

Package: lrstat (via r-universe)

October 22, 2024

Type Package

Title Power and Sample Size Calculation for Non-Proportional Hazards and Beyond

Version 0.2.10

Date 2024-10-17

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](https://doi.org/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](https://doi.org/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). The package can also be used for continuous, binary, and count data. For continuous data, it can handle missing data through mixed-model for repeated measures (MMRM). In crossover designs, it can estimate direct treatment effects while accounting for carryover effects. For binary data, it can design Simon's 2-stage, modified toxicity probability-2 (mTPI-2), and Bayesian optimal interval (BOIN) trials. For count data, it can design group sequential trials for negative binomial endpoints with censoring. Additionally, it facilitates group sequential equivalence trials for all supported data types. Moreover, it can design adaptive group sequential trials for changes in sample size, error spending function, number and spacing or future looks. Finally, it offers various options for adjusted p-values, including graphical and gatekeeping procedures.

License GPL (>= 2)

Imports Rcpp (>= 1.0.9), mvtnorm (>= 1.1-3), lpSolve (>= 5.6.1), shiny (>= 1.7.1)

LinkingTo Rcpp

Suggests testthat (>= 3.0.0), dplyr (>= 1.1.4), tidyr (>= 1.3.0),
knitr (>= 1.45), rmarkdown (>= 2.25), survival (>= 3.5-7)

VignetteBuilder knitr

RoxygenNote 7.3.1

Roxygen list(markdown = TRUE)

Encoding UTF-8

NeedsCompilation yes

Depends R (>= 2.10)

LazyData true

Repository <https://kaifenglu.r-universe.dev>

RemoteUrl <https://github.com/kaifenglu/lrstat>

RemoteRef HEAD

RemoteSha 79df66aa5a85951e419f40758e5cb96ece5bbdfd

Contents

lrstat-package	3
accrual	5
adaptDesign	6
caltime	10
errorSpent	12
exitprob	13
fadjpbon	14
fadjpdun	15
fadjpsim	16
fmodmix	17
fseqbon	18
fstdmix	20
fstp2seq	22
fwgtmat	23
getAccrualDurationFromN	23
getBound	24
getDesign	26
getDurationFromNevents	29
getNeventsFromHazardRatio	31
kmest	33
lrpower	35
lrsamplesize	40
lrstat	44
lrstat2e	49
lrstat2e3a	53
lrstat3a	58
lrstat	62
qtpwexp	65

repeatedPValue 66
 updateGraph 67

Index **69**

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

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 *lrrstat* package can also be used for the following tasks:

- design generic group sequential trials.
- design generic group sequential equivalence trials.
- 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.
- design parallel-group trials with the primary endpoint analyzed using mixed-model for repeated measures (MMRM).
- design crossover trials to estimate direct treatment effects while accounting for carryover effects.
- design one-way repeated measures ANOVA trials.
- design two-way ANOVA trials.
- design Simon's 2-stage trials.
- design modified toxicity probability-2 (mTPI-2) trials.
- design Bayesian optimal interval (BOIN) trials.
- design group sequential trials for negative binomial endpoints with censoring.
- design trials using Wilcoxon, Fisher's exact, and McNemar's test.
- calculate Clopper-Pearson confidence interval for single proportions.
- calculate Brookmeyer-Crowley confidence interval for quantiles of censored survival data.
- calculate Miettinen & Nurminen confidence interval for stratified risk difference, risk ratio, odds ratio, rate difference, and rate ratio.
- perform power and sample size calculation for logistic regression.
- perform power and sample size calculation for Cohen's kappa.
- calculate Hedges' g effect size.
- generate random numbers from truncated piecewise exponential distribution.
- perform power and sample size calculations for negative binomial data.

Author(s)

Kaifeng Lu, <kweifengl@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., <code>c(0, 3)</code> breaks the time axis into 2 accrual intervals: <code>[0, 3)</code> and <code>[3, Inf)</code> .
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, <kaifengl@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, obtains 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_,
varianceRatio = 1
)

```

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", or if conditional power calculation is desired.
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 $\text{rep}(-6, k_{\text{Max}}-1)$ 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.
varianceRatio	The ratio of the variance under H0 to the variance under H1.

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$information)

```

caltime

Calendar Times for Target Number of Events

Description

Obtains the calendar times needed to reach the target number of subjects experiencing 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 $[3, \text{Inf})$.
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 $[6, \text{Inf})$. 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, 8),
        accrualIntensity = 26/9*seq(1, 9),
        piecewiseSurvivalTime = c(0, 6),
        lambda1 = c(0.0533, 0.0309),
        lambda2 = c(0.0533, 0.0533),
        gamma1 = -log(1-0.05)/12,
        gamma2 = -log(1-0.05)/12,
        accrualDuration = 22,
        followupTime = 18, fixedFollowup = FALSE)
```

errorSpent

Error Spending

Description

Obtains the error spent at given spending times for the specified error spending function.

Usage

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

Arguments

t	A vector of spending times, typically equal to information fractions.
error	The total error to spend.
sf	The 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	The 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, <kweifenglu@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 $c(\text{rep}(-6.0, k_{\text{Max}}-1), b[k_{\text{Max}}])$ if left unspecified, where $k_{\text{Max}} = \text{length}(b)$. |
| 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, $\text{theta} = -\log(\text{HR})$. 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 $\text{phi} \cdot (1-\text{phi}) \cdot D$, 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 $\text{seq}(1, k_{\text{Max}})$ 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, <kweifenglu@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	<i>Adjusted p-Values for Dunnett-Based Graphical Approaches</i>
----------	---

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 = TRUE  
)
```

Arguments

<code>p</code>	The raw p-values for elementary hypotheses.
<code>family</code>	The matrix of family indicators for the hypotheses.
<code>serial</code>	The matrix of serial rejection set for the hypotheses.
<code>parallel</code>	The matrix of parallel rejection set for the hypotheses.
<code>gamma</code>	The truncation parameters for each family. The truncation parameter for the last family is automatically set to 1.
<code>test</code>	The component multiple testing procedure. Options include "holm", "hochberg", or "hommel". Defaults to "hommel".
<code>exhaust</code>	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, test = "hommel", exhaust = TRUE)
```

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 & Tsai's 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 = TRUE
)
```

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. The truncation parameter for the last family is automatically set to 1.
test	The component multiple testing procedure. Options include "holm", "hochberg", or "hommel". Defaults to "hommel".
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, test = "hommel",
        exhaust = FALSE)
```

`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

<code>p</code>	The raw p-values for elementary hypotheses.
<code>gamma</code>	The truncation parameters for each family. The truncation parameter for the last family is automatically set to 1.
<code>test</code>	The component multiple testing procedure. It is either "Holm" or "Hochberg", and it defaults to "Hochberg".
<code>retest</code>	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)
```

`fwgmat`*Weight Matrix for All Intersection Hypotheses*

Description

Obtains the weight matrix for all intersection hypotheses.

Usage

```
fwgmat(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 = fwgmat(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))
```

getBound

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 & Tsatis 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")
```

getDesign

*Power and Sample Size for a Generic 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_,
  varianceRatio = 1
)
```

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 & Tsatis 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. The futility bounds are non-binding for the calculation of critical values.
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.
varianceRatio	The ratio of the variance under H0 to the variance under H1.

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.
 - attainedAlpha: The attained significance level, which is different from the overall significance level in the presence of futility stopping.
 - kMax: The number of stages.
 - theta: The parameter value.

- information: The maximum information.
- expectedInformationH1: The expected information under H1.
- expectedInformationH0: The expected information under H0.
- drift: The drift parameter, equal to $\theta \cdot \sqrt{\text{information}}$.
- 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.
 - rejectPerStageH0: The probability for efficacy stopping under H0.
 - futilityPerStageH0: The probability for futility stopping under H0.
 - cumulativeRejectionH0: The cumulative probability for efficacy stopping under H0.
 - cumulativeFutilityH0: The cumulative probability for futility stopping under H0.
- 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.
 - varianceRatio: The ratio of the variance under H0 to the variance under H1.

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
(design1 <- 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
(design2 <- 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 = NA_real_,
  fixedFollowup = 0L,
  npoints = 23L,
  interval = as.numeric(c(0.001, 240))
)
```

Arguments

nevents The target number of events.

<code>allocationRatioPlanned</code>	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
<code>accrualTime</code>	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 $[3, \text{Inf})$.
<code>accrualIntensity</code>	A vector of accrual intensities. One for each accrual time interval.
<code>piecewiseSurvivalTime</code>	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 $[6, \text{Inf})$. Defaults to 0 for exponential distribution.
<code>stratumFraction</code>	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
<code>lambda1</code>	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
<code>lambda2</code>	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
<code>gamma1</code>	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.
<code>gamma2</code>	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.
<code>followupTime</code>	Follow-up time for the last enrolled subjects. Must be provided for fixed follow-up design.
<code>fixedFollowup</code>	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
<code>npoints</code>	The number of accrual duration time points. Defaults to 23.
<code>interval</code>	The interval to search for the solution of <code>accrualDuration</code> . 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.
- `followupTime`: The follow-up time for the last enrolled subject.
- `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),
  lambda1 = c(0.0533, 0.0309),
  lambda2 = c(0.0533, 0.0533),
  gamma1 = -log(1-0.05)/12,
  gamma2 = -log(1-0.05)/12,
  fixedFollowup = FALSE)
```

```
getNeventsFromHazardRatio
```

Required Number of Events Given Hazard Ratio

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 = NA_real_,
  allocationRatioPlanned = 1,
  rounding = 1L
)
```

Arguments

<code>beta</code>	Type II error. Defaults to 0.2.
<code>kMax</code>	The maximum number of stages.
<code>informationRates</code>	The information rates in terms of number of events. Fixed prior to the trial. Defaults to $(1:kMax) / kMax$ if left unspecified.
<code>efficacyStopping</code>	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
<code>futilityStopping</code>	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
<code>criticalValues</code>	Upper boundaries on the z-test statistic scale for stopping for efficacy.
<code>alpha</code>	The significance level. Defaults to 0.025.
<code>typeAlphaSpending</code>	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsatis 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".
<code>parameterAlphaSpending</code>	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
<code>userAlphaSpending</code>	The user defined alpha spending. Cumulative alpha spent up to each stage.
<code>futilityBounds</code>	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. The futility bounds are non-binding for the calculation of critical values.
<code>typeBetaSpending</code>	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".
<code>parameterBetaSpending</code>	The parameter value for the beta spending. Corresponds to rho for "sfKD", and gamma for "sfHSD".
<code>userBetaSpending</code>	The user defined beta spending. Cumulative beta spent up to each stage.
<code>spendingTime</code>	A vector of length $kMax$ for the error spending time at each analysis. Defaults to missing, in which case, it is the same as <code>informationRates</code> .
<code>hazardRatioH0</code>	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 = "sf0F",  
  typeBetaSpending = "sfP",  
  hazardRatio = 0.673)
```

kmest

Kaplan-Meier Estimates of Survival Curve

Description

Obtains the Kaplan-Meier estimates of the survival curve.

Usage

```
kmest(  
  data,  
  rep = "",  
  stratum = "",  
  time = "time",  
  event = "event",  
  conftype = "log-log",  
  conflev = 0.95  
)
```

Arguments

data	The input data frame that contains the following variables: <ul style="list-style-type: none"> • rep: The replication for by-group processing. • stratum: The stratum. • time: The possibly right-censored survival time. • event: The event indicator.
rep	The name(s) of the replication variable(s) in the input data.
stratum	The name(s) of the stratum variable(s) in the input data.
time	The name of the time variable in the input data.
event	The name of the event variable in the input data.
conftype	The type of the confidence interval. One of "none", "plain", "log", "log-log" (the default), or "arcsin". The arcsin option bases the intervals on $\text{asin}(\sqrt{\text{survival}})$.
conflev	The level of the two-sided confidence interval for the survival probabilities. Defaults to 0.95.

Value

A data frame with the following variables:

- size: The number of subjects in the stratum.
- time: The event time.
- nrisk: The number of subjects at risk.
- nevent: The number of subjects having the event.
- survival: The Kaplan-Meier estimate of the survival probability.
- stderr: The standard error of the estimated survival probability based on the Greenwood formula.
- lower: The lower bound of confidence interval if requested.
- upper: The upper bound of confidence interval if requested.
- conflev: The level of confidence interval if requested.
- conftype: The type of confidence interval if requested.
- stratum: The stratum.
- rep: The replication.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

Examples

```
kmest(data = aml, stratum = "x", time = "time", event = "status")
```

Irpower	<i>Log-Rank Test Power</i>
---------	----------------------------

Description

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

Usage

```
Irpower(
  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 = 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,
  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. The futility bounds are non-binding for the calculation of critical values.
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.

<code>piecewiseSurvivalTime</code>	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 $[6, \text{Inf})$. Defaults to 0 for exponential distribution.
<code>stratumFraction</code>	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
<code>lambda1</code>	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
<code>lambda2</code>	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
<code>gamma1</code>	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.
<code>gamma2</code>	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.
<code>accrualDuration</code>	Duration of the enrollment period.
<code>followupTime</code>	Follow-up time for the last enrolled subject.
<code>fixedFollowup</code>	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
<code>rho1</code>	The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
<code>rho2</code>	The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
<code>numSubintervals</code>	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.
<code>estimateHazardRatio</code>	Whether to estimate the hazard ratio from weighted Cox regression model and report the stopping boundaries on the hazard ratio scale.
<code>typeOfComputation</code>	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.
<code>spendingTime</code>	A vector of length <code>kMax</code> for the error spending time at each analysis. Defaults to missing, in which case, it is the same as <code>informationRates</code> .
<code>studyDuration</code>	Study duration for fixed follow-up design. Defaults to missing, which is to be replaced with the sum of <code>accrualDuration</code> and <code>followupTime</code> . If provided, the value is allowed to be less than the sum of <code>accrualDuration</code> and <code>followupTime</code> .

Value

An S3 class lpower 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.
 - information: The maximum information.
 - expectedNumberOfEvents: The expected number of events.
 - expectedNumberOfDropouts: The expected number of dropouts.
 - expectedNumberOfSubjects: The expected number of subjects.
 - expectedStudyDuration: The expected study duration.
 - expectedInformation: The expected information.
 - accrualDuration: The accrual duration.
 - followupTime: The follow-up time.
 - 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.
 - kMax: The number of stages.
 - hazardRatioH0: The hazard ratio under the null hypothesis.
 - 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 if estimateHazardRatio.
 - futilityHR: The futility boundaries on the hazard ratio scale if estimateHazardRatio.
 - 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, allocationRatioPlanned, accrualTime, accrualIntensity, piecewiseSurvivalTime, stratumFraction, lambda1, lambda2, gamma1, gamma2, estimateHazardRatio, 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.
```

```
Irpower(kMax = 2, informationRates = c(0.8, 1),
        alpha = 0.025, typeAlphaSpending = "sfOF",
        allocationRatioPlanned = 1, accrualTime = seq(0, 8),
        accrualIntensity = 26/9*seq(1, 9),
        piecewiseSurvivalTime = c(0, 6),
        lambda1 = c(0.0533, 0.0309),
        lambda2 = c(0.0533, 0.0533),
        gamma1 = -log(1-0.05)/12,
        gamma2 = -log(1-0.05)/12, accrualDuration = 22,
        followupTime = 18, fixedFollowup = FALSE)
```

Irsamplesize	<i>Log-Rank Test Sample Size</i>
--------------	----------------------------------

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 time, and relative accrual rates in a two-group survival design.

Usage

```
Irsamplesize(
  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 = 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,
  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. The futility bounds are non-binding for the calculation of critical values.
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 $[3, \text{Inf})$.
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 $[6, \text{Inf})$. 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, followupTime, 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 time
```

```
Irsamplesize(beta = 0.2, kMax = 2,
              informationRates = c(0.8, 1),
              alpha = 0.025, typeAlphaSpending = "sfOF",
              accrualTime = seq(0, 8),
              accrualIntensity = 26/9*seq(1, 9),
              piecewiseSurvivalTime = c(0, 6),
              lambda1 = c(0.0533, 0.0309),
              lambda2 = c(0.0533, 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 time given power and accrual duration
```

```
Irsamplesize(beta = 0.2, kMax = 2,
              informationRates = c(0.8, 1),
              alpha = 0.025, typeAlphaSpending = "sfOF",
              accrualTime = seq(0, 8),
              accrualIntensity = 26/9*seq(1, 9),
```

```

    piecewiseSurvivalTime = c(0, 6),
    lambda1 = c(0.0533, 0.0309),
    lambda2 = c(0.0533, 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 time, and relative accrual intensity

lrsamplesize(beta = 0.2, kMax = 2,
              informationRates = c(0.8, 1),
              alpha = 0.025, typeAlphaSpending = "sfOF",
              accrualTime = seq(0, 8),
              accrualIntensity = 26/9*seq(1, 9),
              piecewiseSurvivalTime = c(0, 6),
              lambda1 = c(0.0533, 0.0309),
              lambda2 = c(0.0533, 0.0533),
              gamma1 = -log(1-0.05)/12,
              gamma2 = -log(1-0.05)/12,
              accrualDuration = 22,
              followupTime = 18, fixedFollowup = FALSE)

```

Irsim

Log-Rank Test Simulation

Description

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

Usage

```

lrsim(
  kMax = 1L,
  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

<code>kMax</code>	The maximum number of stages.
<code>informationRates</code>	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 <code>plannedEvents / plannedEvents[kMax]</code> when <code>plannedEvents</code> is provided and to <code>plannedTime / plannedTime[kMax]</code> otherwise.
<code>criticalValues</code>	Upper boundaries on the z-test statistic scale for stopping for efficacy.
<code>futilityBounds</code>	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., <code>kMax-1</code> . Defaults to <code>rep(-6, kMax-1)</code> if left unspecified. The futility bounds are non-binding for the calculation of critical values.
<code>hazardRatioH0</code>	Hazard ratio under the null hypothesis for the active treatment versus control. Defaults to 1 for superiority test.
<code>allocation1</code>	Number of subjects in the active treatment group in a randomization block. Defaults to 1 for equal randomization.
<code>allocation2</code>	Number of subjects in the control group in a randomization block. Defaults to 1 for equal randomization.
<code>accrualTime</code>	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., <code>c(0, 3)</code> breaks the time axis into 2 accrual intervals: <code>[0, 3)</code> and <code>[3, Inf)</code> .
<code>accrualIntensity</code>	A vector of accrual intensities. One for each accrual time interval.
<code>piecewiseSurvivalTime</code>	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., <code>c(0, 6)</code> breaks the time axis into 2 event intervals: <code>[0, 6)</code> and <code>[6, Inf)</code> . Defaults to 0 for exponential distribution.
<code>stratumFraction</code>	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
<code>lambda1</code>	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.

<code>lambda2</code>	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
<code>gamma1</code>	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.
<code>gamma2</code>	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.
<code>accrualDuration</code>	Duration of the enrollment period.
<code>followupTime</code>	Follow-up time for the last enrolled subject.
<code>fixedFollowup</code>	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
<code>rho1</code>	The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
<code>rho2</code>	The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
<code>plannedEvents</code>	The planned cumulative total number of events at each stage.
<code>plannedTime</code>	The calendar times for the analyses. To use calendar time to plan the analyses, <code>plannedEvents</code> should be missing.
<code>maxNumberOfIterations</code>	The number of simulation iterations. Defaults to 1000.
<code>maxNumberOfRawDatasetsPerStage</code>	The number of raw datasets per stage to extract.
<code>seed</code>	The seed to reproduce the simulation results. The seed from the environment will be used if left unspecified.

Value

An S3 class `Irsim` 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 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)
```

Irsim2e

Log-Rank Test Simulation for Two Endpoints

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 $[3, \text{Inf})$.
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 $[6, \text{Inf})$. 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).

<code>gamma1e2</code>	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).
<code>gamma2e2</code>	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).
<code>accrualDuration</code>	Duration of the enrollment period.
<code>followupTime</code>	Follow-up time for the last enrolled subject.
<code>fixedFollowup</code>	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
<code>rho1</code>	The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
<code>rho2</code>	The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
<code>plannedEvents</code>	The planned cumulative total number of PFS events at Look 1 to Look <code>kMaxe1</code> and the planned cumulative total number of OS events at Look <code>kMaxe1+1</code> to Look <code>kMax</code> .
<code>plannedTime</code>	The calendar times for the analyses. To use calendar time to plan the analyses, <code>plannedEvents</code> should be missing.
<code>maxNumberOfIterations</code>	The number of simulation iterations. Defaults to 1000.
<code>maxNumberOfRawDatasetsPerStage</code>	The number of raw datasets per stage to extract.
<code>seed</code>	The seed to reproduce the simulation results. The seed from the environment 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 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 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)

```

Irsim2e3a

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 $k_{\text{Max}1}$ 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

```

Irsim2e3a(
  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 $[3, \text{Inf})$.
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 $[6, \text{Inf})$. 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).

<code>gamma3e2</code>	The hazard rate for exponential dropout. A vector of hazard rates for piece-wise 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).
<code>accrualDuration</code>	Duration of the enrollment period.
<code>followupTime</code>	Follow-up time for the last enrolled subject.
<code>fixedFollowup</code>	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
<code>rho1</code>	The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
<code>rho2</code>	The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
<code>plannedEvents</code>	The planned cumulative total number of PFS events at Look 1 to Look <code>kMaxe1</code> for Arms A and C combined and the planned cumulative total number of OS events at Look <code>kMaxe1+1</code> to Look <code>kMax</code> for Arms A and C combined.
<code>plannedTime</code>	The calendar times for the analyses. To use calendar time to plan the analyses, <code>plannedEvents</code> should be missing.
<code>maxNumberOfIterations</code>	The number of simulation iterations. Defaults to 1000.
<code>maxNumberOfRawDatasetsPerStage</code>	The number of raw datasets per stage to extract.
<code>seed</code>	The seed to reproduce the simulation results. The seed from the environment 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 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 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 = 500,
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 $[3, \text{Inf})$.
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 $[6, \text{Inf})$. 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.
seed	The seed to reproduce the simulation results. The seed from the environment 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 randomization for the subject.
 - event: Whether the subject experienced the event.
 - dropoutEvent: Whether the subject dropped out.

Author(s)

Kaifeng Lu, <kaifengl@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)

```

Irstat

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

```

Irstat(
  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 $[3, \text{Inf})$.
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 $[6, \text{Inf})$. 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 <code>predictTarget = 1</code> to predict the number of events only. Set <code>predictTarget = 2</code> (default) to predict the number of events and log-rank score statistic mean and variance. Set <code>predictTarget = 3</code> 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 dropouts in the active treatment group.
- `ndropouts2`: The number of dropouts in 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, 8),
       accrualIntensity = 26/9*seq(1, 9),
       piecewiseSurvivalTime = c(0, 6),
       lambda1 = c(0.0533, 0.0309),
       lambda2 = c(0.0533, 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 truncated piecewise exponential distribution.

Usage

```
qtpwexp(
  p,
  piecewiseSurvivalTime = 0,
  lambda = 0.0578,
  lowerBound = 0,
  lower.tail = TRUE,
  log.p = FALSE
)
```

Arguments

p	The vector of probabilities.
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.
lower.tail	Logical; if TRUE (default), probabilities are $P(X \leq x)$, otherwise, $P(X > x)$.
log.p	Logical; if TRUE, probabilities p are given as log(p).

Value

The quantile q such that $P(X > q \mid X > \text{lowerBound}) = 1 - p$.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

Examples

```
qtpwexp(p = c(0.205, 0.317), 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 & Tsatis 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, 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

accrual, [5](#)
adaptDesign, [6](#)

caltime, [10](#)

errorSpent, [12](#)
exitprob, [13](#)

fadjpbon, [14](#)
fadjpdun, [15](#)
fadjpsim, [16](#)
fmodmix, [17](#)
fseqbon, [18](#)
fstdmix, [20](#)
fstp2seq, [22](#)
fwgtmat, [23](#)

getAccrualDurationFromN, [23](#)
getBound, [24](#)
getDesign, [9](#), [26](#)
getDurationFromNevents, [29](#)
getNeventsFromHazardRatio, [31](#)

kmest, [33](#)

lrpower, [35](#), [43](#)
lrsamplesize, [40](#)
lrsim, [44](#)
lrsim2e, [49](#)
lrsim2e3a, [53](#)
lrsim3a, [58](#)
lrstat, [62](#)
lrstat-package, [3](#)

qtpwexp, [65](#)

repeatedPValue, [66](#)

updateGraph, [67](#)