subject experienced the event.
  - dropoutEvent: Whether the subject dropped out.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

Examples

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

sim1 = lrsim(kMax = 2, informationRates = 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, informationRates = 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	A vector that specifies the starting time of piecewise Poisson enrollment 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 starting time of piecewise exponential survival time intervals. 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: A data frame of summary data by iteration and stage:
  - iterationNumber: The iteration number.
  - eventsNotAchieved: Whether the target number of events is not achieved for the iteration.
  - stageNumber: The stage number, covering all stages even if the trial stops at an interim look.
  - analysisTime: The time for the stage since trial start.
  - accruals1: The number of subjects enrolled at the stage for the treatment group.
  - accruals2: The number of subjects enrolled at the stage for the control group.
  - totalAccruals: The total number of subjects enrolled at the stage.
  - endpoint: The endpoint (1 or 2) under consideration.
  - events1: The number of events at the stage for the treatment group.
  - events2: The number of events at the stage for the control group.
  - totalEvents: The total number of events at the stage.
  - dropouts1: The number of dropouts at the stage for the treatment group.
  - dropouts2: The number of dropouts at the stage for the control group.
  - totalDropouts: The total number of dropouts at the stage.
  - logRankStatistic: The log-rank test Z-statistic for the endpoint.
- rawdata (exists if maxNumberOfRawDatasetsPerStage is a positive integer): A data frame for subject-level data for selected replications, containing the following variables:
  - iterationNumber: The iteration number.
  - stageNumber: The stage under consideration.
  - analysisTime: The time for the stage since trial start.
  - subjectId: The subject ID.
  - arrivalTime: The enrollment time for the subject.
  - stratum: The stratum for the subject.
  - treatmentGroup: The treatment group (1 or 2) for the subject.
  - survivalTime1: The underlying survival time for event endpoint 1 for the subject.

- dropoutTime1: The underlying dropout time for event endpoint 1 for the subject.
- timeUnderObservation1: The time under observation since since randomization for event endpoint 1 for the subject.
- event1: Whether the subject experienced event endpoint 1.
- dropoutEvent1: Whether the subject dropped out for endpoint 1.
- survivalTime2: The underlying survival time for event endpoint 2 for the subject.
- dropoutTime2: The underlying dropout time for event endpoint 2 for the subject.
- timeUnderObservation2: The time under observation since since randomization for event endpoint 2 for the subject.
- event2: Whether the subject experienced event endpoint 2.
- dropoutEvent2: Whether the subject dropped out for endpoint 2.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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	A vector that specifies the starting time of piecewise Poisson enrollment 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 starting time of piecewise exponential survival time intervals. 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: A data frame of summary data by iteration and stage:
  - iterationNumber: The iteration number.
  - eventsNotAchieved: Whether the target number of events is not achieved for the iteration.
  - stageNumber: The stage number, covering all stages even if the trial stops at an interim look.
  - analysisTime: The time for the stage since trial start.
  - accruals1: The number of subjects enrolled at the stage for the active treatment 1 group.
  - accruals2: The number of subjects enrolled at the stage for the active treatment 2 group.
  - accruals3: The number of subjects enrolled at the stage for the control group.
  - totalAccruals: The total number of subjects enrolled at the stage.
  - endpoint: The endpoint (1 or 2) under consideration.
  - events1: The number of events at the stage for the active treatment 1 group.
  - events2: The number of events at the stage for the active treatment 2 group.
  - events3: The number of events at the stage for the control group.
  - totalEvents: The total number of events at the stage.
  - dropouts1: The number of dropouts at the stage for the active treatment 1 group.
  - dropouts2: The number of dropouts at the stage for the active treatment 2 group.
  - dropouts3: The number of dropouts at the stage for the control group.
  - totalDropouts: The total number of dropouts at the stage.
  - logRankStatistic13: The log-rank test Z-statistic comparing the active treatment 1 to the control for the endpoint.
  - logRankStatistic23: The log-rank test Z-statistic comparing the active treatment 2 to the control for the endpoint.
  - logRankStatistic12: The log-rank test Z-statistic comparing the active treatment 1 to the active treatment 2 for the endpoint.
- rawdata (exists if maxNumberOfRawDatasetsPerStage is a positive integer): A data frame for subject-level data for selected replications, containing the following variables:
  - iterationNumber: The iteration number.
  - stageNumber: The stage under consideration.
  - analysisTime: The time for the stage since trial start.
  - subjectId: The subject ID.
  - arrivalTime: The enrollment time for the subject.

- stratum: The stratum for the subject.
- treatmentGroup: The treatment group (1, 2, or 3) for the subject.
- survivalTime1: The underlying survival time for event endpoint 1 for the subject.
- dropoutTime1: The underlying dropout time for event endpoint 1 for the subject.
- timeUnderObservation1: The time under observation since since randomization for event endpoint 1 for the subject.
- event1: Whether the subject experienced event endpoint 1.
- dropoutEvent1: Whether the subject dropped out for endpoint 1.
- survivalTime2: The underlying survival time for event endpoint 2 for the subject.
- dropoutTime2: The underlying dropout time for event endpoint 2 for the subject.
- timeUnderObservation2: The time under observation since since randomization for event endpoint 2 for the subject.
- event2: Whether the subject experienced event endpoint 2.
- dropoutEvent2: Whether the subject dropped out for endpoint 2.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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
: A vector that specifies the starting time of piecewise Poisson enrollment 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 starting time of piecewise exponential survival time intervals. 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: A data frame of summary data by iteration and stage:
  - iterationNumber: The iteration number.
  - eventsNotAchieved: Whether the target number of events is not achieved for the iteration.
  - stageNumber: The stage number, covering all stages even if the trial stops at an interim look.
  - analysisTime: The time for the stage since trial start.
  - accruals1: The number of subjects enrolled at the stage for the active treatment 1 group.
  - accruals2: The number of subjects enrolled at the stage for the active treatment 2 group.
  - accruals3: The number of subjects enrolled at the stage for the control group.
  - totalAccruals: The total number of subjects enrolled at the stage.
  - events1: The number of events at the stage for the active treatment 1 group.
  - events2: The number of events at the stage for the active treatment 2 group.
  - events3: The number of events at the stage for the control group.
  - totalEvents: The total number of events at the stage.
  - dropouts1: The number of dropouts at the stage for the active treatment 1 group.
  - dropouts2: The number of dropouts at the stage for the active treatment 2 group.
  - dropouts3: The number of dropouts at the stage for the control group.
  - totalDropouts: The total number of dropouts at the stage.
  - logRankStatistic13: The log-rank test Z-statistic comparing the active treatment 1 to the control.
  - logRankStatistic23: The log-rank test Z-statistic comparing the active treatment 2 to the control.
  - logRankStatistic12: The log-rank test Z-statistic comparing the active treatment 1 to the active treatment 2.
- rawdata (exists if maxNumberOfRawDatasetsPerStage is a positive integer): A data frame for subject-level data for selected replications, containing the following variables:

- iterationNumber: The iteration number.
- stageNumber: The stage under consideration.
- analysisTime: The time for the stage since trial start.
- subjectId: The subject ID.
- arrivalTime: The enrollment time for the subject.
- stratum: The stratum for the subject.
- treatmentGroup: The treatment group (1, 2, or 3) for the subject.
- survivalTime: The underlying survival time for the subject.
- dropoutTime: The underlying dropout time for the subject.
- timeUnderObservation: The time under observation since since randomization for the subject.
- event: Whether the subject experienced the event.
- dropoutEvent: Whether the subject dropped out.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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, number of dropouts, and number of subjects reaching the maximum follow-up 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,
  predictTarget = 2L
)
```

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`	A vector that specifies the starting time of piecewise Poisson enrollment 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 starting time of piecewise exponential survival time intervals. 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.

predictTarget
: The target of prediction. Set `predictTarget = 1` to predict the number of events only. Set `predictTarget = 2` (default) to predict the umber of events and log-rank score statistic mean and variance. Set `predictTarget = 3` to predict the number of events, log-rank score statistic mean and variance, and hazard ratio and variance of log hazard ratio.

Value

A data frame containing the following variables if `predictTarget = 1`:

- `time`: The analysis time since trial start.
- `subjects`: The number of enrolled subjects.
- `nevents`: The total number of events.
- `nevents1`: The number of events in the active treatment group.
- `nevents2`: The number of events in the control group.
- `ndropouts`: The total number of dropouts.
- `ndropouts1`: The number of events for the active treatment group.
- `ndropouts2`: The number of events for the control group.
- `nfmax`: The total number of subjects reaching maximum follow-up.

- nfmax1: The number of subjects reaching maximum follow-up in the active treatment group.
- nfmax2: The number of subjects reaching maximum follow-up in the control group.

If predictTarget = 2, the following variables will also be included:

- uscore: The numerator of the log-rank test statistic.
- vscore: The variance of the log-rank score test statistic.
- logRankZ: The log-rank test statistic on the Z-scale.
- hazardRatioH0: The hazard ratio under the null hypothesis.

Furthermore, if predictTarget = 3, the following additional variables will also be included:

- HR: The average hazard ratio from weighted Cox regression.
- vlogHR: The variance of log hazard ratio.
- zlogHR: The Z-statistic for log hazard ratio.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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 starting time of piecewise exponential survival time intervals. 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.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

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