Package ‘lrstat’

July 22, 2025

Type Package

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

Version 0.2.15

Date 2025-07-14

Description Performs power and sample size calculation for non-proportional hazards model using the Fleming-Harrington family of weighted log-rank tests. The sequentially calculated log-rank test score statistics are assumed to have independent increments as characterized in Anastasios A. Tsiatis (1982) <doi:10.1080/01621459.1982.10477898>. The mean and variance of log-rank test score statistics are calculated based on Kaifeng Lu (2021) <doi:10.1002/pst.2069>. The boundary crossing probabilities are calculated using the recursive integration algorithm described in Christopher Jennison and Bruce W. Turnbull (2000, ISBN:0849303168). 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 7.3.2

Encoding UTF-8

NeedsCompilation yes

Depends R (>= 2.10)

LazyData true

URL https://github.com/kaifenglu/lrstat

BugReports https://github.com/kaifenglu/lrstat/issues

Config/testthat/edition 3

Author Kaifeng Lu [aut, cre] (ORCID: <https://orcid.org/0000-0002-6160-7119>)

Maintainer Kaifeng Lu <kaifenglu@gmail.com>

Repository CRAN

Date/Publication 2025-07-14 06:20:02 UTC

Contents

`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 lrstat 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, `<kaifenglu@gmail.com>`

References

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

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

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

See Also

rpact, gsDesign

Examples

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

`accrual` *Number of Enrolled Subjects*

Description

Obtains the number of subjects enrolled by given calendar times.

Usage

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

Arguments

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

Value

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

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

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

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


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

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

`adaptDesign` *Adaptive Design at an Interim Look*

Description

Calculates the conditional power for specified incremental information, given the interim results, parameter value, data-dependent changes in the error spending function, and the number and spacing of interim looks. Conversely, calculates the incremental information required to attain a specified conditional power, given the interim results, parameter value, 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 must 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. Represents the cumulative alpha spent up to each stage.
`futilityBounds`	The lower boundaries on the z-test statistic scale for futility stopping for the primary trial. Defaults to `rep(-6, kMax-1)` if left unspecified.
`typeBetaSpending`	The type of beta spending for the primary trial. One of the following: "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, and "none" for no early futility stopping. Defaults to "none".
`parameterBetaSpending`	The parameter value of beta spending for the primary trial. Corresponds to rho for "sfKD", and gamma for "sfHSD".
`spendingTime`	The error spending time of the primary trial. Defaults to missing, in which case it is assumed to be 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. Represents the 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 assumed to be 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 = L, zL = 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 = L, zL = 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)
```

`aml` *Acute myelogenous leukemia survival data from the survival package*

Description

Survival in patients with acute myelogenous leukemia.

`time` Survival or censoring time

`status` censoring status

`x` maintenance chemotherapy given or not

Usage

`aml`

Format

An object of class `data.frame` with 23 rows and 3 columns.

`binary_tte_sim` *Simulation for a Binary and a Time-to-Event Endpoint in Group Sequential Trials*

Description

Performs simulation for two-endpoint two-arm group sequential trials.

- Endpoint 1: Binary endpoint, analyzed using the Mantel-Haenszel test for risk difference.
- Endpoint 2: Time-to-event endpoint, analyzed using the log-rank test for treatment effect.

The analysis times for the binary endpoint are based on calendar times, while the time-to-event analyses are triggered by reaching the pre-specified number of events. The binary endpoint is assessed at the first post-treatment follow-up visit (PTFU1).

Usage

```
binary_tte_sim(
  kMax1 = 1L,
  kMax2 = 1L,
  riskDiffH0 = 0,
  hazardRatioH0 = 1,
  allocation1 = 1L,
  allocation2 = 1L,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  globalOddsRatio = 1,
  pi1 = NA_real_,
  pi2 = NA_real_,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  delta1 = 0L,
  delta2 = 0L,
  upper1 = NA_real_,
  upper2 = NA_real_,
  accrualDuration = NA_real_,
  plannedTime = NA_real_,
  plannedEvents = NA_integer_,
  maxNumberOfIterations = 1000L,
  maxNumberOfRawDatasetsPerStage = 0L,
  seed = NA_integer_
)
```

Arguments

`kMax1`	Number of stages for the binary endpoint.
`kMax2`	Number of stages for the time-to-event endpoint.
`riskDiffH0`	Risk difference under the null hypothesis for the binary endpoint.
`hazardRatioH0`	Hazard ratio under the null hypothesis for the time-to-event endpoint.
`allocation1`	Number of subjects in the treatment group in a randomization block. Defaults to 1 for equal randomization.
`allocation2`	Number of subjects in the control group in a randomization block. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`globalOddsRatio`	Global odds ratio of the Plackett copula linking the two endpoints.
`pi1`	Response probabilities by stratum for the treatment group for the binary endpoint.
`pi2`	Response probabilities by stratum for the control group for the binary endpoint.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the treatment group for the time-to-event endpoint.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group for the time-to-event endpoint.
`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 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.
`delta1`	The hazard rate for exponential treatment discontinuation, a vector of hazard rates for piecewise exponential treatment discontinuation applicable for all strata, or a vector of hazard rates for treatment discontinuation in each analysis time interval by stratum for the treatment group for the binary endpoint.
`delta2`	The hazard rate for exponential treatment discontinuation, a vector of hazard rates for piecewise exponential treatment discontinuation applicable for all strata, or a vector of hazard rates for treatment discontinuation in each analysis time interval by stratum for the control group for the binary endpoint.

`upper1`	Maximim protocol-specified treatment duration for the treatment group.
`upper2`	Maximum protocol-specified treatment duration for the control group.
`accrualDuration`	Duration of the enrollment period.
`plannedTime`	Calendar times for the analyses of the binary endpoint.
`plannedEvents`	Target cumulative number of events for the time-to-event analyses.
`maxNumberOfIterations`	Number of simulation iterations to perform.
`maxNumberOfRawDatasetsPerStage`	Number of subject-level datasets to retain per stage. Set to 0 to skip raw data saving.
`seed`	Random seed for reproducibility. If not specified, the current R environment seed is used.

Details

We consider dual primary endpoints with endpoint 1 being a binary endpoint and endpoint 2 being a time-to-event endpoint. The analyses of endpoint 1 will be based on calendar times, while the analyses of endpoint 2 will be based on the number of events. Therefore, the analyses of the two endpoints are not at the same time points. The correlation between the two endpoints is characterized by the global odds ratio of the Plackett copula. In addition, the time-to-event endpoint will render the binary endpoint as a non-responder, and so does the dropout. In addition, the treatment discontinuation will impact the number of available subjects for analysis. The administrative censoring will exclude subjects from the analysis of the binary endpoint.

Value

A list with 4 components:

- `sumdataBIN`: A data frame of summary data by iteration and stage for the binary endpoint:
  - `iterationNumber`: The iteration number.
  - `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.
  - `source1`: The total number of subjects with response status determined by the underlying latent response variable.
  - `source2`: The total number of subjects with response status (non-responder) determined by experiencing the event for the time-to-event endpoint.
  - `source3`: The total number of subjects with response status (non-responder) determined by dropping out prior to the PTFU1 visit.
  - `n1`: The number of subjects included in the analysis of the binary endpoint for the treatment group.

  - `n2`: The number of subjects included in the analysis of the binary endpoint for the control group.
  - `n`: The total number of subjects included in the analysis of the binary endpoint at the stage.
  - `y1`: The number of responders for the binary endpoint in the treatment group.
  - `y2`: The number of responders for the binary endpoint in the control group.
  - `y`: The total number of responders for the binary endpoint at the stage.
  - `riskDiff`: The estimated risk difference for the binary endpoint.
  - `seRiskDiff`: The standard error for risk difference based on the Sato approximation.
  - `mnStatistic`: The Mantel-Haenszel test Z-statistic for the binary endpoint.
- `sumdataTTE`: A data frame of summary data by iteration and stage for the time-to-event endpoint:
  - `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.
  - `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 time-to-event endpoint.
- `rawdataBIN` (exists if `maxNumberOfRawDatasetsPerStage` is a positive integer): A data frame for subject-level data for the binary endpoint 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.
  - `survivalTime`: The underlying survival time for the time-to-event endpoint for the subject.
  - `dropoutTime`: The underlying dropout time for the time-to-event endpoint for the subject.
  - `ptfu1Time`:The underlying assessment time for the binary endpoint for the subject.

  - `timeUnderObservation`: The time under observation since randomization for the binary endpoint for the subject.
  - `responder`: Whether the subject is a responder for the binary endpoint.
  - `source`: The source of the determination of responder status for the binary endpoint: = 1 based on the underlying latent response variable, = 2 based on the occurrence of the time-to-event endpoint before the assessment time of the binary endpoint (imputed as a non-responder), = 3 based on the dropout before the assessment time of the binary endpoint (imputed as a non-responder), = 4 excluded from analysis due to administrative censoring.
- `rawdataTTE` (exists if `maxNumberOfRawDatasetsPerStage` is a positive integer): A data frame for subject-level data for the time-to-event endpoint 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.
  - `survivalTime`: The underlying survival time for the time-to-event endpoint for the subject.
  - `dropoutTime`: The underlying dropout time for the time-to-event endpoint for the subject.
  - `timeUnderObservation`: The time under observation since randomization for the time-to-event endpoint for the subject.
  - `event`: Whether the subject experienced the event for the time-to-event endpoint.
  - `dropoutEvent`: Whether the subject dropped out for the time-to-event endpoint.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
tcut = c(0, 12, 36, 48)
surv = c(1, 0.95, 0.82, 0.74)
lambda2 = (log(surv[1:3]) - log(surv[2:4]))/(tcut[2:4] - tcut[1:3])

sim1 = binary_tte_sim(
  kMax1 = 1,
  kMax2 = 2,
  accrualTime = 0:8,
  accrualIntensity = c(((1:8) - 0.5)/8, 1)*40,
  piecewiseSurvivalTime = c(0,12,36),
  globalOddsRatio = 1,
  pi1 = 0.80,
  pi2 = 0.65,
  lambda1 = 0.65*lambda2,
  lambda2 = lambda2,
```

```
gamma1 = -log(1-0.04)/12,
gamma2 = -log(1-0.04)/12,
delta1 = -log(1-0.02)/12,
delta2 = -log(1-0.02)/12,
upper1 = 15*28/30.4,
upper2 = 12*28/30.4,
accrualDuration = 20,
plannedTime = 20 + 15*28/30.4,
plannedEvents = c(130, 173),
maxNumberOfIterations = 1000,
maxNumberOfRawDatasetsPerStage = 1,
seed = 314159)
```

`BOINTable` *BOIN Decision Table for Dose-Finding Trials*

Description

Generates the decision table for the Bayesian Optimal Interval (BOIN) design, a widely used approach for dose-escalation trials that guides dose-finding decisions based on observed toxicity rates.

Usage

```
BOINTable(
  nMax = NA_integer_,
  pT = 0.3,
  phi1 = 0.6 * pT,
  phi2 = 1.4 * pT,
  a = 1,
  b = 1,
  pExcessTox = 0.95
)
```

Arguments

`nMax`	The maximum number of subjects allowed in a dose cohort.
`pT`	The target toxicity probability. Defaults to 0.3.
`phi1`	The lower equivalence limit for the target toxicity probability.
`phi2`	The upper equivalence limit for the target toxicity probability.
`a`	The prior toxicity shape parameter for the Beta prior.
`b`	The prior non-toxicity shape parameter for the Beta prior.
`pExcessTox`	The threshold for excessive toxicity. If the posterior probability that the true toxicity rate exceeds `pT` is greater than `pExcessTox`, the current and all higher doses will be excluded from further use to protect future participants. Defaults to 0.95.

Value

An S3 class `BOINTable` object with the following components:

- `settings`: The input settings data frame with the following variables:
  - `nMax`: The maximum number of subjects in a dose cohort.
  - `pT`: The target toxicity probability.
  - `phi1`: The lower equivalence limit for target toxicity probability.
  - `phi2`: The upper equivalence limit for target toxicity probability.
  - `lambda1`: The lower decision boundary for observed toxicity probability.
  - `lambda2`: The upper decision boundary for observed toxicity probability.
  - `a`: The prior toxicity parameter for the beta prior.
  - `b`: The prior non-toxicity parameter for the beta prior.
  - `pExcessTox`: The threshold for excessive toxicity.
- `decisionDataFrame`: A data frame listing dose-finding decisions for each combination of sample size (`n`) and number of observed toxicities (`y`):
  - `n`: Cohort size.
  - `y`: Number of observed toxicities.
  - `decision`: Recommended action: escalate, de-escalate, or stay at the current dose.
- `decisionMatrix`: A matrix version of the decision table showing the recommended action based on the number of toxicities for each possible cohort size.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

Liu, S., & Yuan, Y. (2015). Bayesian optimal interval designs for phase I clinical trials. Journal of the Royal Statistical Society: Series C (Applied Statistics), 64(3), 507-523.

Examples

```
BOINTable(nMax = 18, pT = 0.3, phi = 0.6*0.3, phi2 = 1.4*0.3)
```

`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, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.

`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.

Value

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

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

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

caltime(nevents = c(24, 80), allocationRatioPlanned = 1,
        accrualTime = seq(0, 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)
```

`ClopperPearsonCI` *Clopper-Pearson Confidence Interval for One-Sample Proportion*

Description

Obtains the Clopper-Pearson exact confidence interval for a one-sample proportion.

Usage

```
ClopperPearsonCI(n, y, cilevel = 0.95)
```

Arguments

`n`	The sample size.
`y`	The number of responses.
`cilevel`	The confidence interval level.

Value

A data frame with the following variables:

- `n`: The sample size.
- `y`: The number of responses.
- `phat`: The observed proportion of responses.
- `lower`: The lower limit of the confidence interval.
- `upper`: The upper limit of the confidence interval.
- `cilevel`: The confidence interval level.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

Clopper, C. J., & Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika, 26(4), 404-413.

Examples

```
ClopperPearsonCI(20, 3)
```

`covrmst` *Covariance Between Restricted Mean Survival Times*

Description

Obtains the covariance between restricted mean survival times at two different time points.

Usage

```
covrmst(
  t2 = NA_real_,
  tau1 = NA_real_,
  tau2 = NA_real_,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = NA_real_,
  maxFollowupTime = NA_real_
)
```

Arguments

`t2`	The calendar time for analysis 2.
`tau1`	The milestone time for analysis 1.
`tau2`	The milestone time for analysis 2.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`lambda1`	A vector of hazard rates for the event for the active treatment group. One for each analysis time interval.
`lambda2`	A vector of hazard rates for the event for the control group. One for each analysis time interval.
`gamma1`	The hazard rate for exponential dropout, or a vector of hazard rates for piecewise exponential dropout for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, or a vector of hazard rates for piecewise exponential dropout for the control group.
`accrualDuration`	Duration of the enrollment period.
`maxFollowupTime`	Follow-up time for the first enrolled subject. For fixed follow-up, `maxFollowupTime = minFollowupTime`. For variable follow-up, `maxFollowupTime = accrualDuration + minFollowupTime`.

Value

The covariance between the restricted mean survival times for each treatment group.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
covrmst(t2 = 25, tau1 = 16, tau2 = 18, allocationRatioPlanned = 1,
        accrualTime = c(0, 3), accrualIntensity = c(10, 20),
        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 = 12, maxFollowupTime = 30)
```

`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".

Details

This function implements a variety of error spending functions commonly used in group sequential designs, assuming one-sided hypothesis testing.

O'Brien-Fleming-Type Spending Function

This spending function allocates very little alpha early on and more alpha later in the trial. It is defined as:

$$\alpha(t) = 2 - 2\Phi\left(\frac{z_{\alpha/2}}{\sqrt{t}}\right),$$

where Φ is the standard normal cumulative distribution function, $z_{\alpha/2}$ is the critical value from the standard normal distribution, and $t \in [0, 1]$ denotes the information fraction.

Pocock-Type Spending Function

This function spends alpha more evenly throughout the study:

$$\alpha(t) = \alpha \log(1 + (e - 1)t),$$

where e is Euler's number (approximately 2.718).

Kim and DeMets Power-Type Spending Function

This family of spending functions is defined as:

$$\alpha(t) = \alpha t^{\rho}, \quad \rho > 0.$$

- When $\rho = 1$, the function mimics Pocock-type boundaries.

- When $\rho = 3$, it approximates O'Brien-Fleming-type boundaries.

Hwang, Shih, and DeCani Spending Function

This flexible family of functions is given by:

$$\alpha(t) = \begin{cases} \alpha \frac{1-e^{-\gamma t}}{1-e^{-\gamma}}, & \text{if } \gamma \neq 0 \\ \alpha t, & \text{if } \gamma = 0. \end{cases}$$

- When $\gamma = -4$, the spending function resembles O'Brien-Fleming boundaries.
- When $\gamma = 1$, it resembles Pocock boundaries.

Value

A vector of errors spent up to the interim look.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
errorSpent(t = 0.5, error = 0.025, sf = "sfOF")

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

`exitprob` *Stagewise Exit Probabilities*

Description

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

Usage

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

Arguments

b	Upper boundaries on the z-test statistic scale.
a	Lower boundaries on the z-test statistic scale. Defaults to `c(rep(-6.0, kMax-1), b[kMax])` if left unspecified, where `kMax = length(b)`.
theta	Stagewise parameter of interest, e.g., `-U/V` for weighted log-rank test, where `U` is the mean and `V` is the variance of the weighted log-rank test score statistic at each stage. For proportional hazards and conventional log-rank test, use the scalar input, `theta = -log(HR)`. Defaults to 0 corresponding to the null hypothesis.

`I`	Stagewise cumulative information, e.g., `V`, the variance of the weighted log-rank test score statistic at each stage. For conventional log-rank test, information can be approximated by `phi*(1-phi)*D`, where `phi` is the probability of being allocated to the active arm, and `D` is the total number of events at each stage. Defaults to `seq(1, kMax)` if left unspecified.

Value

A list of stagewise exit probabilities:

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

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

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

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

`fadjpbon` *Adjusted p-Values for Bonferroni-Based Graphical Approaches*

Description

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

Usage

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

Arguments

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

Value

A matrix of adjusted p-values.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

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

Examples

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

`fadjpdun` *Adjusted p-Values for Dunnett-Based Graphical Approaches*

Description

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

Usage

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

Arguments

`wgtmat`	The weight matrix for intersection hypotheses.
`p`	The raw p-values for elementary hypotheses.
`family`	The matrix of family indicators for elementary hypotheses.
`corr`	The correlation matrix that should be used for the parametric test. Can contain NAs for unknown correlations between families.

Value

A matrix of adjusted p-values.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

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

Examples

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

family = matrix(c(1,1,0,0,0,0,1,1), nrow=2, ncol=4, byrow=TRUE)
corr = matrix(c(1,0.5,NA,NA, 0.5,1,NA,NA,
                NA,NA,1,0.5, NA,NA,0.5,1),
              nrow = 4, byrow = TRUE)
fadjpdun(wgtmat, pvalues, family, corr)
```

`fadjpsim` *Adjusted p-Values for Simes-Based Graphical Approaches*

Description

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

Usage

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

Arguments

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

Value

A matrix of adjusted p-values.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

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

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

Examples

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

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

`float_to_fraction` *Converting a decimal to a fraction*

Description

Converts a decimal to a fraction based on the algorithm from http://stackoverflow.com/a/5128558/221955.

Usage

```
float_to_fraction(x, tol = 1e-06)
```

Arguments

`x`	The fraction in decimal form.
`tol`	The tolerance level for the conversion error.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
float_to_fraction(5/3)
```

`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

`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, 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)
```

`fquantile` *The Quantiles of a Survival Distribution*

Description

Obtains the quantiles of a survival distribution.

Usage

```
fquantile(S, probs, ...)
```

Arguments

`S`	The survival function of a univariate survival time.
`probs`	The numeric vector of probabilities.
`...`	Additional arguments to be passed to S.

Value

A vector of `length(probs)` for the quantiles.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
fquantile(pweibull, probs = c(0.25, 0.5, 0.75),
         shape = 1.37, scale = 1/0.818, lower.tail = FALSE)
```

`fseqbon` *Group Sequential Trials Using Bonferroni-Based Graphical Approaches*

Description

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

Usage

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

Arguments

`w`	The vector of initial weights for elementary hypotheses.
`G`	The initial transition matrix.
`alpha`	The significance level. Defaults to 0.025.
`kMax`	The maximum number of stages.

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

Value

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

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

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

Examples

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

fseqbon(
  w = c(0.5, 0.5, 0, 0),
  G = matrix(c(0, 0.5, 0.5, 0,  0.5, 0, 0, 0.5,
               0, 1, 0, 0,  1, 0, 0, 0),
             nrow=4, ncol=4, byrow=TRUE),
  alpha = 0.025,
  kMax = 3,
  typeAlphaSpending = rep("sfOF", 4),
```

```
  maxInformation = rep(1, 4),
  p = matrix(c(0.0062, 0.017, 0.009, 0.13,
               0.0002, 0.0035, 0.002, 0.06),
             nrow=4, ncol=2),
  information = matrix(c(rep(1/3, 4), rep(2/3, 4)),
                       nrow=4, ncol=2))
```

`fstdmix` *Adjusted p-Values for Standard Mixture Gatekeeping Procedures*

Description

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

Usage

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

`p`	The raw p-values for elementary 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. It is either "Holm" or "Hochberg", and it defaults to "Hochberg".
`retest`	Whether to allow retesting. It defaults to `TRUE`.

Value

A matrix of adjusted p-values.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

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

`ftrunc` *Adjusted p-Values for Holm, Hochberg, and Hommel Procedures*

Description

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

Usage

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

Arguments

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

Value

A matrix of adjusted p-values.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

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

Examples

```
pvalues <- matrix(c(0.01,0.005,0.015,0.022, 0.02,0.015,0.010,0.023),
                  nrow=2, ncol=4, byrow=TRUE)
ftrunc(pvalues, "hochberg")
```

`fwgtmat` *Weight Matrix for All Intersection Hypotheses*

Description

Obtains the weight matrix for all intersection hypotheses.

Usage

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

Arguments

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

Value

The weight matrix starting with the global null hypothesis.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

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

`getAccrualDurationFromN`

Accrual Duration to Enroll Target Number of Subjects

Description

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

Usage

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

Arguments

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

Value

A vector of accrual durations.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

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

`getADCI` *Confidence Interval After Adaptation*

Description

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

Usage

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

Arguments

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

`alpha`	The significance level of the primary trial. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending for the primary trial. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value of alpha spending for the primary trial. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`spendingTime`	The error spending time of the primary trial. Defaults to missing, in which case, it is the same as `informationRates`.
`L2`	The termination look of the secondary trial.
`zL2`	The z-test statistic at the termination look of the secondary trial.
`INew`	The maximum information of the secondary trial.
`MullerSchafer`	Whether to use the Muller and Schafer (2001) method for trial adaptation.
`informationRatesNew`	The spacing of looks of the secondary trial up to look `L2`.
`efficacyStoppingNew`	The indicators of whether efficacy stopping is allowed at each look of the secondary trial up to look `L2`. Defaults to true if left unspecified.
`typeAlphaSpendingNew`	The type of alpha spending for the secondary trial. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpendingNew`	The parameter value of alpha spending for the secondary trial. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`spendingTimeNew`	The error spending time of the secondary trial up to look `L2`. Defaults to missing, in which case, it is the same as `informationRatesNew`.

Value

A data frame with the following variables:

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

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

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

See Also

`adaptDesign`

Examples

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

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

t = des1$byStageResults$informationRates

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

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

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

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

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

`getADRCI` *Repeated Confidence Interval After Adaptation*

Description

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

Usage

```
getADRCI(
  L = NA_integer_,
  zL = NA_real_,
  IMax = NA_real_,
  kMax = NA_integer_,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  spendingTime = NA_real_,
  L2 = NA_integer_,
  zL2 = NA_real_,
  INew = NA_real_,
  MullerSchafer = 0L,
```

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

Arguments

`L`	The interim adaptation look of the primary trial.
`zL`	The z-test statistic at the interim adaptation look of the primary trial.
`IMax`	The maximum information of the primary trial.
`kMax`	The maximum number of stages of the primary trial.
`informationRates`	The information rates of the primary trial.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage of the primary trial. Defaults to true if left unspecified.
`criticalValues`	The upper boundaries on the z-test statistic scale for efficacy stopping for the primary trial.
`alpha`	The significance level of the primary trial. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending for the primary trial. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value of alpha spending for the primary trial. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`spendingTime`	The error spending time of the primary trial. Defaults to missing, in which case, it is the same as `informationRates`.
`L2`	The look of interest in the secondary trial.
`zL2`	The z-test statistic at the look of the secondary trial.
`INew`	The maximum information of the secondary trial.
`MullerSchafer`	Whether to use the Muller and Schafer (2001) method for trial adaptation.
`informationRatesNew`	The spacing of looks of the secondary trial.
`efficacyStoppingNew`	The indicators of whether efficacy stopping is allowed at each look of the secondary trial up to look `L2`. Defaults to true if left unspecified.

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

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

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

Value

A data frame with the following variables:

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

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

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

See Also

`adaptDesign`

Examples

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

# interim look results
L = 1
```

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

t = des1$byStageResults$informationRates

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

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

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

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

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

`getBound` *Efficacy Boundaries for Group Sequential Design*

Description

Obtains the efficacy stopping boundaries for a group sequential design.

Usage

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

Arguments

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

Details

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

Value

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

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

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

---

getCI *Confidence Interval After Trial Termination*

Description

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

Usage

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

Arguments

`L`	The termination look.
`zL`	The z-test statistic at the termination look.
`IMax`	The maximum information of the trial.

`informationRates`	The information rates up to look `L`.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage up to look `L`. Defaults to true if left unspecified.
`criticalValues`	The upper boundaries on the z-test statistic scale for efficacy stopping up to look `L`.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value of alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`spendingTime`	The error spending time up to look `L`. Defaults to missing, in which case, it is the same as `informationRates`.

Value

A data frame with the following components:

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

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

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

Examples

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

```
                  parameterAlphaSpending = -4))

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

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

---

getCP *Conditional Power Allowing for Varying Parameter Values*

Description

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

Usage

```
getCP(
  INew = NA_real_,
  L = NA_integer_,
  zL = NA_real_,
  theta = NA_real_,
  IMax = NA_real_,
  kMax = NA_integer_,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  spendingTime = NA_real_,
  MullerSchafer = 0L,
  kNew = NA_integer_,
  informationRatesNew = NA_real_,
```

```
  efficacyStoppingNew = NA_integer_,
  futilityStoppingNew = NA_integer_,
  typeAlphaSpendingNew = "sfOF",
  parameterAlphaSpendingNew = NA_real_,
  typeBetaSpendingNew = "none",
  parameterBetaSpendingNew = NA_real_,
  spendingTimeNew = NA_real_,
  varianceRatio = 1
)
```

Arguments

`INew`	The maximum information of the secondary trial.
`L`	The interim adaptation look of the primary trial.
`zL`	The z-test statistic at the interim adaptation look of the primary trial.
`theta`	A scalar or a vector of parameter values of length `kMax + kMax - L` if `MullerSchafer = FALSE` or length `kMax + kNew` if `MullerSchafer = TRUE`.
`IMax`	The maximum information of the primary trial.
`kMax`	The maximum number of stages of the primary trial.
`informationRates`	The information rates of the primary trial.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage of the primary trial. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage of the primary trial. Defaults to true if left unspecified.
`criticalValues`	The upper boundaries on the z-test statistic scale for efficacy stopping for the primary trial.
`alpha`	The significance level of the primary trial. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending for the primary trial. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value of alpha spending for the primary trial. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending for the primary trial. Cumulative alpha spent up to each stage.
`futilityBounds`	The lower boundaries on the z-test statistic scale for futility stopping for the primary trial. Defaults to `rep(-6, kMax-1)` if left unspecified.

`typeBetaSpending`	The type of beta spending for the primary trial. One of the following: "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, and "none" for no early futility stopping. Defaults to "none".
`parameterBetaSpending`	The parameter value of beta spending for the primary trial. Corresponds to rho for "sfKD", and gamma for "sfHSD".
`spendingTime`	The error spending time of the primary trial. Defaults to missing, in which case, it is the same as `informationRates`.
`MullerSchafer`	Whether to use the Muller and Schafer (2001) method for trial adaptation.
`kNew`	The number of looks of the secondary trial.
`informationRatesNew`	The spacing of looks of the secondary trial.
`efficacyStoppingNew`	The indicators of whether efficacy stopping is allowed at each look of the secondary trial. Defaults to true if left unspecified.
`futilityStoppingNew`	The indicators of whether futility stopping is allowed at each look of the secondary trial. Defaults to true if left unspecified.
`typeAlphaSpendingNew`	The type of alpha spending for the secondary trial. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpendingNew`	The parameter value of alpha spending for the secondary trial. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`typeBetaSpendingNew`	The type of beta spending for the secondary trial. One of the following: "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, and "none" for no early futility stopping. Defaults to "none".
`parameterBetaSpendingNew`	The parameter value of beta spending for the secondary trial. Corresponds to rho for "sfKD", and gamma for "sfHSD".
`spendingTimeNew`	The error spending time of the secondary trial. Defaults to missing, in which case, it is the same as `informationRatesNew`.
`varianceRatio`	The ratio of the variance under H0 to the variance under H1.

Value

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

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

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

See Also

`getDesign`

Examples

```
# Conditional power calculation with delayed treatment effect

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

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

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

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

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

# The original target number of events was 298 and the new target is 335
mo2 <- caltime(
  nevents = c(298, 335),
  allocationRatioPlanned = 1,
  accrualTime = seq(0, Ta1),
```

```
  accrualIntensity = enrate*seq(1, Ta1+1)/(Ta1+1),
  piecewiseSurvivalTime = c(0, t1),
  lambda1 = c(lam1, lam2),
  lambda2 = c(lam1, lam1),
  gamma1 = gam,
  gamma2 = gam,
  accrualDuration = Ta,
  followupTime = 1000)

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


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

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

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

`getDesign` *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 & 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.
`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*sqrt(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"))
```

`getDesignAgreement` *Power and Sample Size for Cohen's kappa*

Description

Obtains the power given sample size or obtains the sample size given power for Cohen's kappa.

Usage

```
getDesignAgreement(
  beta = NA_real_,
  n = NA_real_,
  ncats = NA_integer_,
  kappaH0 = NA_real_,
  kappa = NA_real_,
  p1 = NA_real_,
  p2 = NA_real_,
  rounding = TRUE,
  alpha = 0.025
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`ncats`	The number of categories.
`kappaH0`	The kappa coefficient under the null hypothesis.
`kappa`	The kappa coefficient under the alternative hypothesis.
`p1`	The marginal probabilities for the first rater.
`p2`	The marginal probabilities for the second rater. Defaults to be equal to the marginal probabilities for the first rater if not provided.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`alpha`	The one-sided significance level. Defaults to 0.025.

Details

The kappa coefficient is defined as

$$\kappa = \frac{\pi_o - \pi_e}{1 - \pi_e},$$

where $\pi_o = \sum_i \pi_{ii}$ is the observed agreement, and $\pi_e = \sum_i \pi_{i.}\pi_{.i}$ is the expected agreement by chance.

By Fleiss et al. (1969), the variance of $\hat{\kappa}$ is given by

$$Var(\hat{\kappa}) = \frac{v_1}{n},$$

where

$$v_1 = \frac{Q_1 + Q_2 - Q3 - Q4}{(1-\pi_e)^4},$$

$$Q_1 = \pi_o(1-\pi_e)^2,$$

$$Q_2 = (1-\pi_o)^2 \sum_i \sum_j \pi_{ij}(\pi_{i.} + \pi_{.j})^2,$$

$$Q_3 = 2(1-\pi_o)(1-\pi_e) \sum_i \pi_{ii}(\pi_{i.} + \pi_{.i}),$$

$$Q_4 = (\pi_o \pi_e - 2\pi_e + \pi_o)^2.$$

Given κ and marginals $\{(\pi_{i.}, \pi_{.i}) : i = 1, \ldots, k\}$, we obtain π_o. The only unknowns are the double summation in Q_2 and the single summation in Q_3.

We find the optimal configuration of cell probabilities that yield the maximum variance of $\hat{\kappa}$ by treating the problem as a linear programming problem with constraints to match the given marginal probabilities and the observed agreement and ensure that the cell probabilities are nonnegative. This is an extension of Flack et al. (1988) by allowing unequal marginal probabilities of the two raters.

We perform the optimization under both the null and alternative hypotheses to obtain $\max Var(\hat{\kappa}|\kappa = \kappa_0)$ and $\max Var(\hat{\kappa}|\kappa = \kappa_1)$ for a single subject, and then calculate the sample size or power according to the following equation:

$$\sqrt{n}|\kappa - \kappa_0| = z_{1-\alpha}\sqrt{\max Var(\hat{\kappa}|\kappa = \kappa_0)} + z_{1-\beta}\sqrt{\max Var(\hat{\kappa}|\kappa = \kappa_1)}.$$

Value

An S3 class `designAgreement` object with the following components:

- `power`: The power to reject the null hypothesis.
- `alpha`: The one-sided significance level.
- `n`: The total sample size.
- `ncats`: The number of categories.
- `kappaH0`: The kappa coefficient under the null hypothesis.
- `kappa`: The kappa coefficient under the alternative hypothesis.
- `p1`: The marginal probabilities for the first rater.
- `p2`: The marginal probabilities for the second rater.
- `piH0`: The cell probabilities that maximize the variance of estimated kappa under H0.
- `pi`: The cell probabilities that maximize the variance of estimated kappa under H1.
- `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

V. F. Flack, A. A. Afifi, and P. A. Lachenbruch. Sample size determinations for the two rater kappa statistic. Psychometrika 1988; 53:321-325.

Examples

```
(design1 <- getDesignAgreement(
  beta = 0.2, n = NA, ncats = 4, kappaH0 = 0.4, kappa = 0.6,
  p1 = c(0.1, 0.2, 0.3, 0.4), p2 = c(0.15, 0.2, 0.24, 0.41),
  rounding = TRUE, alpha = 0.05))
```

`getDesignANOVA` *Power and Sample Size for One-Way ANOVA*

Description

Obtains the power and sample size for one-way analysis of variance.

Usage

```
getDesignANOVA(
  beta = NA_real_,
  n = NA_real_,
  ngroups = 2,
  means = NA_real_,
  stDev = 1,
  allocationRatioPlanned = NA_real_,
  rounding = TRUE,
  alpha = 0.05
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`ngroups`	The number of treatment groups.
`means`	The treatment group means.
`stDev`	The common standard deviation.
`allocationRatioPlanned`	Allocation ratio for the treatment groups. It has length `ngroups - 1` or `ngroups`. If it is of length `ngroups - 1`, then the last treatment group will assume value 1 for allocation ratio.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`alpha`	The two-sided significance level. Defaults to 0.05.

Details

Let $\{\mu_i : i = 1, \ldots, k\}$ denote the group means, and $\{r_i : i = 1, \ldots, k\}$ denote the randomization probabilities to the k treatment groups. Let σ denote the common standard deviation, and n denote the total sample size. Then the F-statistic

$$F = \frac{SSR/(k-1)}{SSE/(n-k)} \sim F_{k-1,n-k,\lambda}$$

where

$$\lambda = n \sum_{i=1}^{k} r_i(\mu_i - \bar{\mu})^2/\sigma^2$$

is the noncentrality parameter, and $\bar{\mu} = \sum_{i=1}^{k} r_i \mu_i$.

Value

An S3 class `designANOVA` object with the following components:

- `power`: The power to reject the null hypothesis that there is no difference among the treatment groups.
- `alpha`: The two-sided significance level.
- `n`: The number of subjects.
- `ngroups`: The number of treatment groups.
- `means`: The treatment group means.
- `stDev`: The common standard deviation.
- `effectsize`: The effect size.
- `allocationRatioPlanned`: Allocation ratio for the treatment groups.
- `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
(design1 <- getDesignANOVA(
  beta = 0.1, ngroups = 4, means = c(1.5, 2.5, 2, 0),
  stDev = 3.5, allocationRatioPlanned = c(2, 2, 2, 1),
  alpha = 0.05))
```

---

getDesignANOVAContrast

Power and Sample Size for One-Way ANOVA Contrast

Description

Obtains the power and sample size for a single contrast in one-way analysis of variance.

Usage

```
getDesignANOVAContrast(
  beta = NA_real_,
  n = NA_real_,
  ngroups = 2,
  means = NA_real_,
  stDev = 1,
  contrast = NA_real_,
  meanContrastH0 = 0,
  allocationRatioPlanned = NA_real_,
  rounding = TRUE,
  alpha = 0.025
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`ngroups`	The number of treatment groups.
`means`	The treatment group means.
`stDev`	The common standard deviation.
`contrast`	The coefficients for the single contrast.
`meanContrastH0`	The mean of the contrast under the null hypothesis.
`allocationRatioPlanned`	Allocation ratio for the treatment groups. It has length `ngroups - 1` or `ngroups`. If it is of length `ngroups - 1`, then the last treatment group will assume value 1 for allocation ratio.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`alpha`	The one-sided significance level. Defaults to 0.025.

Value

An S3 class `designANOVAContrast` object with the following components:

- `power`: The power to reject the null hypothesis for the treatment contrast.

- `alpha`: The one-sided significance level.
- `n`: The number of subjects.
- `ngroups`: The number of treatment groups.
- `means`: The treatment group means.
- `stDev`: The common standard deviation.
- `contrast`: The coefficients for the single contrast.
- `meanContrastH0`: The mean of the contrast under the null hypothesis.
- `meanContrast`: The mean of the contrast under the alternative hypothesis.
- `effectsize`: The effect size.
- `allocationRatioPlanned`: Allocation ratio for the treatment groups.
- `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignANOVAContrast(
  beta = 0.1, ngroups = 4, means = c(1.5, 2.5, 2, 0),
  stDev = 3.5, contrast = c(1, 1, 1, -3),
  allocationRatioPlanned = c(2, 2, 2, 1),
  alpha = 0.025))
```

`getDesignEquiv` *Power and Sample Size for a Generic Group Sequential Equivalence Design*

Description

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

Usage

```
getDesignEquiv(
  beta = NA_real_,
  IMax = NA_real_,
  thetaLower = NA_real_,
  thetaUpper = NA_real_,
  theta = 0,
  kMax = 1L,
  informationRates = NA_real_,
```

```
  criticalValues = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`IMax`	The maximum information. Either `beta` or `IMax` should be provided while the other one should be missing.
`thetaLower`	The parameter value at the lower equivalence limit.
`thetaUpper`	The parameter value at the upper equivalence limit.
`theta`	The parameter value under the alternative hypothesis.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level for each of the two one-sided tests, e.g., 0.05.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.

Details

Consider the equivalence design with two one-sided hypotheses:

$$H_{10} : \theta \leq \theta_{10},$$

$$H_{20} : \theta \geq \theta_{20}.$$

We reject H_{10} at or before look k if

$$Z_{1j} = (\hat{\theta}_j - \theta_{10})\sqrt{I_j} \geq b_j$$

for some $j = 1, \ldots, k$, where $\{b_j : j = 1, \ldots, K\}$ are the critical values associated with the specified alpha-spending function, and I_j is the information for θ (inverse variance of $\hat{\theta}$) at the jth look. For example, for estimating the risk difference $\theta = \pi_1 - \pi_2$,

$$I_j = \left\{ \frac{\pi_1(1-\pi_1)}{n_{1j}} + \frac{\pi_2(1-\pi_2)}{n_{2j}} \right\}^{-1}.$$

It follows that

$$(Z_{1j} \geq b_j) = (Z_j \geq b_j + \theta_{10}\sqrt{I_j}),$$

where $Z_j = \hat{\theta}_j \sqrt{I_j}$.

Similarly, we reject H_{20} at or before look k if

$$Z_{2j} = (\hat{\theta}_j - \theta_{20})\sqrt{I_j} \leq -b_j$$

for some $j = 1, \ldots, k$. We have

$$(Z_{2j} \leq -b_j) = (Z_j \leq -b_j + \theta_{20}\sqrt{I_j}).$$

Let $l_j = b_j + \theta_{10}\sqrt{I_j}$, and $u_j = -b_j + \theta_{20}\sqrt{I_j}$. The cumulative probability to reject $H_0 = H_{10} \cup H_{20}$ at or before look k under the alternative hypothesis H_1 is given by

$$P_\theta \left(\cup_{j=1}^k (Z_{1j} \geq b_j) \cap \cup_{j=1}^k (Z_{2j} \leq -b_j) \right) = p_1 + p_2 - p_{12},$$

where

$$p_1 = P_\theta \left(\cup_{j=1}^k (Z_{1j} \geq b_j) \right) = P_\theta \left(\cup_{j=1}^k (Z_j \geq l_j) \right),$$
$$p_2 = P_\theta \left(\cup_{j=1}^k (Z_{2j} \leq -b_j) \right) = P_\theta \left(\cup_{j=1}^k (Z_j \leq u_j) \right),$$

and

$$p_{12} = P_\theta \left(\cup_{j=1}^k (Z_j \geq l_j) \cup (Z_j \leq u_j) \right).$$

Of note, both p_1 and p_2 can be evaluated using one-sided exit probabilities for group sequential designs. If there exists $j \leq k$ such that $l_j \leq u_j$, then $p_{12} = 1$. Otherwise, p_{12} can be evaluated using two-sided exit probabilities for group sequential designs.

Since the equivalent hypothesis is tested using two one-sided tests, the type I error is controlled. To evaluate the attained type I error of the equivalence trial under H_{10} (or H_{20}), we simply fix the control group parameters, update the active treatment group parameters according to the null hypothesis, and use the parameters in the power calculation outlined above.

Value

An S3 class `designEquiv` object with three components:

- `overallResults`: A data frame containing the following variables:
  - `overallReject`: The overall rejection probability.
  - `alpha`: The overall significance level.
  - `attainedAlphaH10`: The attained significance level under H10.
  - `attainedAlphaH20`: The attained significance level under H20.
  - `kMax`: The number of stages.

  - `thetaLower`: The parameter value at the lower equivalence limit.
  - `thetaUpper`: The parameter value at the upper equivalence limit.
  - `theta`: The parameter value under the alternative hypothesis.
  - `information`: The maximum information.
  - `expectedInformationH1`: The expected information under H1.
  - `expectedInformationH10`: The expected information under H10.
  - `expectedInformationH20`: The expected information under H20.
- `byStageResults`: A data frame containing the following variables:
  - `informationRates`: The information rates.
  - `efficacyBounds`: The efficacy boundaries on the Z-scale for each of the two one-sided tests.
  - `rejectPerStage`: The probability for efficacy stopping.
  - `cumulativeRejection`: The cumulative probability for efficacy stopping.
  - `cumulativeAlphaSpent`: The cumulative alpha for each of the two one-sided tests.
  - `cumulativeAttainedAlphaH10`: The cumulative probability for efficacy stopping under H10.
  - `cumulativeAttainedAlphaH20`: The cumulative probability for efficacy stopping under H20.
  - `efficacyThetaLower`: The efficacy boundaries on the parameter scale for the one-sided null hypothesis at the lower equivalence limit.
  - `efficacyThetaUpper`: The efficacy boundaries on the parameter scale for the one-sided null hypothesis at the upper equivalence limit.
  - `efficacyP`: The efficacy bounds on the p-value scale for each of the two one-sided tests.
  - `information`: The cumulative information.
- `settings`: A list containing the following components:
  - `typeAlphaSpending`: The type of alpha spending.
  - `parameterAlphaSpending`: The parameter value for alpha spending.
  - `userAlphaSpending`: The user defined alpha spending.
  - `spendingTime`: The error spending time at each analysis.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
# Example 1: obtain the maximum information given power
(design1 <- getDesignEquiv(
  beta = 0.2, thetaLower = log(0.8), thetaUpper = log(1.25),
  kMax = 2, informationRates = c(0.5, 1),
  alpha = 0.05, typeAlphaSpending = "sfOF"))


# Example 2: obtain power given the maximum information
(design2 <- getDesignEquiv(
  IMax = 72.5, thetaLower = log(0.7), thetaUpper = -log(0.7),
```

```
  kMax = 3, informationRates = c(0.5, 0.75, 1),
  alpha = 0.05, typeAlphaSpending = "sfOF"))
```

`getDesignFisherExact` *Power and Sample Size for Fisher's Exact Test for Two Proportions*

Description

Obtains the power given sample size or obtains the sample size given power for Fisher's exact test for two proportions.

Usage

```
getDesignFisherExact(
  beta = NA_real_,
  n = NA_real_,
  pi1 = NA_real_,
  pi2 = NA_real_,
  allocationRatioPlanned = 1,
  alpha = 0.05
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`pi1`	The assumed probability for the active treatment group.
`pi2`	The assumed probability for the control group.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`alpha`	The two-sided significance level. Defaults to 0.05.

Value

A data frame with the following variables:

- `alpha`: The two-sided significance level.
- `power`: The power.
- `n`: The sample size.
- `pi1`: The assumed probability for the active treatment group.
- `pi2`: The assumed probability for the control group.
- `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
(design1 <- getDesignFisherExact(
  beta = 0.2, pi1 = 0.5, pi2 = 0.2, alpha = 0.05))
```

`getDesignLogistic` *Power and Sample Size for Logistic Regression*

Description

Obtains the power given sample size or obtains the sample size given power for logistic regression of a binary response given the covariate of interest and other covariates.

Usage

```
getDesignLogistic(
  beta = NA_real_,
  n = NA_real_,
  ncovariates = NA_integer_,
  nconfigs = NA_integer_,
  x = NA_real_,
  pconfigs = NA_real_,
  corr = 0,
  oddsratios = NA_real_,
  responseprob = NA_real_,
  rounding = TRUE,
  alpha = 0.05
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`ncovariates`	The number of covariates.
`nconfigs`	The number of configurations of discretized covariate values.
`x`	The matrix of covariate values.
`pconfigs`	The vector of probabilities for the configurations.
`corr`	The multiple correlation between the predictor and other covariates. Defaults to 0.
`oddsratios`	The odds ratios for one unit increase in the covariates.

`responseprob`	The response probability in the full model when all predictor variables are equal to their means.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`alpha`	The two-sided significance level. Defaults to 0.05.

Details

We consider the logistic regression of a binary response variable Y on a set of predictor variables $x = (x_1, \ldots, x_K)^T$ with x_1 being the covariate of interest: $\log \frac{P(Y_i=1)}{1-P(Y_i=1)} = \psi_0 + x_i^T \psi$, where $\psi = (\psi_1, \ldots, \psi_K)^T$. Similar to Self et al (1992), we assume that all covariates are either inherently discrete or discretized from continuous distributions (e.g. using the quantiles). Let m denote the total number of configurations of the covariate values. Let

$$\pi_i = P(x = x_i), i = 1, \ldots, m$$

denote the probabilities for the configurations of the covariates under independence. The likelihood ratio test statistic for testing $H_0 : \psi_1 = 0$ can be approximated by a noncentral chi-square distribution with one degree of freedom and noncentrality parameter

$$\Delta = 2\sum_{i=1}^m \pi_i [b'(\theta_i)(\theta_i - \theta_i^*) - \{b(\theta_i) - b(\theta_i^*)\}],$$

where

$$\theta_i = \psi_0 + \sum_{j=1}^k \psi_j x_{ij},$$

$$\theta_i^* = \psi_0^* + \sum_{j=2}^k \psi_j^* x_{ij},$$

for $\psi_0^* = \psi_0 + \psi_1\mu_1$, and $\psi_j^* = \psi_j$ for $j = 2, \ldots, K$. Here μ_1 is the mean of x_1, e.g., $\mu_1 = \sum_i \pi_i x_{i1}$. In addition, by formulating the logistic regression in the framework of generalized linear models,

$$b(\theta) = \log(1 + \exp(\theta)),$$

and

$$b'(\theta) = \frac{\exp(\theta)}{1 + \exp(\theta)}.$$

The regression coefficients ψ can be obtained by taking the log of the odds ratios for the covariates. The intercept ψ_0 can be derived as

$$\psi_0 = \log(\bar{\mu}/(1 - \bar{\mu})) - \sum_{j=1}^K \psi_j \mu_j,$$

where $\bar{\mu}$ denotes the response probability when all predictor variables are equal to their means.

Finally, let ρ denote the multiple correlation between the predictor and other covariates. The noncentrality parameter of the chi-square test is adjusted downward by multiplying by $1 - \rho^2$.

Value

An S3 class `designLogistic` object with the following components:

- `power`: The power to reject the null hypothesis.
- `alpha`: The two-sided significance level.
- `n`: The total sample size.
- `ncovariates`: The number of covariates.
- `nconfigs`: The number of configurations of discretized covariate values.
- `x`: The matrix of covariate values.
- `pconfigs`: The vector of probabilities for the configurations.
- `corr`: The multiple correlation between the predictor and other covariates.
- `oddsratios`: The odds ratios for one unit increase in the covariates.
- `responseprob`: The response probability in the full model when all predictor variables are equal to their means.
- `effectsize`: The effect size for the chi-square test.
- `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

Steven G. Self, Robert H. Mauritsen and Jill Ohara. Power calculations for likelihood ratio tests in generalized linear models. Biometrics 1992; 48:31-39.

Examples

```
# two ordinal covariates
x1 = c(5, 10, 15, 20)
px1 = c(0.2, 0.3, 0.3, 0.2)

x2 = c(2, 4, 6)
px2 = c(0.4, 0.4, 0.2)

# discretizing a normal distribution with mean 4 and standard deviation 2
nbins = 10
x3 = qnorm(((1:nbins) - 0.5)/nbins)*2 + 4
px3 = rep(1/nbins, nbins)

# combination of covariate values
nconfigs = length(x1)*length(x2)*length(x3)
x = expand.grid(x3 = x3, x2 = x2, x1 = x1)
x = as.matrix(x[, ncol(x):1])

# probabilities for the covariate configurations under independence
pconfigs = as.numeric(px1 %x% px2 %x% px3)
```

```
# convert the odds ratio for the predictor variable in 5-unit change
# to the odds ratio in 1-unit change
(design1 <- getDesignLogistic(
  beta = 0.1, ncovariates = 3,
  nconfigs = nconfigs,
  x = x,
  pconfigs = pconfigs,
  oddsratios = c(1.2^(1/5), 1.4, 1.3),
  responseprob = 0.25,
  alpha = 0.1))
```

`getDesignMeanDiff` *Group Sequential Design for Two-Sample Mean Difference*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for two-sample mean difference.

Usage

```
getDesignMeanDiff(
  beta = NA_real_,
  n = NA_real_,
  meanDiffH0 = 0,
  meanDiff = 0.5,
  stDev = 1,
  allocationRatioPlanned = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`meanDiffH0`	The mean difference under the null hypothesis. Defaults to 0.
`meanDiff`	The mean difference under the alternative hypothesis.
`stDev`	The standard deviation.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.

Value

An S3 class `designMeanDiff` 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*sqrt(information)`.
    - `inflationFactor`: The inflation factor (relative to the fixed design).
    - `numberOfSubjects`: The maximum number of subjects.
    - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
    - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
    - `meanDiffH0`: The mean difference under the null hypothesis.
    - `meanDiff`: The mean difference under the alternative hypothesis.
    - `stDev`: The standard deviation.
- `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.
    - `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.
  - `efficacyMeanDiff`: The efficacy boundaries on the mean difference scale.
  - `futilityMeanDiff`: The futility boundaries on the mean difference scale.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
# Example 1: group sequential trial power calculation
(design1 <- getDesignMeanDiff(
  beta = NA, n = 456, meanDiff = 9, stDev = 32,
  kMax = 5, alpha = 0.025, typeAlphaSpending = "sfOF",
  typeBetaSpending = "sfP"))

# Example 2: sample size calculation for two-sample t-test
(design2 <- getDesignMeanDiff(
  beta = 0.1, n = NA, meanDiff = 0.3, stDev = 1,
  normalApproximation = FALSE, alpha = 0.025))
```

`getDesignMeanDiffCarryover` *Power and Sample Size for Direct Treatment Effects in Crossover Trials*

Description

Obtains the power and sample size for direct treatment effects in crossover trials accounting or without accounting for carryover effects.

Usage

```
getDesignMeanDiffCarryover(
  beta = NA_real_,
  n = NA_real_,
  trtpair = NA_real_,
  carryover = TRUE,
  meanDiffH0 = 0,
  meanDiff = 0.5,
  stDev = 1,
  corr = 0.5,
  design = NA_real_,
  cumdrop = NA_real_,
  allocationRatioPlanned = NA_real_,
  normalApproximation = FALSE,
  rounding = TRUE,
  alpha = 0.025
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`trtpair`	The treatment pair of interest to power the study. If not given, it defaults to comparing the first treatment to the last treatment.
`carryover`	Whether to account for carryover effects in the power calculation. Defaults to TRUE.
`meanDiffH0`	The mean difference for the treatment pair of interest under the null hypothesis. Defaults to 0.
`meanDiff`	The mean difference for the treatment pair of interest under the alternative hypothesis.
`stDev`	The standard deviation for within-subject random error.
`corr`	The intra-subject correlation due to subject random effect.
`design`	The crossover design represented by a matrix with rows indexing the sequences, columns indexing the periods, and matrix entries indicating the treatments.

`cumdrop`	The cumulative dropout rate over periods.
`allocationRatioPlanned`	Allocation ratio for the sequences. Defaults to equal randomization if not provided.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
`rounding`	Whether to round up the sample size. Defaults to TRUE for sample size rounding.
`alpha`	The one-sided significance level. Defaults to 0.025.

Details

The linear mixed-effects model to assess the direct treatment effects in the presence of carryover treatment effects is given by

$$y_{ijk} = \mu + \alpha_i + b_{ij} + \gamma_k + \tau_{d(i,k)} + \lambda_{c(i,k-1)} + e_{ijk}$$

$$i = 1, \ldots, n; j = 1, \ldots, r_i; k = 1, \ldots, p; d, c = 1, \ldots, t$$

where μ is the general mean, α_i is the effect of the ith treatment sequence, b_{ij} is the random effect with variance σ_b^2 for the jth subject of the ith treatment sequence, γ_k is the period effect, and e_{ijk} is the random error with variance σ^2 for the subject in period k. The direct effect of the treatment administered in period k of sequence i is $\tau_{d(i,k)}$, and $\lambda_{c(i,k-1)}$ is the carryover effect of the treatment administered in period $k-1$ of sequence i. The value of the carryover effect for the observed response in the first period is $\lambda_{c(i,0)} = 0$ since there is no carryover effect in the first period. The intra-subject correlation due to the subject random effect is

$$\rho = \frac{\sigma_b^2}{\sigma_b^2 + \sigma^2}.$$

Therefore, `stDev` = σ^2 and `corr` = ρ. By constructing the design matrix X for the linear model with a compound symmetry covariance matrix for the response vector of a subject, we can obtain

$$Var(\hat{\beta}) = (X'V^{-1}X)^{-1}.$$

The covariance matrix for the direct treatment effects and carryover treatment effects can be extracted from the appropriate sub-matrices. The covariance matrix for the direct treatment effects without accounting for the carryover treatment effects can be obtained by omitting the carryover effect terms from the model.

The power is for the direct treatment effect for the treatment pair of interest with or without accounting for carryover effects as determined by the input parameter `carryover`. The relative efficiency is for the direct treatment effect for the treatment pair of interest accounting for carryover effects relative to that without accounting for carryover effects.

The degrees of freedom for the t-test accounting for carryover effects can be calculated as the total number of observations minus the number of subjects minus $p-1$ minus $2(t-1)$ to account for the subject effect, period effect, and direct and carryover treatment effects. The degrees of freedom for the t-test without accounting for carryover effects is equal to the total number of observations minus the number of subjects minus $p-1$ minus $t-1$ to account for the subject effect, period effect, and direct treatment effects.

Value

An S3 class `designMeanDiffCarryover` object with the following components:

- `power`: The power to reject the null hypothesis.
- `alpha`: The one-sided significance level.
- `numberOfSubjects`: The maximum number of subjects.
- `trtpair`: The treatment pair of interest to power the study.
- `carryover`: Whether to account for carryover effects in the power calculation.
- `meanDiffH0`: The mean difference for the treatment pair of interest under the null hypothesis.
- `meanDiff`: The mean difference for the treatment pair of interest under the alternative hypothesis.
- `stDev`: The standard deviation for within-subject random error.
- `corr`: The intra-subject correlation due to subject random effect.
- `design`: The crossover design represented by a matrix with rows indexing the sequences, columns indexing the periods, and matrix entries indicating the treatments.
- `designMatrix`: The design matrix accounting for intercept, sequence, period, direct treatment effects and carryover treatment effects when `carryover = TRUE`, or the design matrix accounting for intercept, sequence, period, and direct treatment effects when `carryover = FALSE`.
- `nseq`: The number of sequences.
- `nprd`: The number of periods.
- `ntrt`: The number of treatments.
- `cumdrop`: The cumulative dropout rate over periods.
- `V_direct_only`: The covariance matrix for direct treatment effects without accounting for carryover effects. The treatment comparisons for the covariance matrix are for the first $t-1$ treatments relative to the last treatment.
- `V_direct_carry`: The covariance matrix for direct and carryover treatment effects.
- `v_direct_only`: The variance of the direct treatment effect for the treatment pair of interest without accounting for carryover effects.
- `v_direct`: The variance of the direct treatment effect for the treatment pair of interest accounting for carryover effects.
- `v_carry`: The variance of the carryover treatment effect for the treatment pair of interest.
- `releff_direct`: The relative efficiency of the design for estimating the direct treatment effect for the treatment pair of interest after accounting for carryover effects with respect to that without accounting for carryover effects. This is equal to `v_direct_only/v_direct`.
- `releff_carry`: The relative efficiency of the design for estimating the carryover effect for the treatment pair of interest. This is equal to `v_direct_only/v_carry`.
- `half_width`: The half-width of the confidence interval for the direct treatment effect for the treatment pair of interest.
- `nu`: Degrees of freedom for the t-test.
- `allocationRatioPlanned`: Allocation ratio for the sequences.
- `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
- `rounding`: Whether to round up the sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

Robert O. Kuehl. Design of Experiments: Statistical Principles of Research Design and Analysis. Brooks/Cole: Pacific Grove, CA. 2000.

Examples

```
# Williams design for 4 treatments

(design1 = getDesignMeanDiffCarryover(
  beta = 0.2, n = NA,
  meanDiff = 0.5, stDev = 1,
  design = matrix(c(1, 4, 2, 3,
                    2, 1, 3, 4,
                    3, 2, 4, 1,
                    4, 3, 1, 2),
                  4, 4, byrow = TRUE),
  alpha = 0.025))
```

`getDesignMeanDiffCarryoverEquiv` *Power and Sample Size for Equivalence in Direct Treatment Effects in Crossover Trials*

Description

Obtains the power and sample size for equivalence in direct treatment effects in crossover trials accounting or without accounting for carryover effects.

Usage

```
getDesignMeanDiffCarryoverEquiv(
  beta = NA_real_,
  n = NA_real_,
  trtpair = NA_real_,
  carryover = TRUE,
  meanDiffLower = NA_real_,
  meanDiffUpper = NA_real_,
  meanDiff = 0,
  stDev = 1,
  corr = 0.5,
  design = NA_real_,
  cumdrop = NA_real_,
  allocationRatioPlanned = NA_real_,
```

```
  normalApproximation = FALSE,
  rounding = TRUE,
  alpha = 0.025
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`trtpair`	The treatment pair of interest to power the study. If not given, it defaults to comparing the first treatment to the last treatment.
`carryover`	Whether to account for carryover effects in the power calculation. Defaults to TRUE.
`meanDiffLower`	The lower equivalence limit of mean difference for the treatment pair of interest.
`meanDiffUpper`	The upper equivalence limit of mean difference for the treatment pair of interest.
`meanDiff`	The mean difference under the alternative hypothesis,
`stDev`	The standard deviation for within-subject random error.
`corr`	The intra-subject correlation due to subject random effect.
`design`	The crossover design represented by a matrix with rows indexing the sequences, columns indexing the periods, and matrix entries indicating the treatments.
`cumdrop`	The cumulative dropout rate over periods.
`allocationRatioPlanned`	Allocation ratio for the sequences. Defaults to equal randomization if not provided.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
`rounding`	Whether to round up the sample size. Defaults to TRUE for sample size rounding.
`alpha`	The one-sided significance level. Defaults to 0.025.

Details

The linear mixed-effects model to assess the direct treatment effects in the presence of carryover treatment effects is given by

$$y_{ijk} = \mu + \alpha_i + b_{ij} + \gamma_k + \tau_{d(i,k)} + \lambda_{c(i,k-1)} + e_{ijk}$$

$$i = 1, \ldots, n; j = 1, \ldots, r_i; k = 1, \ldots, p; d, c = 1, \ldots, t$$

where μ is the general mean, α_i is the effect of the ith treatment sequence, b_{ij} is the random effect with variance σ_b^2 for the jth subject of the ith treatment sequence, γ_k is the period effect, and e_{ijk} is the random error with variance σ^2 for the subject in period k. The direct effect of the treatment administered in period k of sequence i is $\tau_{d(i,k)}$, and $\lambda_{c(i,k-1)}$ is the carryover effect of the treatment administered in period $k-1$ of sequence i. The value of the carryover effect for the observed response in the first period is $\lambda_{c(i,0)} = 0$ since there is no carryover effect in the first period. The intra-subject correlation due to the subject random effect is

$$\rho = \frac{\sigma_b^2}{\sigma_b^2 + \sigma^2}.$$

Therefore, `stDev` = σ^2 and `corr` = ρ. By constructing the design matrix X for the linear model with a compound symmetry covariance matrix for the response vector of a subject, we can obtain

$$Var(\hat{\beta}) = (X'V^{-1}X)^{-1}.$$

The covariance matrix for the direct treatment effects and carryover treatment effects can be extracted from the appropriate sub-matrices. The covariance matrix for the direct treatment effects without accounting for the carryover treatment effects can be obtained by omitting the carryover effect terms from the model.

The power is for the direct treatment effect for the treatment pair of interest with or without accounting for carryover effects as determined by the input parameter `carryover`. The relative efficiency is for the direct treatment effect for the treatment pair of interest accounting for carryover effects relative to that without accounting for carryover effects.

The degrees of freedom for the t-test accounting for carryover effects can be calculated as the total number of observations minus the number of subjects minus $p-1$ minus $2(t-1)$ to account for the subject effect, period effect, and direct and carryover treatment effects. The degrees of freedom for the t-test without accounting for carryover effects is equal to the total number of observations minus the number of subjects minus $p-1$ minus $t-1$ to account for the subject effect, period effect, and direct treatment effects.

Value

An S3 class `designMeanDiffCarryover` object with the following components:

- `power`: The power to reject the null hypothesis.
- `alpha`: The one-sided significance level.
- `numberOfSubjects`: The maximum number of subjects.
- `trtpair`: The treatment pair of interest to power the study.
- `carryover`: Whether to account for carryover effects in the power calculation.
- `meanDiffLower`: The lower equivalence limit of mean difference for the treatment pair of interest.
- `meanDiffUpper`: The upper equivalence limit of mean difference for the treatment pair of interest.
- `meanDiff`: The mean difference for the treatment pair of interest under the alternative hypothesis.
- `stDev`: The standard deviation for within-subject random error.
- `corr`: The intra-subject correlation due to subject random effect.
- `design`: The crossover design represented by a matrix with rows indexing the sequences, columns indexing the periods, and matrix entries indicating the treatments.

- `designMatrix`: The design matrix accounting for intercept, sequence, period, direct treatment effects and carryover treatment effects when `carryover = TRUE`, or the design matrix accounting for intercept, sequence, period, and direct treatment effects when `carryover = FALSE`.
- `nseq`: The number of sequences.
- `nprd`: The number of periods.
- `ntrt`: The number of treatments.
- `cumdrop`: The cumulative dropout rate over periods.
- `V_direct_only`: The covariance matrix for direct treatment effects without accounting for carryover effects. The treatment comparisons for the covariance matrix are for the first $t-1$ treatments relative to the last treatment.
- `V_direct_carry`: The covariance matrix for direct and carryover treatment effects.
- `v_direct_only`: The variance of the direct treatment effect for the treatment pair of interest without accounting for carryover effects.
- `v_direct`: The variance of the direct treatment effect for the treatment pair of interest accounting for carryover effects.
- `v_carry`: The variance of the carryover treatment effect for the treatment pair of interest.
- `releff_direct`: The relative efficiency of the design for estimating the direct treatment effect for the treatment pair of interest after accounting for carryover effects with respect to that without accounting for carryover effects. This is equal to `v_direct_only/v_direct`.
- `releff_carry`: The relative efficiency of the design for estimating the carryover effect for the treatment pair of interest. This is equal to `v_direct_only/v_carry`.
- `half_width`: The half-width of the confidence interval for the direct treatment effect for the treatment pair of interest.
- `nu`: Degrees of freedom for the t-test.
- `allocationRatioPlanned`: Allocation ratio for the sequences.
- `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
- `rounding`: Whether to round up the sample size.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

Robert O. Kuehl. Design of Experiments: Statistical Principles of Research Design and Analysis. Brooks/Cole: Pacific Grove, CA. 2000.

Examples

```
# Williams design for 4 treatments

(design1 = getDesignMeanDiffCarryoverEquiv(
```

```
  beta = 0.2, n = NA,
  meanDiffLower = -1.3, meanDiffUpper = 1.3,
  meanDiff = 0, stDev = 2.2,
  design = matrix(c(1, 4, 2, 3,
                    2, 1, 3, 4,
                    3, 2, 4, 1,
                    4, 3, 1, 2),
                  4, 4, byrow = TRUE),
  alpha = 0.025))
```

`getDesignMeanDiffEquiv`

Group Sequential Design for Equivalence in Two-Sample Mean Difference

---

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for equivalence in two-sample mean difference.

Usage

```
getDesignMeanDiffEquiv(
  beta = NA_real_,
  n = NA_real_,
  meanDiffLower = NA_real_,
  meanDiffUpper = NA_real_,
  meanDiff = 0,
  stDev = 1,
  allocationRatioPlanned = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`meanDiffLower`	The lower equivalence limit of mean difference.

`meanDiffUpper`	The upper equivalence limit of mean difference.
`meanDiff`	The mean difference under the alternative hypothesis.
`stDev`	The standard deviation.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.

Value

An S3 class `designMeanDiffEquiv` object with three components:

- `overallResults`: A data frame containing the following variables:
  - `overallReject`: The overall rejection probability.
  - `alpha`: The significance level for each of the two one-sided tests. Defaults to 0.05.
  - `attainedAlpha`: The attained significance level.
  - `kMax`: The number of stages.
  - `information`: The maximum information.
  - `expectedInformationH1`: The expected information under H1.
  - `expectedInformationH0`: The expected information under H0.
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.

  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `meanDiffLower`: The lower equivalence limit of mean difference.
  - `meanDiffUpper`: The upper equivalence limit of mean difference.
  - `meanDiff`: The mean difference under the alternative hypothesis.
  - `stDev`: The standard deviation.
- `byStageResults`: A data frame containing the following variables:
  - `informationRates`: The information rates.
  - `efficacyBounds`: The efficacy boundaries on the Z-scale for each of the two one-sided tests.
  - `rejectPerStage`: The probability for efficacy stopping.
  - `cumulativeRejection`: The cumulative probability for efficacy stopping.
  - `cumulativeAlphaSpent`: The cumulative alpha for each of the two one-sided tests.
  - `cumulativeAttainedAlpha`: The cumulative probability for efficacy stopping under H0.
  - `efficacyMeanDiffLower`: The efficacy boundaries on the mean difference scale for the one-sided null hypothesis on the lower equivalence limit.
  - `efficacyMeanDiffUpper`: The efficacy boundaries on the mean difference scale for the one-sided null hypothesis on the upper equivalence limit.
  - `efficacyP`: The efficacy bounds on the p-value scale for each of the two one-sided tests.
  - `information`: The cumulative information.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `spendingTime`: The error spending time at each analysis.
  - `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Example 1: group sequential trial power calculation
(design1 <- getDesignMeanDiffEquiv(
  beta = 0.1, n = NA, meanDiffLower = -1.3, meanDiffUpper = 1.3,
  meanDiff = 0, stDev = 2.2,
  kMax = 4, alpha = 0.05, typeAlphaSpending = "sfOF"))

# Example 2: sample size calculation for t-test
(design2 <- getDesignMeanDiffEquiv(
```

```
  beta = 0.1, n = NA, meanDiffLower = -1.3, meanDiffUpper = 1.3,
  meanDiff = 0, stDev = 2.2,
  normalApproximation = FALSE, alpha = 0.05))
```

---

getDesignMeanDiffMMRM *Group Sequential Design for Two-Sample Mean Difference From the MMRM Model*

Description

Obtains the power and sample size for two-sample mean difference at the last time point from the mixed-model for repeated measures (MMRM) model.

Usage

```
getDesignMeanDiffMMRM(
  beta = NA_real_,
  meanDiffH0 = 0,
  meanDiff = 0.5,
  k = 1,
  t = NA_real_,
  covar1 = diag(k),
  covar2 = NA_real_,
  accrualTime = 0,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0,
  gamma1 = 0,
  gamma2 = 0,
  accrualDuration = NA_real_,
  allocationRatioPlanned = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`meanDiffH0`	The mean difference at the last time point under the null hypothesis. Defaults to 0.
`meanDiff`	The mean difference at the last time point under the alternative hypothesis.
`k`	The number of postbaseline time points.
`t`	The postbaseline time points.
`covar1`	The covariance matrix for the repeated measures given baseline for the active treatment group.
`covar2`	The covariance matrix for the repeated measures given baseline for the control group. If missing, it will be set equal to the covariance matrix for the active treatment group.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`gamma1`	The hazard rate for exponential dropout, or a vector of hazard rates for piecewise exponential dropout for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, or a vector of hazard rates for piecewise exponential dropout for the control group.
`accrualDuration`	Duration of the enrollment period.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The degrees of freedom for the t-distribution is the total effective sample size minus 2. The exact calculation using the t distribution is only implemented for the fixed design.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. 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.
`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`.

Details

Consider a longitudinal study with two treatment groups. The outcome is measured at baseline and at k postbaseline time points. For each treatment group, the outcomes are assumed to follow a multivariate normal distribution. Conditional on baseline, the covariance matrix of the post-baseline outcomes is denoted by Σ_1 for the active treatment group and Σ_2 for the control group. Let μ_1 and μ_2 denote the mean vectors of post-baseline outcomes for the active and control groups, respectively. We are interested in testing the null hypothesis $H_0 : \mu_{1,k} - \mu_{2,k} = \delta_0$ against the alternative $H_1 : \mu_{1,k} - \mu_{2,k} = \delta$.

The study design is based on the information for treatment difference at the last postbaseline time point. This information is given by

$$I = 1/\text{Var}(\hat{\mu}_{1,k} - \hat{\mu}_{2,k})$$

In the presence of monotone missing data, let $p_{g,1},\ldots,p_{g,k}$ denote the proportions of subjects in observed data patterns 1 through k for treatment group $g=1$ (active) or 2 (control). A subject in pattern j has complete data up to time t_j, i.e., the observed outcomes are $y_{i,1},\ldots,y_{i,j}$, with missing values for $y_{i,j+1},\ldots,y_{i,k}$.

According to Lu et al. (2008), the information matrix for the post-baseline mean vector in group g is

$$I_g = n\pi_g J_g$$

where π_g is the proportion of subjects in group g, and

$$J_g = \sum_{j=1}^{k} p_{g,j} \left(\begin{array}{cc} \Sigma_{g,j}^{-1} & 0 \\ 0 & 0 \end{array} \right)$$

Here, $\Sigma_{g,j}$ is the leading $j \times j$ principal submatrix of Σ_g. It follows that

$$\text{Var}(\hat{\mu}_{1,k} - \hat{\mu}_{2,k}) = \frac{1}{n}\left(\frac{1}{\pi_1}J_1^{-1}[k,k] + \frac{1}{\pi_2}J_2^{-1}[k,k]\right)$$

The observed data pattern probabilities depend on the accrual and dropout distributions. Let $H(u)$ denote the cumulative distribution function of enrollment time u, $G_g(t)$ denote the survival function of dropout time t for treatment group g, and τ denote the calendar time at interim or final analysis. Then, for $j=1,\ldots,k-1$, the probability that a subject in group g falls into observed data pattern j is

$$p_{g,j} = H(\tau - t_j)G_g(t_j) - H(\tau - t_{j+1})G_g(t_{j+1})$$

For the last pattern ($j=k$, i.e., completers),

$$p_{g,k} = H(\tau - t_k)G_g(t_k)$$

For the final analysis, τ is the study duration, so $H(\tau - t_j) = 1$ for all j. Therefore, the pattern probabilities depend only on the dropout distribution:

$$p_{g,j} = G_g(t_j) - G_g(t_{j+1}), \quad j = 1,\ldots,k-1$$

and

$$p_{g,k} = G_g(t_k)$$

Cumulative dropout probabilities at post-baseline time points can be used to define a piecewise exponential dropout distribution. Let $F_g(t_j)$ denote the cumulative probability of dropout by time t_j for treatment group g. The left endpoints of the piecewise survival time intervals are given by $t_0 = 0, t_1, \ldots, t_{k-1}$. The hazard rate in the interval $(t_{j-1}, t_j]$ is given by

$$\gamma_{g,j} = -\log\left(\frac{1-F_g(t_j)}{1-F_g(t_{j-1})}\right)/(t_j - t_{j-1}), \quad j = 1,\ldots,k$$

Value

An S3 class `designMeanDiffMMRM` 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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `studyDuration`: The maximum study duration.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `expectedStudyDurationH1`: The expected study duration under H1.
  - `expectedStudyDurationH0`: The expected study duration under H0.
  - `accrualDuration`: The accrual duration.
  - `followupTime`: The follow-up time.
  - `fixedFollowup`: Whether a fixed follow-up design is used.
  - `meanDiffH0`: The mean difference under H0.
  - `meanDiff`: The mean difference under H1.
- `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.
  - `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.
  - `efficacyMeanDiff`: The efficacy boundaries on the mean difference scale.
  - `futilityMeanDiff`: The futility boundaries on the mean difference scale.
  - `numberOfSubjects`: The number of subjects.

  - `numberOfCompleters`: The number of completers.
  - `analysisTime`: The average time since trial start.
- `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.
  - `allocationRatioPlanned`: The allocation ratio for the active treatment versus control.
  - `accrualTime`: A vector that specifies the starting time of piecewise Poisson enrollment time intervals.
  - `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.
  - `gamma1`: The hazard rate for exponential dropout or a vector of hazard rates for piecewise exponential dropout for the active treatment group.
  - `gamma2`: The hazard rate for exponential dropout or a vector of hazard rates for piecewise exponential dropout for the control group.
  - `k`: The number of postbaseline time points.
  - `t`: The postbaseline time points.
  - `covar1`: The covariance matrix for the repeated measures given baseline for the active treatment group.
  - `covar2`: The covariance matrix for the repeated measures given baseline for the control group.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

Kaifeng Lu, Xiaohui Luo, and Pei-Yun Chen. Sample size estimation for repeated measures analysis in randomized clinical trials with missing data. The International Journal of Biostatistics 2008; 14(1), Article 9.

Examples

```
# function to generate the AR(1) correlation matrix
ar1_cor <- function(n, corr) {
  exponent <- abs(matrix((1:n) - 1, n, n, byrow = TRUE) - ((1:n) - 1))
  corr^exponent
```

```
}

(design1 = getDesignMeanDiffMMRM(
  beta = 0.2,
  meanDiffH0 = 0,
  meanDiff = 0.5,
  k = 4,
  t = c(1,2,3,4),
  covar1 = ar1_cor(4, 0.7),
  accrualIntensity = 10,
  gamma1 = 0.02634013,
  gamma2 = 0.02634013,
  accrualDuration = NA,
  allocationRatioPlanned = 1,
  kMax = 3,
  alpha = 0.025,
  typeAlphaSpending = "sfOF"))
```

`getDesignMeanDiffXO` *Group Sequential Design for Mean Difference in 2x2 Crossover*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for two-sample mean difference in 2x2 crossover.

Usage

```
getDesignMeanDiffXO(
  beta = NA_real_,
  n = NA_real_,
  meanDiffH0 = 0,
  meanDiff = 0.5,
  stDev = 1,
  allocationRatioPlanned = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
```

```
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`meanDiffH0`	The mean difference under the null hypothesis. Defaults to 0.
`meanDiff`	The mean difference under the alternative hypothesis.
`stDev`	The standard deviation for within-subject random error.
`allocationRatioPlanned`	Allocation ratio for sequence A/B versus sequence B/A. Defaults to 1 for equal randomization.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.

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

Value

An S3 class `designMeanDiffXO` 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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `meanDiffH0`: The mean difference under the null hypothesis.
  - `meanDiff`: The mean difference under the alternative hypothesis.
  - `stDev`: The standard deviation for within-subject random error.
- `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.
  - `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.
  - `efficacyMeanDiff`: The efficacy boundaries on the mean difference scale.
  - `futilityMeanDiff`: The futility boundaries on the mean difference scale.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `allocationRatioPlanned`: Allocation ratio for sequence A/B versus sequence B/A.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignMeanDiffXO(
  beta = 0.2, n = NA, meanDiff = 75, stDev = 150,
  normalApproximation = FALSE, alpha = 0.05))
```

---

getDesignMeanDiffXOEquiv

Group Sequential Design for Equivalence in Mean Difference in 2x2 Crossover

---

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for equivalence in mean difference in 2x2 crossover.

Usage

```
getDesignMeanDiffXOEquiv(
  beta = NA_real_,
  n = NA_real_,
  meanDiffLower = NA_real_,
  meanDiffUpper = NA_real_,
  meanDiff = 0,
  stDev = 1,
  allocationRatioPlanned = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`meanDiffLower`	The lower equivalence limit of mean difference.
`meanDiffUpper`	The upper equivalence limit of mean difference.
`meanDiff`	The mean difference under the alternative hypothesis.
`stDev`	The standard deviation for within-subject random error.
`allocationRatioPlanned`	Allocation ratio for sequence A/B versus sequence B/A. Defaults to 1 for equal randomization.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.

`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.

Value

An S3 class `designMeanDiffXOEquiv` 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.
    - `kMax`: The number of stages.
    - `information`: The maximum information.
    - `expectedInformationH1`: The expected information under H1.
    - `expectedInformationH0`: The expected information under H0.
    - `numberOfSubjects`: The maximum number of subjects.
    - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
    - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
    - `meanDiffLower`: The lower equivalence limit of mean difference.
    - `meanDiffUpper`: The upper equivalence limit of mean difference.
    - `meanDiff`: The mean difference under the alternative hypothesis.
    - `stDev`: The standard deviation for within-subject random error.
- `byStageResults`: A data frame containing the following variables:
    - `informationRates`: The information rates.
    - `efficacyBounds`: The efficacy boundaries on the Z-scale for each of the two one-sided tests.

  - `rejectPerStage`: The probability for efficacy stopping.
  - `cumulativeRejection`: The cumulative probability for efficacy stopping.
  - `cumulativeAlphaSpent`: The cumulative alpha for each of the two one-sided tests.
  - `cumulativeAttainedAlpha`: The cumulative probability for efficacy stopping under H0.
  - `efficacyMeanDiffLower`: The efficacy boundaries on the mean difference scale for the one-sided null hypothesis on the lower equivalence limit.
  - `efficacyMeanDiffUpper`: The efficacy boundaries on the mean difference scale for the one-sided null hypothesis on the upper equivalence limit.
  - `efficacyP`: The efficacy bounds on the p-value scale for each of the two one-sided tests.
  - `information`: The cumulative information.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `spendingTime`: The error spending time at each analysis.
  - `allocationRatioPlanned`: Allocation ratio for sequence A/B versus sequence B/A.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Example 1: group sequential trial power calculation
(design1 <- getDesignMeanDiffXOEquiv(
  beta = 0.1, n = NA, meanDiffLower = -1.3, meanDiffUpper = 1.3,
  meanDiff = 0, stDev = 2.2,
  kMax = 4, alpha = 0.05, typeAlphaSpending = "sfOF"))

# Example 2: sample size calculation for t-test
(design2 <- getDesignMeanDiffXOEquiv(
  beta = 0.1, n = NA, meanDiffLower = -1.3, meanDiffUpper = 1.3,
  meanDiff = 0, stDev = 2.2,
  normalApproximation = FALSE, alpha = 0.05))
```

`getDesignMeanRatio` *Group Sequential Design for Two-Sample Mean Ratio*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for two-sample mean ratio.

Usage

```
getDesignMeanRatio(
  beta = NA_real_,
  n = NA_real_,
  meanRatioH0 = 1,
  meanRatio = 1.25,
  CV = 1,
  allocationRatioPlanned = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`meanRatioH0`	The mean ratio under the null hypothesis. Defaults to 1.
`meanRatio`	The mean ratio under the alternative hypothesis.
`CV`	The coefficient of variation. The standard deviation on the log scale is equal to `sqrt(log(1 + CV^2))`.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.

`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.

Value

An S3 class `designMeanRatio` 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*sqrt(information)`.
    - `inflationFactor`: The inflation factor (relative to the fixed design).
    - `numberOfSubjects`: The maximum number of subjects.
    - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
    - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
    - `meanRatioH0`: The mean ratio under the null hypothesis.
    - `meanRatio`: The mean ratio under the alternative hypothesis.
    - `CV`: The coefficient of variation.
- `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.
    - `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.
    - `numberOfSubjects`: The number of subjects.
    - `efficacyMeanRatio`: The efficacy boundaries on the mean ratio scale.
    - `futilityMeanRatio`: The futility boundaries on the mean ratio scale.

- `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.
  - `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignMeanRatio(
  beta = 0.1, n = NA, meanRatio = 1.25, CV = 0.25,
  alpha = 0.05, normalApproximation = FALSE))
```

`getDesignMeanRatioEquiv`

Group Sequential Design for Equivalence in Two-Sample Mean Ratio

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for equivalence in two-sample mean ratio.

Usage

```
getDesignMeanRatioEquiv(
  beta = NA_real_,
  n = NA_real_,
  meanRatioLower = NA_real_,
  meanRatioUpper = NA_real_,
  meanRatio = 1,
  CV = 1,
  allocationRatioPlanned = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
```

```
  informationRates = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`meanRatioLower`	The lower equivalence limit of mean ratio.
`meanRatioUpper`	The upper equivalence limit of mean ratio.
`meanRatio`	The mean ratio under the alternative hypothesis.
`CV`	The coefficient of variation.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.

Value

An S3 class `designMeanRatioEquiv` object with three components:

- `overallResults`: A data frame containing the following variables:
    - `overallReject`: The overall rejection probability.
    - `alpha`: The significance level for each of the two one-sided tests. Defaults to 0.05.
    - `attainedAlpha`: The attained significance level.
    - `kMax`: The number of stages.
    - `information`: The maximum information.
    - `expectedInformationH1`: The expected information under H1.
    - `expectedInformationH0`: The expected information under H0.
    - `numberOfSubjects`: The maximum number of subjects.
    - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
    - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
    - `meanRatioLower`: The lower equivalence limit of mean ratio.
    - `meanRatioUpper`: The upper equivalence limit of mean ratio.
    - `meanRatio`: The mean ratio under the alternative hypothesis.
    - `CV`: The coefficient of variation.
- `byStageResults`: A data frame containing the following variables:
    - `informationRates`: The information rates.
    - `efficacyBounds`: The efficacy boundaries on the Z-scale for each of the two one-sided tests.
    - `rejectPerStage`: The probability for efficacy stopping.
    - `cumulativeRejection`: The cumulative probability for efficacy stopping.
    - `cumulativeAlphaSpent`: The cumulative alpha for each of the two one-sided tests.
    - `cumulativeAttainedAlpha`: The cumulative probability for efficacy stopping under H0.
    - `efficacyP`: The efficacy bounds on the p-value scale for each of the two one-sided tests.
    - `information`: The cumulative information.
    - `numberOfSubjects`: The number of subjects.
    - `efficacyMeanRatioLower`: The efficacy boundaries on the mean ratio scale for the one-sided null hypothesis on the lower equivalence limit.
    - `efficacyMeanRatioUpper`: The efficacy boundaries on the mean ratio scale for the one-sided null hypothesis on the upper equivalence limit.
- `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.
    - `spendingTime`: The error spending time at each analysis.
    - `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
    - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
    - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

Examples

```
# Example 1: group sequential trial power calculation
(design1 <- getDesignMeanRatioEquiv(
  beta = 0.1, n = NA, meanRatioLower = 0.8, meanRatioUpper = 1.25,
  meanRatio = 1, CV = 0.35,
  kMax = 4, alpha = 0.05, typeAlphaSpending = "sfOF"))

# Example 2: sample size calculation for t-test
(design2 <- getDesignMeanRatioEquiv(
  beta = 0.1, n = NA, meanRatioLower = 0.8, meanRatioUpper = 1.25,
  meanRatio = 1, CV = 0.35,
  normalApproximation = FALSE, alpha = 0.05))
```

`getDesignMeanRatioXO` *Group Sequential Design for Mean Ratio in 2x2 Crossover*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for two-sample mean ratio in 2x2 crossover.

Usage

```
getDesignMeanRatioXO(
  beta = NA_real_,
  n = NA_real_,
  meanRatioH0 = 1,
  meanRatio = 1.25,
  CV = 1,
  allocationRatioPlanned = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
```

```
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`meanRatioH0`	The mean ratio under the null hypothesis. Defaults to 1.
`meanRatio`	The mean ratio under the alternative hypothesis.
`CV`	The coefficient of variation. The standard deviation on the log scale is equal to `sqrt(log(1 + CV^2))`.
`allocationRatioPlanned`	Allocation ratio for sequence A/B versus sequence B/A. Defaults to 1 for equal randomization.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.

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

Value

An S3 class `designMeanRatioXO` 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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `meanRatioH0`: The mean ratio under the null hypothesis.
  - `meanRatio`: The mean ratio under the alternative hypothesis.
  - `CV`: The coefficient of variation.
- `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.
  - `efficacyMeanRatio`: The efficacy boundaries on the mean ratio scale.
  - `futilityMeanRatio`: The futility boundaries on the mean ratio 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.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `allocationRatioPlanned`: Allocation ratio for sequence A/B versus sequence B/A.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignMeanRatioXO(
  beta = 0.1, n = NA, meanRatio = 1.25, CV = 0.25,
  alpha = 0.05, normalApproximation = FALSE))
```

---

getDesignMeanRatioXOEquiv

Group Sequential Design for Equivalence in Mean Ratio in 2x2 Crossover

---

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for equivalence mean ratio in 2x2 crossover.

Usage

```
getDesignMeanRatioXOEquiv(
  beta = NA_real_,
  n = NA_real_,
  meanRatioLower = NA_real_,
  meanRatioUpper = NA_real_,
  meanRatio = 1,
  CV = 1,
  allocationRatioPlanned = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`meanRatioLower`	The lower equivalence limit of mean ratio.
`meanRatioUpper`	The upper equivalence limit of mean ratio.
`meanRatio`	The mean ratio under the alternative hypothesis.
`CV`	The coefficient of variation.
`allocationRatioPlanned`	Allocation ratio for sequence A/B versus sequence B/A. Defaults to 1 for equal randomization.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.

`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.

Value

An S3 class `designMeanRatioEquiv` 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.
  - `kMax`: The number of stages.
  - `information`: The maximum information.
  - `expectedInformationH1`: The expected information under H1.
  - `expectedInformationH0`: The expected information under H0.
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `meanRatioLower`: The lower equivalence limit of mean ratio.
  - `meanRatioUpper`: The upper equivalence limit of mean ratio.
  - `meanRatio`: The mean ratio under the alternative hypothesis.
  - `CV`: The coefficient of variation.
- `byStageResults`: A data frame containing the following variables:
  - `informationRates`: The information rates.
  - `efficacyBounds`: The efficacy boundaries on the Z-scale for each of the two one-sided tests.

  - `rejectPerStage`: The probability for efficacy stopping.
  - `cumulativeRejection`: The cumulative probability for efficacy stopping.
  - `cumulativeAlphaSpent`: The cumulative alpha for each of the two one-sided tests.
  - `cumulativeAttainedAlpha`: The cumulative probability for efficacy stopping under H0.
  - `efficacyMeanRatioLower`: The efficacy boundaries on the mean ratio scale for the one-sided null hypothesis on the lower equivalence limit.
  - `efficacyMeanRatioUpper`: The efficacy boundaries on the mean ratio scale for the one-sided null hypothesis on the upper equivalence limit.
  - `efficacyP`: The efficacy bounds on the p-value scale for each of the two one-sided tests.
  - `information`: The cumulative information.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `spendingTime`: The error spending time at each analysis.
  - `allocationRatioPlanned`: Allocation ratio for sequence A/B versus sequence B/A.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Example 1: group sequential trial power calculation
(design1 <- getDesignMeanRatioXOEquiv(
  beta = 0.1, n = NA, meanRatioLower = 0.8, meanRatioUpper = 1.25,
  meanRatio = 1, CV = 0.35,
  kMax = 4, alpha = 0.05, typeAlphaSpending = "sfOF"))

# Example 2: sample size calculation for t-test
(design2 <- getDesignMeanRatioXOEquiv(
  beta = 0.1, n = NA, meanRatioLower = 0.8, meanRatioUpper = 1.25,
  meanRatio = 1, CV = 0.35,
  normalApproximation = FALSE, alpha = 0.05))
```

`getDesignOddsRatio` *Group Sequential Design for Two-Sample Odds Ratio*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for two-sample odds ratio.

Usage

```
getDesignOddsRatio(
  beta = NA_real_,
  n = NA_real_,
  oddsRatioH0 = 1,
  pi1 = NA_real_,
  pi2 = NA_real_,
  nullVariance = FALSE,
  allocationRatioPlanned = 1,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`oddsRatioH0`	The odds ratio under the null hypothesis. Defaults to 1.
`pi1`	The assumed probability for the active treatment group.
`pi2`	The assumed probability for the control group.
`nullVariance`	Whether to use the variance under the null or the empirical variance under the alternative.

`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.

Details

Consider a group sequential design for two-sample odds ratio. The parameter of interest is

$$\psi = \frac{\pi_1(1-\pi_2)}{(1-\pi_1)\pi_2}$$

where π_1 is the response probability for the active treatment group and π_2 is the response probability for the control group. For statistical inference, the parameter is often transformed to the log scale:

$$\theta = \log(\psi) = \log(\pi_1/(1-\pi_1)) - \log(\pi_2/(1-\pi_2))$$

The variance of the estimator $\hat{\theta}$ can be derived from the binomial distributions as follows:

$$Var(\hat{\theta}) = \frac{1}{n}\{\frac{1}{\pi_1(1-\pi_1)r} + \frac{1}{\pi_2(1-\pi_2)(1-r)}\}$$

where n is the total number of subjects and r is the randomization probability for the active treatment group. When `nullVariance = TRUE`, the variance is computed under the null hypothesis. In this case, the values of π_1 and π_2 in the variance formula are replaced with their restricted maximum likelihood counterparts, subject to the constraint

$$\frac{\pi_1(1-\pi_2)}{(1-\pi_1)\pi_2} = \psi_0$$

Value

An S3 class `designOddsRatio` 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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `oddsRatioH0`: The odds ratio under the null hypothesis.
  - `pi1`: The assumed probability for the active treatment group.
  - `pi2`: The assumed probability for the control group.
- `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.
  - `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.
  - `efficacyOdddsRatio`: The efficacy boundaries on the odds ratio scale.
  - `futilityOddsRatio`: The futility boundaries on the odds ratio scale.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `nullVariance`: Whether to use the variance under the null or the empirical variance under the alternative.
  - `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignOddsRatio(
  beta = 0.1, n = NA, pi1 = 0.5, pi2 = 0.3,
  alpha = 0.05))
```

`getDesignOddsRatioEquiv` *Group Sequential Design for Equivalence in Two-Sample Odds Ratio*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for equivalence in two-sample odds ratio.

Usage

```
getDesignOddsRatioEquiv(
  beta = NA_real_,
  n = NA_real_,
  oddsRatioLower = NA_real_,
  oddsRatioUpper = NA_real_,
  pi1 = NA_real_,
  pi2 = NA_real_,
  allocationRatioPlanned = 1,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  criticalValues = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`oddsRatioLower`	The lower equivalence limit of odds ratio.
`oddsRatioUpper`	The upper equivalence limit of odds ratio.
`pi1`	The assumed probability for the active treatment group.
`pi2`	The assumed probability for the control group.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.

`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.

Value

An S3 class `designOddsRatioEquiv` object with three components:

- `overallResults`: A data frame containing the following variables:
    - `overallReject`: The overall rejection probability.
    - `alpha`: The significance level for each of the two one-sided tests. Defaults to 0.05.
    - `attainedAlphaH10`: The attained significance level under H10.
    - `attainedAlphaH20`: The attained significance level under H20.
    - `kMax`: The number of stages.
    - `information`: The maximum information.
    - `expectedInformationH1`: The expected information under H1.
    - `expectedInformationH10`: The expected information under H10.
    - `expectedInformationH20`: The expected information under H20.
    - `numberOfSubjects`: The maximum number of subjects.
    - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
    - `expectedNumberOfSubjectsH10`: The expected number of subjects under H10.
    - `expectedNumberOfSubjectsH20`: The expected number of subjects under H20.
    - `oddsRatioLower`: The lower equivalence limit of odds ratio.
    - `oddsRatioUpper`: The upper equivalence limit of odds ratio.
    - `pi1`: The assumed probability for the active treatment group.
    - `pi2`: The assumed probability for the control group.
    - `oddsRatio`: The odds ratio.
- `byStageResults`: A data frame containing the following variables:

  - `informationRates`: The information rates.
  - `efficacyBounds`: The efficacy boundaries on the Z-scale for each of the two one-sided tests.
  - `rejectPerStage`: The probability for efficacy stopping.
  - `cumulativeRejection`: The cumulative probability for efficacy stopping.
  - `cumulativeAlphaSpent`: The cumulative alpha for each of the two one-sided tests.
  - `cumulativeAttainedAlphaH10`: The cumulative alpha attained under H10.
  - `cumulativeAttainedAlphaH20`: The cumulative alpha attained under H20.
  - `efficacyOddsRatioLower`: The efficacy boundaries on the odds ratio scale for the one-sided null hypothesis on the lower equivalence limit.
  - `efficacyOddsRatioUpper`: The efficacy boundaries on the odds ratio scale for the one-sided null hypothesis on the upper equivalence limit.
  - `efficacyP`: The efficacy bounds on the p-value scale for each of the two one-sided tests.
  - `information`: The cumulative information.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `spendingTime`: The error spending time at each analysis.
  - `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignOddsRatioEquiv(
  beta = 0.2, n = NA, oddsRatioLower = 0.8,
  oddsRatioUpper = 1.25, pi1 = 0.12, pi2 = 0.12,
  kMax = 3, alpha = 0.05, typeAlphaSpending = "sfOF"))
```

`getDesignOneMean` *Group Sequential Design for One-Sample Mean*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for one-sample mean.

Usage

```
getDesignOneMean(
  beta = NA_real_,
  n = NA_real_,
  meanH0 = 0,
  mean = 0.5,
  stDev = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`meanH0`	The mean under the null hypothesis. Defaults to 0.
`mean`	The mean under the alternative hypothesis.
`stDev`	The standard deviation.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.

`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.

Value

An S3 class `designOneMean` 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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `meanH0`: The mean under the null hypothesis.
  - `mean`: The mean under the alternative hypothesis.
  - `stDev`: The standard deviation.
- `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.
  - `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.
  - `efficacyMean`: The efficacy boundaries on the mean scale.
  - `futilityMean`: The futility boundaries on the mean scale.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Example 1: group sequential trial power calculation
(design1 <- getDesignOneMean(
  beta = 0.1, n = NA, meanH0 = 7, mean = 6, stDev = 2.5,
  kMax = 5, alpha = 0.025, typeAlphaSpending = "sfOF",
  typeBetaSpending = "sfP"))

# Example 2: sample size calculation for one-sample t-test
(design2 <- getDesignOneMean(
  beta = 0.1, n = NA, meanH0 = 7, mean = 6, stDev = 2.5,
  normalApproximation = FALSE, alpha = 0.025))
```

`getDesignOneMultinom` *Power and Sample Size for One-Sample Multinomial Response*

Description

Obtains the power given sample size or obtains the sample size given power for one-sample multinomial response.

Usage

```
getDesignOneMultinom(
  beta = NA_real_,
  n = NA_real_,
  ncats = NA_integer_,
  piH0 = NA_real_,
  pi = NA_real_,
  rounding = TRUE,
  alpha = 0.05
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`ncats`	The number of categories of the multinomial response.
`piH0`	The prevalence of each category under the null hypothesis. Only need to provide the values for the first `ncats-1` categories.
`pi`	The prevalence of each category. Only need to provide the values for the first `ncats-1` categories.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`alpha`	The two-sided significance level. Defaults to 0.05.

Details

A single-arm multinomial response design is used to test whether the prevalence of each category is different from the null hypothesis prevalence. The null hypothesis is that the prevalence of each category is equal to π_{0i}, while the alternative hypothesis is that the prevalence of each category is equal to π_i, for $i = 1, \ldots, C$, where C is the number of categories.

The sample size is calculated based on the chi-square test for multinomial response. The test statistic is given by

$$X^2 = \sum_{i=1}^{C} \frac{(n_i - n\pi_{0i})^2}{n\pi_{0i}}$$

where n_i is the number of subjects in category i, and n is the total sample size.

- Under the null hypothesis, X^2 follows a chi-square distribution with $C-1$ degrees of freedom.
- Under the alternative hypothesis, X^2 follows a non-central chi-square distribution with non-centrality parameter
$$\lambda = n \sum_{i=1}^{C} \frac{(\pi_i - \pi_{0i})^2}{\pi_{0i}}$$

The sample size is chosen such that the power to reject the null hypothesis is at least $1 - \beta$ for a given significance level α.

Value

An S3 class `designOneMultinom` object with the following components:

- `power`: The power to reject the null hypothesis.
- `alpha`: The two-sided significance level.
- `n`: The maximum number of subjects.
- `ncats`: The number of categories of the multinomial response.
- `piH0`: The prevalence of each category under the null hypothesis.
- `pi`: The prevalence of each category.
- `effectsize`: The effect size for the chi-square test.
- `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignOneMultinom(
  beta = 0.1, ncats = 3, piH0 = c(0.25, 0.25),
  pi = c(0.3, 0.4), alpha = 0.05))
```

---

getDesignOneProportion *Group Sequential Design for One-Sample Proportion*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for one-sample proportion.

Usage

```
getDesignOneProportion(
  beta = NA_real_,
  n = NA_real_,
  piH0 = 0.1,
  pi = 0.2,
  nullVariance = TRUE,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`piH0`	The response probability under the null hypothesis.
`pi`	The response probability under the alternative hypothesis.
`nullVariance`	Whether to use the variance under the null or the variance under the alternative.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the normal approximation will be used, otherwise the calculations are performed with the binomial distribution. The exact calculation using the binomial distribution is only implemented for the fixed design.

`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.

Value

An S3 class `designOneProportion` 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 as well as for the binomial exact test in a fixed design.
  - `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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `piH0`: The response probability under the null hypothesis.
  - `pi`: The response probability under the alternative hypothesis.
- `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.
  - `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.
  - `efficacyResponses`: The efficacy boundaries on the number of responses scale.
  - `futilityResponses`: The futility boundaries on the number of responses scale.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `nullVariance`: Whether to use the variance under the null or the empirical variance under the alternative.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the binomial distribution.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Example 1: group sequential trial power calculation
(design1 <- getDesignOneProportion(
  beta = 0.2, n = NA, piH0 = 0.15, pi = 0.25,
  kMax = 3, alpha = 0.05, typeAlphaSpending = "sfOF"))

# Example 2: sample size calculation for one-sample binomial exact test
(design2 <- getDesignOneProportion(
  beta = 0.2, n = NA, piH0 = 0.15, pi = 0.25,
  normalApproximation = FALSE, alpha = 0.05))
```

`getDesignOneRateExact` *Power and Sample Size for One-Sample Poisson Rate Exact Test*

Description

Obtains the power given sample size or obtains the sample size given power for one-sample Poisson rate.

Usage

```
getDesignOneRateExact(
  beta = NA_real_,
  n = NA_real_,
  lambdaH0 = NA_real_,
  lambda = NA_real_,
  D = 1,
  alpha = 0.025
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`lambdaH0`	The Poisson rate under the null hypothesis.
`lambda`	The Poisson rate under the alternative hypothesis.
`D`	The average exposure per subject.
`alpha`	The one-sided significance level. Defaults to 0.025.

Value

A data frame containing the following variables:

- `alpha`: The specified significance level.
- `attainedAlpha`: The attained type I error of the exact test.
- `power`: The actual power of the exact test.
- `n`: The sample size.
- `lambdaH0`: The Poisson rate under the null hypothesis.
- `lambda`: The Poisson rate under the alternative hypothesis.
- `D`: The average exposure per subject.
- `r`: The critical value of the number of events for rejecting the null hypothesis. Reject H0 if `Y` >= `r` for upper-tailed test, and reject H0 if `Y` <= `r` for lower-tailed test.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Example 1: power calculation
(design1 <- getDesignOneRateExact(
  n = 525, lambdaH0 = 0.049, lambda = 0.012,
  D = 0.5, alpha = 0.025))

# Example 2: sample size calculation
(design2 <- getDesignOneRateExact(
  beta = 0.2, lambdaH0 = 0.2, lambda = 0.3,
  D = 1, alpha = 0.05))
```

`getDesignOneSlope` *Group Sequential Design for One-Sample Slope*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for one-sample slope.

Usage

```
getDesignOneSlope(
  beta = NA_real_,
  n = NA_real_,
  slopeH0 = 0,
  slope = 0.5,
  stDev = 1,
  stDevCovariate = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`slopeH0`	The slope under the null hypothesis. Defaults to 0.
`slope`	The slope under the alternative hypothesis.
`stDev`	The standard deviation of the residual.
`stDevCovariate`	The standard deviation of the covariate.

`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.

Details

We assume a simple linear regression of the form

$$y_i = \alpha + \beta x_i + \epsilon_i$$

where ϵ_i is the residual error, which is assumed to be normally distributed with mean 0 and standard deviation σ_ϵ. The covariate x_i is assumed to be normally distributed with mean 0 and standard deviation σ_x. The slope under the null hypothesis is β_0, and the slope under the alternative hypothesis is β. Since

$$\hat{\beta} = \frac{\sum_{i=1}^n (x_i - \bar{x}) y_i}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

it follows that

$$\hat{\beta} \sim N(\beta, \frac{\sigma_\epsilon^2}{\sum_{i=1}^n (x_i - \bar{x})^2}).$$

Since the variance of $\hat{\beta}$ is

$$\frac{\sigma_\epsilon^2}{n\sigma_x^2}$$

we can use it to calculate the power and sample size for the group sequential design.

Value

An S3 class `designOneSlope` 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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `slopeH0`: The slope under the null hypothesis.
  - `slope`: The slope under the alternative hypothesis.
  - `stDev`: The standard deviation of the residual.
  - `stDevCovariate`: The standard deviation of the covariate.
- `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.
  - `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.
  - `efficacySlope`: The efficacy boundaries on the slope scale.
  - `futilitySlope`: The futility boundaries on the slope scale.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignOneSlope(
  beta = 0.1, n = NA, slope = 0.5,
  stDev = 15, stDevCovariate = 9,
  normalApproximation = FALSE,
  alpha = 0.025))
```

`getDesignOrderedBinom` *Power and Sample Size for Cochran-Armitage Trend Test for Ordered Multi-Sample Binomial Response*

Description

Obtains the power given sample size or obtains the sample size given power for the Cochran-Armitage trend test for ordered multi-sample binomial response.

Usage

```
getDesignOrderedBinom(
  beta = NA_real_,
  n = NA_real_,
  ngroups = NA_integer_,
  pi = NA_real_,
  w = NA_real_,
  allocationRatioPlanned = NA_integer_,
  rounding = TRUE,
  alpha = 0.05
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`ngroups`	The number of treatment groups.
`pi`	The response probabilities for the treatment groups.
`w`	The scores assigned to the treatment groups. This should reflect the ordinal nature of the treatment groups, e.g. dose levels. Defaults to equally spaced scores.
`allocationRatioPlanned`	Allocation ratio for the treatment groups.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`alpha`	The two-sided significance level. Defaults to 0.05.

Details

An ordered multi-sample binomial response design is used to test whether the response probabilities differ across multiple treatment groups. The null hypothesis is that the response probabilities are equal across all treatment groups, while the alternative hypothesis is that the response probabilities are ordered, i.e. the response probability increases with the treatment group index. The Cochran-Armitage trend test is used to test this hypothesis. This test effectively regresses the response probabilities against treatment group scores, and test whether the slope of the regression line is significantly different from zero.

The trend parameter is defined as

$$\theta = \sum_{g=1}^{G} r_g (w_g - \bar{w}) \pi_g$$

where G is the number of treatment groups, r_g is the randomization probability for treatment group g, w_g is the score assigned to treatment group g, π_g is the response probability for treatment group g, and $\bar{w} = \sum_{g=1}^{G} r_g w_g$ is the weighted average score across all treatment groups.

Since $\hat{\theta}$ is a linear combination of the estimated response probabilities, its variance is given by

$$Var(\hat{\theta}) = \frac{1}{n} \sum_{g=1}^{G} r_g (w_g - \bar{w})^2 \pi_g (1 - \pi_g)$$

where n is the total sample size.

The sample size is chosen such that the power to reject the null hypothesis is at least $1 - \beta$ for a given significance level α.

Value

An S3 class `designOrderedBinom` object with the following components:

- `power`: The power to reject the null hypothesis.
- `alpha`: The two-sided significance level.
- `n`: The maximum number of subjects.
- `ngroups`: The number of treatment groups.
- `pi`: The response probabilities for the treatment groups.
- `w`: The scores assigned to the treatment groups.
- `trendstat`: The Cochran-Armitage trend test statistic.
- `allocationRatioPlanned`: Allocation ratio for the treatment groups.
- `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignOrderedBinom(
  beta = 0.1, ngroups = 3, pi = c(0.1, 0.25, 0.5), alpha = 0.05))
```

`getDesignPairedMeanDiff`

Group Sequential Design for Paired Mean Difference

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for paired mean difference.

Usage

```
getDesignPairedMeanDiff(
  beta = NA_real_,
  n = NA_real_,
  pairedDiffH0 = 0,
  pairedDiff = 0.5,
  stDev = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`pairedDiffH0`	The paired difference under the null hypothesis. Defaults to 0.
`pairedDiff`	The paired difference under the alternative hypothesis.
`stDev`	The standard deviation for paired difference.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.

`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.

Value

An S3 class `designPairedMeanDiff` 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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `pairedDiffH0`: The paired difference under the null hypothesis.
  - `pairedDiff`: The paired difference under the alternative hypothesis.
  - `stDev`: The standard deviation for paired difference.
- `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.
  - `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.
  - `efficacyPairedDiff`: The efficacy boundaries on the paired difference scale.
  - `futilityPairedDiff`: The futility boundaries on the paired difference scale.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Example 1: group sequential trial power calculation
(design1 <- getDesignPairedMeanDiff(
  beta = 0.1, n = NA, pairedDiffH0 = 0, pairedDiff = -2, stDev = 5,
  kMax = 5, alpha = 0.05, typeAlphaSpending = "sfOF"))

# Example 2: sample size calculation for one-sample t-test
(design2 <- getDesignPairedMeanDiff(
  beta = 0.1, n = NA, pairedDiffH0 = 0, pairedDiff = -2, stDev = 5,
  normalApproximation = FALSE, alpha = 0.025))
```

`getDesignPairedMeanDiffEquiv`

Group Sequential Design for Equivalence in Paired Mean Difference

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for equivalence in paired mean difference.

Usage

```
getDesignPairedMeanDiffEquiv(
  beta = NA_real_,
  n = NA_real_,
  pairedDiffLower = NA_real_,
  pairedDiffUpper = NA_real_,
  pairedDiff = 0,
  stDev = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
```

```
  kMax = 1L,
  informationRates = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`pairedDiffLower`	The lower equivalence limit of paired difference.
`pairedDiffUpper`	The upper equivalence limit of paired difference.
`pairedDiff`	The paired difference under the alternative hypothesis.
`stDev`	The standard deviation for paired difference.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.

Value

An S3 class `designPairedMeanDiffEquiv` object with three components:

- `overallResults`: A data frame containing the following variables:
    - `overallReject`: The overall rejection probability.
    - `alpha`: The significance level for each of the two one-sided tests. Defaults to 0.05.
    - `attainedAlpha`: The attained significance level under H0.
    - `kMax`: The number of stages.
    - `information`: The maximum information.
    - `expectedInformationH1`: The expected information under H1.
    - `expectedInformationH0`: The expected information under H0.
    - `numberOfSubjects`: The maximum number of subjects.
    - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
    - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
    - `pairedDiffLower`: The lower equivalence limit of paired difference.
    - `pairedDiffUpper`: The upper equivalence limit of paired difference.
    - `pairedDiff`: The paired difference under the alternative hypothesis.
    - `stDev`: The standard deviation for paired difference.
- `byStageResults`: A data frame containing the following variables:
    - `informationRates`: The information rates.
    - `efficacyBounds`: The efficacy boundaries on the Z-scale for each of the two one-sided tests.
    - `rejectPerStage`: The probability for efficacy stopping.
    - `cumulativeRejection`: The cumulative probability for efficacy stopping.
    - `cumulativeAlphaSpent`: The cumulative alpha for each of the two one-sided tests.
    - `cumulativeAttainedAlpha`: The cumulative probability for efficacy stopping under H0.
    - `efficacyPairedDiffLower`: The efficacy boundaries on the paired difference scale for the one-sided null hypothesis on the lower equivalence limit.
    - `efficacyPairedDiffUpper`: The efficacy boundaries on the paired difference scale for the one-sided null hypothesis on the upper equivalence limit.
    - `efficacyP`: The efficacy bounds on the p-value scale for each of the two one-sided tests.
    - `information`: The cumulative information.
    - `numberOfSubjects`: The number of subjects.
- `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.
    - `spendingTime`: The error spending time at each analysis.
    - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
    - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
# Example 1: group sequential trial power calculation
(design1 <- getDesignPairedMeanDiffEquiv(
  beta = 0.1, n = NA, pairedDiffLower = -1.3, pairedDiffUpper = 1.3,
  pairedDiff = 0, stDev = 2.2,
  kMax = 4, alpha = 0.05, typeAlphaSpending = "sfOF"))

# Example 2: sample size calculation for t-test
(design2 <- getDesignPairedMeanDiffEquiv(
  beta = 0.1, n = NA, pairedDiffLower = -1.3, pairedDiffUpper = 1.3,
  pairedDiff = 0, stDev = 2.2,
  normalApproximation = FALSE, alpha = 0.05))
```

`getDesignPairedMeanRatio` *Group Sequential Design for Paired Mean Ratio*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for paired mean ratio.

Usage

```
getDesignPairedMeanRatio(
  beta = NA_real_,
  n = NA_real_,
  pairedRatioH0 = 1,
  pairedRatio = 1.2,
  CV = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
```

```
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`pairedRatioH0`	The paired ratio under the null hypothesis.
`pairedRatio`	The paired ratio under the alternative hypothesis.
`CV`	The coefficient of variation for paired ratio.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.

Value

An S3 class `designPairedMeanRatio` 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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `pairedRatioH0`: The paired ratio under the null hypothesis.
  - `pairedRatio`: The paired ratio under the alternative hypothesis.
  - `CV`: The coefficient of variation for paired ratio.
- `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.
  - `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.
  - `numberOfSubjects`: The number of subjects.
  - `efficacyPairedRatio`: The efficacy boundaries on the paired ratio scale.
  - `futilityPairedRatio`: The futility boundaries on the paired ratio scale.
- `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.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Example 1: group sequential trial power calculation
(design1 <- getDesignPairedMeanRatio(
  beta = 0.1, n = NA, pairedRatio = 1.2, CV = 0.35,
  kMax = 5, alpha = 0.05, typeAlphaSpending = "sfOF"))

# Example 2: sample size calculation for one-sample t-test
(design2 <- getDesignPairedMeanRatio(
  beta = 0.1, n = NA, pairedRatio = 1.2, CV = 0.35,
  normalApproximation = FALSE, alpha = 0.05))
```

`getDesignPairedMeanRatioEquiv` *Group Sequential Design for Equivalence in Paired Mean Ratio*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for equivalence in paired mean ratio.

Usage

```
getDesignPairedMeanRatioEquiv(
  beta = NA_real_,
  n = NA_real_,
  pairedRatioLower = NA_real_,
  pairedRatioUpper = NA_real_,
  pairedRatio = 1,
  CV = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`pairedRatioLower`	The lower equivalence limit of paired ratio.
`pairedRatioUpper`	The upper equivalence limit of paired ratio.
`pairedRatio`	The paired ratio under the alternative hypothesis.
`CV`	The coefficient of variation for paired ratio.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.

`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.

Value

An S3 class `designPairedMeanRatioEquiv` object with three components:

- `overallResults`: A data frame containing the following variables:
  - `overallReject`: The overall rejection probability.
  - `alpha`: The significance level for each of the two one-sided tests. Defaults to 0.05.
  - `attainedAlpha`: The attained significance level under H0.
  - `kMax`: The number of stages.
  - `information`: The maximum information.
  - `expectedInformationH1`: The expected information under H1.
  - `expectedInformationH0`: The expected information under H0.
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `pairedRatioLower`: The lower equivalence limit of paired ratio.
  - `pairedRatioUpper`: The upper equivalence limit of paired ratio.
  - `pairedRatio`: The paired ratio under the alternative hypothesis.
  - `CV`: The coefficient of variation for paired ratios.
- `byStageResults`: A data frame containing the following variables:
  - `informationRates`: The information rates.
  - `efficacyBounds`: The efficacy boundaries on the Z-scale for each of the two one-sided tests.
  - `rejectPerStage`: The probability for efficacy stopping.
  - `cumulativeRejection`: The cumulative probability for efficacy stopping.

  - `cumulativeAlphaSpent`: The cumulative alpha for each of the two one-sided tests.
  - `cumulativeAttainedAlpha`: The cumulative alpha attained under H0.
  - `efficacyP`: The efficacy bounds on the p-value scale for each of the two one-sided tests.
  - `information`: The cumulative information.
  - `numberOfSubjects`: The number of subjects.
  - `efficacyPairedRatioLower`: The efficacy boundaries on the paired ratio scale for the one-sided null hypothesis on the lower equivalence limit.
  - `efficacyPairedRatioUpper`: The efficacy boundaries on the paired ratio scale for the one-sided null hypothesis on the upper equivalence limit.
- `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.
  - `spendingTime`: The error spending time at each analysis.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Example 1: group sequential trial power calculation
(design1 <- getDesignPairedMeanRatioEquiv(
  beta = 0.1, n = NA, pairedRatioLower = 0.8, pairedRatioUpper = 1.25,
  pairedRatio = 1, CV = 0.35,
  kMax = 4, alpha = 0.05, typeAlphaSpending = "sfOF"))

# Example 2: sample size calculation for t-test
(design2 <- getDesignPairedMeanRatioEquiv(
  beta = 0.1, n = NA, pairedRatioLower = 0.8, pairedRatioUpper = 1.25,
  pairedRatio = 1, CV = 0.35,
  normalApproximation = FALSE, alpha = 0.05))
```

`getDesignPairedPropMcNemar` *Group Sequential Design for McNemar's Test for Paired Proportions*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for McNemar's test for paired proportions.

Usage

```
getDesignPairedPropMcNemar(
  beta = NA_real_,
  n = NA_real_,
  pDiscordant = NA_real_,
  riskDiff = NA_real_,
  nullVariance = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`pDiscordant`	The proportion of discordant pairs (xi = pi01 + pi10).
`riskDiff`	The risk difference between the active and control treatments (delta = pi_t - pi_c = pi01 - pi10)
`nullVariance`	Whether to use the variance under the null or the variance under the alternative.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.

`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.

Details

Consider a group sequential design for McNemar's test for paired proportions. The table below shows joint probabilities for each cell (π_{ij} where i is for control group and j is for experimental group), with marginal totals.

	Experimental: No Response	Experimental: Response	Row Total
Control: No Response	π_{00}	π_{01}	$1-\pi_c$
Control: Response	π_{10}	π_{11}	π_c
Column Total	$1-\pi_t$	π_t	1

The parameters π_{01} and π_{10} are the discordant pairs (i.e., $\pi_{01}+\pi_{10}=\xi$) and the risk difference is $\pi_{01}-\pi_{10}=\delta$. The parameter π_t is the proportion of experimental group response, and π_c is the proportion of control group response. The parameter of interest is

$$\theta = \pi_t - \pi_c = \pi_{01} - \pi_{10} = \delta$$

The variance of $\hat\theta$ can be obtained from the multinomial distribution as follows:

$$Var(\hat\theta) = \frac{1}{n}\{\pi_{01}(1-\pi_{01}) + \pi_{10}(1-\pi_{10}) + 2\pi_{01}\pi_{10}\}$$

which can be simplified to

$$Var(\hat{\theta}) = \frac{1}{n}(\xi - \delta^2)$$

Here, n is the total number of treatment pairs. This is the unconditional variance, which is used for the overall design.

Value

An S3 class `designPairedPropMcNemar` 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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `pDiscordant`: The proportion of discordant pairs (xi = pi01 + pi10).
  - `riskDiff`: The risk difference between the active and control treatments (delta = pi_t - pi_c = pi01 - pi10)
- `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.
  - `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.
  - `efficacyRiskDiff`: The efficacy boundaries on the risk difference scale.
  - `futilityRiskDiff`: The futility boundaries on the risk difference scale.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
# Example 1: fixed design
(design1 <- getDesignPairedPropMcNemar(
  beta = 0.1, n = NA, pDiscordant = 0.16, riskDiff = 0.1,
  alpha = 0.025))

# Example 2: group sequential design
(design2 <- getDesignPairedPropMcNemar(
  beta = 0.1, n = NA, pDiscordant = 0.16, riskDiff = 0.1,
  alpha = 0.025, kMax = 3, typeAlphaSpending = "sfOF"))
```

`getDesignRepeatedANOVA` *Power and Sample Size for Repeated-Measures ANOVA*

Description

Obtains the power and sample size for one-way repeated measures analysis of variance. Each subject takes all treatments in the longitudinal study.

Usage

```
getDesignRepeatedANOVA(
  beta = NA_real_,
  n = NA_real_,
  ngroups = 2,
  means = NA_real_,
  stDev = 1,
  corr = 0,
  rounding = TRUE,
  alpha = 0.05
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`ngroups`	The number of treatment groups.
`means`	The treatment group means.
`stDev`	The total standard deviation.
`corr`	The correlation among the repeated measures.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`alpha`	The two-sided significance level. Defaults to 0.05.

Details

Let y_{ij} denote the measurement under treatment condition $j (j = 1, \ldots, k)$ for subject $i (i = 1, \ldots, n)$. Then

$$y_{ij} = \alpha + \beta_j + b_i + e_{ij}$$

where b_i denotes the subject random effect, $b_i \sim N(0, \sigma_b^2)$ and $e_{ij} \sim N(0, \sigma_e^2)$ denotes the within-subject residual. If we set $\beta_k = 0$, then α is the mean of the last treatment (control), and β_j is the difference in means between the jth treatment and the control for $j = 1, \ldots, k-1$.

The repeated measures have a compound symmetry covariance structure. Let $\sigma^2 = \sigma_b^2 + \sigma_e^2$, and $\rho = \frac{\sigma_b^2}{\sigma_b^2 + \sigma_e^2}$. Then $Var(y_i) = \sigma^2 \{(1-\rho) I_k + \rho 1_k 1_k^T\}$. Let X_i denote the design matrix for subject i. Let $\theta = (\alpha, \beta_1, \ldots, \beta_{k-1})^T$. It follows that

$$Var(\hat{\theta}) = \left(\sum_{i=1}^{n} X_i^T V_i^{-1} X_i \right)^{-1}.$$

It can be shown that

$$Var(\hat{\beta}) = \frac{\sigma^2 (1-\rho)}{n} (I_{k-1} + 1_{k-1} 1_{k-1}^T).$$

It follows that $\hat{\beta}^T \hat{V}_{\hat{\beta}}^{-1} \hat{\beta} \sim F_{k-1,(n-1)(k-1),\lambda}$ where the noncentrality parameter for the F distribution is

$$\lambda = \beta^T V_{\hat{\beta}}^{-1} \beta = \frac{n \sum_{j=1}^{k} (\mu_j - \bar{\mu})^2}{\sigma^2 (1-\rho)}.$$

Value

An S3 class `designRepeatedANOVA` object with the following components:

- `power`: The power to reject the null hypothesis that there is no difference among the treatment groups.
- `alpha`: The two-sided significance level.
- `n`: The number of subjects.
- `ngroups`: The number of treatment groups.
- `means`: The treatment group means.
- `stDev`: The total standard deviation.
- `corr`: The correlation among the repeated measures.
- `effectsize`: The effect size.
- `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
(design1 <- getDesignRepeatedANOVA(
  beta = 0.1, ngroups = 4, means = c(1.5, 2.5, 2, 0),
  stDev = 5, corr = 0.2, alpha = 0.05))
```

`getDesignRepeatedANOVAContrast` *Power and Sample Size for One-Way Repeated Measures ANOVA Contrast*

Description

Obtains the power and sample size for a single contrast in one-way repeated measures analysis of variance.

Usage

```
getDesignRepeatedANOVAContrast(
  beta = NA_real_,
  n = NA_real_,
  ngroups = 2,
  means = NA_real_,
  stDev = 1,
  corr = 0,
  contrast = NA_real_,
```

```
  meanContrastH0 = 0,
  rounding = TRUE,
  alpha = 0.025
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`ngroups`	The number of treatment groups.
`means`	The treatment group means.
`stDev`	The total standard deviation.
`corr`	The correlation among the repeated measures.
`contrast`	The coefficients for the single contrast.
`meanContrastH0`	The mean of the contrast under the null hypothesis.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`alpha`	The one-sided significance level. Defaults to 0.025.

Value

An S3 class `designRepeatedANOVAContrast` object with the following components:

- `power`: The power to reject the null hypothesis for the treatment contrast.
- `alpha`: The one-sided significance level.
- `n`: The number of subjects.
- `ngroups`: The number of treatment groups.
- `means`: The treatment group means.
- `stDev`: The total standard deviation.
- `corr`: The correlation among the repeated measures.
- `contrast`: The coefficients for the single contrast.
- `meanContrastH0`: The mean of the contrast under the null hypothesis.
- `meanContrast`: The mean of the contrast under the alternative hypothesis.
- `effectsize`: The effect size.
- `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
(design1 <- getDesignRepeatedANOVAContrast(
  beta = 0.1, ngroups = 4, means = c(1.5, 2.5, 2, 0),
  stDev = 5, corr = 0.2, contrast = c(1, 1, 1, -3)/3,
  alpha = 0.025))
```

`getDesignRiskDiff` *Group Sequential Design for Two-Sample Risk Difference*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for two-sample risk difference.

Usage

```
getDesignRiskDiff(
  beta = NA_real_,
  n = NA_real_,
  riskDiffH0 = 0,
  pi1 = NA_real_,
  pi2 = NA_real_,
  nullVariance = TRUE,
  allocationRatioPlanned = 1,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`riskDiffH0`	The risk difference under the null hypothesis. Defaults to 0.
`pi1`	The assumed probability for the active treatment group.
`pi2`	The assumed probability for the control group.
`nullVariance`	Whether to use the variance under the null or the empirical variance under the alternative.

`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.

Details

Consider a group sequential design for two-sample risk difference. The parameter of interest is

$$\theta = \pi_1 - \pi_2$$

where π_1 is the response probability for the active treatment group and π_2 is the response probability for the control group. The variance of $\hat{\theta}$ can be obtained from the binomial distributions as follows:

$$Var(\hat{\theta}) = \frac{1}{n}\{\frac{\pi_1(1-\pi_1)}{r} + \frac{\pi_2(1-\pi_2)}{1-r}\}$$

where n is the total number of subjects and r is the randomization probability for the active treatment group. When `nullVariance = TRUE`, the variance is computed under the null hypothesis. In this case, the values of π_1 and π_2 in the variance formula are replaced with their restricted maximum likelihood counterparts, subject to the constraint

$$\pi_1 - \pi_2 = \theta_0$$

Value

An S3 class `designRiskDiff` 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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `riskDiffH0`: The risk difference under the null hypothesis.
  - `pi1`: The assumed probability for the active treatment group.
  - `pi2`: The assumed probability for the control group.
- `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.
  - `efficacyRiskDiff`: The efficacy boundaries on the risk difference scale.
  - `futilityRiskDiff`: The futility boundaries on the risk difference 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.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `nullVariance`: Whether to use the variance under the null or the empirical variance under the alternative.
  - `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignRiskDiff(
  beta = 0.2, n = NA, pi1 = 0.1, pi2 = 0.15,
  kMax = 3, alpha = 0.025, typeAlphaSpending = "sfOF",
  nullVariance = FALSE))
```

`getDesignRiskDiffEquiv` *Group Sequential Design for Equivalence in Two-Sample Risk Difference*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for equivalence in two-sample risk difference.

Usage

```
getDesignRiskDiffEquiv(
  beta = NA_real_,
  n = NA_real_,
  riskDiffLower = NA_real_,
  riskDiffUpper = NA_real_,
  pi1 = NA_real_,
  pi2 = NA_real_,
  allocationRatioPlanned = 1,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  criticalValues = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`riskDiffLower`	The lower equivalence limit of risk difference.
`riskDiffUpper`	The upper equivalence limit of risk difference.
`pi1`	The assumed probability for the active treatment group.
`pi2`	The assumed probability for the control group.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.

`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.

Value

An S3 class `designRiskDiffEquiv` object with three components:

- `overallResults`: A data frame containing the following variables:
    - `overallReject`: The overall rejection probability.
    - `alpha`: The significance level for each of the two one-sided tests. Defaults to 0.05.
    - `attainedAlphaH10`: The attained significance level under H10.
    - `attainedAlphaH20`: The attained significance level under H20.
    - `kMax`: The number of stages.
    - `information`: The maximum information.
    - `expectedInformationH1`: The expected information under H1.
    - `expectedInformationH10`: The expected information under H10.
    - `expectedInformationH20`: The expected information under H20.
    - `numberOfSubjects`: The maximum number of subjects.
    - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
    - `expectedNumberOfSubjectsH10`: The expected number of subjects under H10.
    - `expectedNumberOfSubjectsH20`: The expected number of subjects under H20.
    - `riskDiffLower`: The lower equivalence limit of risk difference.
    - `riskDiffUpper`: The upper equivalence limit of risk difference.
    - `pi1`: The assumed probability for the active treatment group.
    - `pi2`: The assumed probability for the control group.
    - `riskDiff`: The risk difference.
- `byStageResults`: A data frame containing the following variables:

  - `informationRates`: The information rates.
  - `efficacyBounds`: The efficacy boundaries on the Z-scale for each of the two one-sided tests.
  - `rejectPerStage`: The probability for efficacy stopping.
  - `cumulativeRejection`: The cumulative probability for efficacy stopping.
  - `cumulativeAlphaSpent`: The cumulative alpha for each of the two one-sided tests.
  - `cumulativeAttainedAlphaH10`: The cumulative alpha attained under H10.
  - `cumulativeAttainedAlphaH20`: The cumulative alpha attained under H20.
  - `efficacyP`: The efficacy bounds on the p-value scale for each of the two one-sided tests.
  - `information`: The cumulative information.
  - `efficacyRiskDiffLower`: The efficacy boundaries on the risk difference scale for the one-sided null hypothesis on the lower equivalence limit.
  - `efficacyRiskDiffUpper`: The efficacy boundaries on the risk difference scale for the one-sided null hypothesis on the upper equivalence limit.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `spendingTime`: The error spending time at each analysis.
  - `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignRiskDiffEquiv(
  beta = 0.2, n = NA, riskDiffLower = -0.1,
  riskDiffUpper = 0.1, pi1 = 0.12, pi2 = 0.12,
  kMax = 3, alpha = 0.05, typeAlphaSpending = "sfOF"))
```

`getDesignRiskDiffExact`

Power and Sample Size for Exact Unconditional Test for Risk Difference

Description

Obtains the power given sample size or obtains the sample size given power for exact unconditional test of risk difference.

Usage

```
getDesignRiskDiffExact(
  beta = NA_real_,
  n = NA_real_,
  riskDiffH0 = 0,
  pi1 = NA_real_,
  pi2 = NA_real_,
  allocationRatioPlanned = 1,
  alpha = 0.025
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`riskDiffH0`	The risk difference under the null hypothesis. Defaults to 0.
`pi1`	The assumed probability for the active treatment group.
`pi2`	The assumed probability for the control group.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`alpha`	The one-sided significance level. Defaults to 0.025.

Value

A data frame with the following variables:

- `alpha`: The specified one-sided significance level.
- `attainedAlpha`: The attained one-sided significance level.
- `power`: The power.
- `n`: The sample size.
- `riskDiffH0`: The risk difference under the null hypothesis.
- `pi1`: The assumed probability for the active treatment group.
- `pi2`: The assumed probability for the control group.
- `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
- `zstatRiskDiffBound`: The critical value on the scale of score test statistic for risk difference.
- `pi2star`: The response probability in the control group at which the critical value of the test statistic is attained.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
# Superiority test

getDesignRiskDiffExact(n = 50, pi1 = 0.6, pi2 = 0.25, alpha = 0.025)


# Non-inferiority test

getDesignRiskDiffExact(beta = 0.2, riskDiffH0 = -0.3,
                       pi1 = 0.9, pi2 = 0.9, alpha = 0.025)
```

`getDesignRiskDiffExactEquiv` *Power and Sample Size for Exact Unconditional Test for Equivalence in Risk Difference*

Description

Obtains the power given sample size or obtains the sample size given power for exact unconditional test of equivalence in risk difference.

Usage

```
getDesignRiskDiffExactEquiv(
  beta = NA_real_,
  n = NA_real_,
  riskDiffLower = NA_real_,
  riskDiffUpper = NA_real_,
  pi1 = NA_real_,
  pi2 = NA_real_,
  allocationRatioPlanned = 1,
  alpha = 0.05
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`riskDiffLower`	The lower equivalence limit of risk difference.
`riskDiffUpper`	The upper equivalence limit of risk difference.
`pi1`	The assumed probability for the active treatment group.
`pi2`	The assumed probability for the control group.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.

Value

A data frame with the following variables:

- `alpha`: The specified significance level for each of the two one-sided tests.
- `attainedAlpha`: The attained significance level.
- `power`: The power.
- `n`: The sample size.
- `riskDiffLower`: The lower equivalence limit of risk difference.
- `riskDiffUpper`: The upper equivalence limit of risk difference.
- `pi1`: The assumed probability for the active treatment group.
- `pi2`: The assumed probability for the control group.
- `riskDiff`: The risk difference.
- `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
- `zstatRiskDiffLower`: The efficacy boundaries on the z-test statistic scale for the one-sided null hypothesis on the lower equivalence limit.
- `zstatRiskDiffUpper`: The efficacy boundaries on the z-test statistic scale for the one-sided null hypothesis on the upper equivalence limit.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
getDesignRiskDiffExactEquiv(
  n = 200, riskDiffLower = -0.2, riskDiffUpper = 0.2,
  pi1 = 0.775, pi2 = 0.775, alpha = 0.05)
```

`getDesignRiskRatio` *Group Sequential Design for Two-Sample Risk Ratio*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for two-sample risk ratio.

Usage

```
getDesignRiskRatio(
  beta = NA_real_,
  n = NA_real_,
  riskRatioH0 = 1,
  pi1 = NA_real_,
  pi2 = NA_real_,
  nullVariance = TRUE,
  allocationRatioPlanned = 1,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`riskRatioH0`	The risk ratio under the null hypothesis. Defaults to 1.
`pi1`	The assumed probability for the active treatment group.
`pi2`	The assumed probability for the control group.
`nullVariance`	Whether to use the variance under the null or the empirical variance under the alternative.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.

`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.

Details

Consider a group sequential design for two-sample risk ratio. The parameter of interest is

$$\rho = \pi_1/\pi_2$$

where π_1 is the response probability for the active treatment group and π_2 is the response probability for the control group. For statistical inference, the parameter is often transformed to the log scale:

$$\theta = \log(\rho) = \log(\pi_1) - \log(\pi_2)$$

The variance of the estimator $\hat{\theta}$ can be derived from the binomial distributions as follows:

$$Var(\hat{\theta}) = \frac{1}{n}\{\frac{1-\pi_1}{\pi_1 r} + \frac{1-\pi_2}{\pi_2(1-r)}\}$$

where n is the total number of subjects and r is the randomization probability for the active treatment group. When `nullVariance = TRUE`, the variance is computed under the null hypothesis. In this case, the values of π_1 and π_2 in the variance formula are replaced with their restricted maximum likelihood counterparts, subject to the constraint

$$\pi_1/\pi_2 = \rho_0$$

Value

An S3 class `designRiskRatio` 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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `riskRatioH0`: The risk ratio under the null hypothesis.
  - `pi1`: The assumed probability for the active treatment group.
  - `pi2`: The assumed probability for the control group.
- `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.
  - `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.
  - `efficacyRiskRatio`: The efficacy boundaries on the risk ratio scale.
  - `futilityRiskRatio`: The futility boundaries on the risk ratio scale.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `nullVariance`: Whether to use the variance under the null or the empirical variance under the alternative.
  - `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
(design1 <- getDesignRiskRatio(
  beta = 0.1, n = NA, pi1 = 0.5, pi2 = 0.3,
  alpha = 0.05))
```

`getDesignRiskRatioEquiv` *Group Sequential Design for Equivalence in Two-Sample Risk Ratio*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for equivalence in two-sample risk ratio.

Usage

```
getDesignRiskRatioEquiv(
  beta = NA_real_,
  n = NA_real_,
  riskRatioLower = NA_real_,
  riskRatioUpper = NA_real_,
  pi1 = NA_real_,
  pi2 = NA_real_,
  allocationRatioPlanned = 1,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  criticalValues = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`riskRatioLower`	The lower equivalence limit of risk ratio.
`riskRatioUpper`	The upper equivalence limit of risk ratio.
`pi1`	The assumed probability for the active treatment group.
`pi2`	The assumed probability for the control group.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".

`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.

Value

An S3 class `designRiskRatioEquiv` object with three components:

- `overallResults`: A data frame containing the following variables:
  - `overallReject`: The overall rejection probability.
  - `alpha`: The significance level for each of the two one-sided tests. Defaults to 0.05.
  - `attainedAlphaH10`: The attained significance level under H10.
  - `attainedAlphaH20`: The attained significance level under H20.
  - `kMax`: The number of stages.
  - `information`: The maximum information.
  - `expectedInformationH1`: The expected information under H1.
  - `expectedInformationH10`: The expected information under H10.
  - `expectedInformationH20`: The expected information under H20.
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH10`: The expected number of subjects under H10.
  - `expectedNumberOfSubjectsH20`: The expected number of subjects under H20.
  - `riskRatioLower`: The lower equivalence limit of risk ratio.
  - `riskRatioUpper`: The upper equivalence limit of risk ratio.
  - `pi1`: The assumed probability for the active treatment group.
  - `pi2`: The assumed probability for the control group.
  - `riskRatio`: The risk ratio.
- `byStageResults`: A data frame containing the following variables:
  - `informationRates`: The information rates.
  - `efficacyBounds`: The efficacy boundaries on the Z-scale for each of the two one-sided tests.
  - `rejectPerStage`: The probability for efficacy stopping.
  - `cumulativeRejection`: The cumulative probability for efficacy stopping.
  - `cumulativeAlphaSpent`: The cumulative alpha for each of the two one-sided tests.
  - `cumulativeAttainedAlphaH10`: The cumulative alpha attained under H10.
  - `cumulativeAttainedAlphaH20`: The cumulative alpha attained under H20.
  - `efficacyRiskRatioLower`: The efficacy boundaries on the risk ratio scale for the one-sided null hypothesis on the lower equivalence limit.
  - `efficacyRiskRatioUpper`: The efficacy boundaries on the risk ratio scale for the one-sided null hypothesis on the upper equivalence limit.

  - `efficacyP`: The efficacy bounds on the p-value scale for each of the two one-sided tests.
  - `information`: The cumulative information.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `spendingTime`: The error spending time at each analysis.
  - `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignRiskRatioEquiv(
  beta = 0.2, n = NA, riskRatioLower = 0.8,
  riskRatioUpper = 1.25, pi1 = 0.12, pi2 = 0.12,
  kMax = 3, alpha = 0.05, typeAlphaSpending = "sfOF"))
```

`getDesignRiskRatioExact`

Power and Sample Size for Exact Unconditional Test for Risk Ratio

Description

Obtains the power given sample size or obtains the sample size given power for exact unconditional test of risk ratio.

Usage

```
getDesignRiskRatioExact(
  beta = NA_real_,
  n = NA_real_,
  riskRatioH0 = 1,
  pi1 = NA_real_,
  pi2 = NA_real_,
  allocationRatioPlanned = 1,
  alpha = 0.025
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`riskRatioH0`	The risk ratio under the null hypothesis. Defaults to 0.
`pi1`	The assumed probability for the active treatment group.
`pi2`	The assumed probability for the control group.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`alpha`	The one-sided significance level. Defaults to 0.025.

Value

A data frame with the following variables:

- `alpha`: The specified one-sided significance level.
- `attainedAlpha`: The attained one-sided significance level.
- `power`: The power.
- `n`: The sample size.
- `riskRatioH0`: The risk ratio under the null hypothesis.
- `pi1`: The assumed probability for the active treatment group.
- `pi2`: The assumed probability for the control group.
- `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
- `zstatRiskRatioBound`: The critical value on the scale of score test statistic for risk ratio.
- `pi2star`: The response probability in the control group at which the critical value of the test statistic is attained.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Non-inferiority test

getDesignRiskRatioExact(beta = 0.2, riskRatioH0 = 0.7,
                        pi1 = 0.95, pi2 = 0.95, alpha = 0.025)
```

`getDesignRiskRatioExactEquiv` *Power and Sample Size for Exact Unconditional Test for Equivalence in Risk Ratio*

Description

Obtains the power given sample size or obtains the sample size given power for exact unconditional test of equivalence in risk ratio.

Usage

```
getDesignRiskRatioExactEquiv(
  beta = NA_real_,
  n = NA_real_,
  riskRatioLower = NA_real_,
  riskRatioUpper = NA_real_,
  pi1 = NA_real_,
  pi2 = NA_real_,
  allocationRatioPlanned = 1,
  alpha = 0.05
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`riskRatioLower`	The lower equivalence limit of risk ratio.
`riskRatioUpper`	The upper equivalence limit of risk ratio.
`pi1`	The assumed probability for the active treatment group.
`pi2`	The assumed probability for the control group.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.

Value

A data frame with the following variables:

- `alpha`: The specified significance level for each of the two one-sided tests.
- `attainedAlpha`: The attained significance level.
- `power`: The power.
- `n`: The sample size.

- `riskRatioLower`: The lower equivalence limit of risk ratio.
- `riskRatioUpper`: The upper equivalence limit of risk ratio.
- `pi1`: The assumed probability for the active treatment group.
- `pi2`: The assumed probability for the control group.
- `riskRatio`: The risk ratio.
- `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
- `zstatRiskRatioLower`: The efficacy boundaries on the z-test statistic scale for the one-sided null hypothesis on the lower equivalence limit.
- `zstatRiskRatioUpper`: The efficacy boundaries on the z-test statistic scale for the one-sided null hypothesis on the upper equivalence limit.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
getDesignRiskRatioExactEquiv(
  n = 200, riskRatioLower = 0.8, riskRatioUpper = 1.25,
  pi1 = 0.775, pi2 = 0.775, alpha = 0.05)
```

`getDesignRiskRatioFM` *Group Sequential Design for Two-Sample Risk Ratio Based on the Farrington-Manning Score Test*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for two-sample risk ratio based on the Farrington-Manning score test

Usage

```
getDesignRiskRatioFM(
  beta = NA_real_,
  n = NA_real_,
  riskRatioH0 = 1,
  pi1 = NA_real_,
  pi2 = NA_real_,
  nullVariance = TRUE,
  allocationRatioPlanned = 1,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
```

```
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`riskRatioH0`	The risk ratio under the null hypothesis. Defaults to 1.
`pi1`	The assumed probability for the active treatment group.
`pi2`	The assumed probability for the control group.
`nullVariance`	Whether to use the variance under the null or the empirical variance under the alternative.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".

`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.

Details

Consider a group sequential design for two-sample risk ratio. The parameter of interest is

$$\rho = \pi_1/\pi_2$$

where π_1 is the response probability for the active treatment group and π_2 is the response probability for the control group. Let ρ_0 denote the risk ratio under the null hypothesis. The Farrington-Manning score test statistic is constructed as

$$Z = \frac{\hat{\pi}_1 - \rho_0\hat{\pi}_2}{\sqrt{Var(\hat{\pi}_1 - \rho_0\hat{\pi}_2)}}$$

The variance can be derived from the binomial distributions as follows:

$$Var(\hat{\pi}_1 - \rho_0\hat{\pi}_2) = \frac{1}{n}\{\frac{\pi_1(1-\pi_1)}{r} + \frac{\rho_0^2\pi_2(1-\pi_2)}{1-r}\}$$

where n is the total number of subjects and r is the randomization probability for the active treatment group. When `nullVariance = TRUE`, the variance is computed under the null hypothesis. In this case, the values of π_1 and π_2 in the variance formula are replaced with their restricted maximum likelihood counterparts, subject to the constraint

$$\pi_1/\pi_2 = \rho_0$$

Value

An S3 class `designRiskRatioFM` 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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `riskRatioH0`: The risk ratio under the null hypothesis.
  - `pi1`: The assumed probability for the active treatment group.
  - `pi2`: The assumed probability for the control group.
- `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.
  - `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.
  - `efficacyRiskRatioScore`: The efficacy boundaries on the score test `pi1 - riskRatioH0*pi2` score.
  - `futilityRiskRatioScore`: The futility boundaries on the score test `pi1 - riskRatioH0*pi2` scale.

  - `numberOfSubjects`: The number of subjects.
- `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.
  - `nullVariance`: Whether to use the variance under the null or the empirical variance under the alternative.
  - `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignRiskRatioFM(
  beta = 0.2, riskRatioH0 = 1.3, pi1 = 0.125, pi2 = 0.125,
  alpha = 0.05))
```

`getDesignSlopeDiff` *Group Sequential Design for Two-Sample Slope Difference*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for two-sample slope difference.

Usage

```
getDesignSlopeDiff(
  beta = NA_real_,
  n = NA_real_,
  slopeDiffH0 = 0,
  slopeDiff = 0.5,
  stDev = 1,
  stDevCovariate = 1,
  allocationRatioPlanned = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
```

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

Arguments

`beta`	The type II error.
`n`	The total sample size.
`slopeDiffH0`	The slope difference under the null hypothesis. Defaults to 0.
`slopeDiff`	The slope difference under the alternative hypothesis.
`stDev`	The standard deviation of the residual.
`stDevCovariate`	The standard deviation of the covariate.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The exact calculation using the t distribution is only implemented for the fixed design.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.

`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.

Details

We assume a simple linear regression of the form

$$y_{g,i} = \alpha_g + \beta_g x_{g,i} + \epsilon_{g,i}$$

for treatment group g, where $\epsilon_{g,i}$ is the residual error for subject i in group g, which is assumed to be normally distributed with mean 0 and standard deviation σ_ϵ. The covariate $x_{g,i}$ is assumed to be normally distributed with mean 0 and standard deviation σ_x. Since

$$\hat{\beta}_g = \frac{\sum_{i=1}^{n_g}(x_{g,i} - \bar{x}_g) y_{g,i}}{\sum_{i=1}^{n_g}(x_{g,i} - \bar{x}_g)^2}$$

where n_g is the sample size for group g, it follows that

$$\hat{\beta}_g \sim N(\beta_g, \frac{\sigma_\epsilon^2}{\sum_{i=1}^{n_g}(x_{g,i} - \bar{x}_g)^2}).$$

The slope difference is defined as

$$\hat{\beta}_1 - \hat{\beta}_2$$

where $\hat{\beta}_1$ and $\hat{\beta}_2$ are the estimated slopes for treatment groups 1 and 2, respectively. Since the variance of $\hat{\beta}_g$ is

$$\frac{\sigma_\epsilon^2}{n_g \sigma_x^2}$$

we have

$$\hat{\beta}_1 - \hat{\beta}_2 \sim N(\beta_1 - \beta_2, (n_1^{-1} + n_2^{-1})\sigma_\epsilon^2/\sigma_x^2)$$

which can be used to calculate the power and sample size for the group sequential design.

Value

An S3 class `designSlopeDiff` 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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `slopeDiffH0`: The slope difference under the null hypothesis.
  - `slopeDiff`: The slope difference under the alternative hypothesis.
  - `stDev`: The standard deviation of the residual.
  - `stDevCovariate`: The standard deviation of the covariate.
- `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.
  - `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.
  - `efficacySlopeDiff`: The efficacy boundaries on the slope difference scale.
  - `futilitySlopeDiff`: The futility boundaries on the slope difference scale.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignSlopeDiff(
  beta = 0.1, n = NA, slopeDiff = -0.5,
  stDev = 10, stDevCovariate = 6,
  normalApproximation = FALSE, alpha = 0.025))
```

`getDesignSlopeDiffMMRM` *Group Sequential Design for Two-Sample Slope Difference From the MMRM Model*

Description

Obtains the power given sample size or obtains the sample size given power for two-sample slope difference from the growth curve MMRM model.

Usage

```
getDesignSlopeDiffMMRM(
  beta = NA_real_,
  slopeDiffH0 = 0,
  slopeDiff = 0.5,
  stDev = 1,
  stDevIntercept = 1,
  stDevSlope = 1,
  corrInterceptSlope = 0.5,
  w = NA_real_,
  N = NA_real_,
  accrualTime = 0,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0,
  gamma1 = 0,
  gamma2 = 0,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = FALSE,
  allocationRatioPlanned = 1,
  normalApproximation = TRUE,
  rounding = TRUE,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  spendingTime = NA_real_
)
```

Arguments

`beta`	The type II error.
`slopeDiffH0`	The slope difference under the null hypothesis. Defaults to 0.
`slopeDiff`	The slope difference under the alternative hypothesis.
`stDev`	The standard deviation of the residual.
`stDevIntercept`	The standard deviation of the random intercept.
`stDevSlope`	The standard deviation of the random slope.

`corrInterceptSlope`	The correlation between the random intercept and random slope.
`w`	The number of time units (e.g. weeks) per measurement visit in a period. In general, visits are more frequent in the beginning of the study and less frequent towards the end.
`N`	The number of measurement visits in a period. For example, `w = c(8, 16)` and `N = c(2, Inf)` means that the response variable will be collected at baseline, week 8, week 16, and every 16 weeks thereafter.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`gamma1`	The hazard rate for exponential dropout, or a vector of hazard rates for piecewise exponential dropout for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, or a vector of hazard rates for piecewise exponential dropout 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.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`normalApproximation`	The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution. The degrees of freedom for the t-distribution for testing the slope difference is calculated using the containment method, and is equal to the total number of observations minus two times the total number of subjects. The exact calculation using the t distribution is only implemented for the fixed design.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. 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.
`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`.

Details

We use the following random-effects model to compare two slopes:

$$y_{ij} = \alpha + (\beta + \gamma x_i)t_j + a_i + b_i t_j + e_{ij}$$

where

- α: overall intercept common across treatment groups due to randomization
- β: slope for the control group
- γ: difference in slopes between the active treatment and control groups
- x_i: treatment indicator for subject i, 1 for the active treatment and 0 for the control
- t_j: time point j for repeated measurements, $t_1 = 0 < t_2 < \ldots < t_k$
- (a_i, b_i): random intercept and random slope for subject i, $Var(a_i) = \sigma_a^2$, $Var(b_i) = \sigma_b^2$, $Corr(a_i, b_i) = \rho$

- e_{ij}: within-subject residual with variance σ_e^2

By accounting for randomization, we improve the efficiency for estimating the difference in slopes. The model also accommodates unequally spaced time points and missing data. Specifically, given a calendar time τ for an interim or final analysis, let k be the number of scheduled time points up to and including τ, subject to the follow-up duration for fixed follow-up designs. Let the observed time points be $t_1, t_2, \ldots, t_k$, where $t_1 = 0$ denotes baseline.

For a subject in treatment group g with observed data pattern j, the design matrix for the fixed effects $(\alpha, \beta, \gamma)'$ is given by

$$X_{g,j} = (\mathbf{1}_j, \vec{t}_j, I(g=1)\vec{t}_j)$$

where $\mathbf{1}_j$ is a j-vector of ones, and $\vec{t}_j = (t_1, \ldots, t_j)'$ is the column vector of observed time points. The design matrix for the random effects $(a_i, b_i)'$ is

$$Z_j = (\mathbf{1}_j, \vec{t}_j)$$

The variance-covariance matrix of the random effects is

$$D = \left(\begin{array}{cc} \sigma_a^2 & \rho\sigma_a\sigma_b \\ \rho\sigma_a\sigma_b & \sigma_b^2 \end{array} \right)$$

Therefore, the variance-covariance matrix for the observed data for the subject is

$$V_j = Z_j D Z_j' + \sigma_e^2 I_j$$

where I_j is the $j \times j$ identity matrix. Let π_g denote the proportion of subjects in group g. The information matrix for the fixed effects is

$$I = nJ$$

where

$$J = \sum_{g=1}^{2} \pi_g \sum_{j=1}^{k} p_{g,j} X_{g,j}' V_j^{-1} X_{g,j}$$

and $p_{g,j}$ is the proportion of subjects in group g with observed data pattern j.

The variance of the estimator for the slope difference $\hat{\gamma}$ is given by

$$\mathrm{Var}(\hat{\gamma}) = \frac{1}{n} J^{-1}[3,3]$$

which can be used to calculate the power and sample size for the group sequential design to detect a slope difference.

Value

An S3 class `designSlopeDiffMMRM` 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*sqrt(information)`.
  - `inflationFactor`: The inflation factor (relative to the fixed design).
  - `numberOfSubjects`: The maximum number of subjects.
  - `studyDuration`: The maximum study duration.
  - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
  - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
  - `expectedStudyDurationH1`: The expected study duration under H1.
  - `expectedStudyDurationH0`: The expected study duration under H0.
  - `accrualDuration`: The accrual duration.
  - `followupTime`: The follow-up time.
  - `fixedFollowup`: Whether a fixed follow-up design is used.
  - `slopeDiffH0`: The slope difference under H0.
  - `slopeDiff`: The slope difference under H1.
- `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.
  - `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.
  - `efficacySlopeDiff`: The efficacy boundaries on the slope difference scale.
  - `futilitySlopeDiff`: The futility boundaries on the slope difference scale.
  - `numberOfSubjects`: The number of subjects.
  - `analysisTime`: The average time since trial start.
- `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.
  - `allocationRatioPlanned`: The allocation ratio for the active treatment versus control.
  - `accrualTime`: A vector that specifies the starting time of piecewise Poisson enrollment time intervals.
  - `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.
  - `gamma1`: The hazard rate for exponential dropout or a vector of hazard rates for piecewise exponential dropout for the active treatment group.
  - `gamma2`: The hazard rate for exponential dropout or a vector of hazard rates for piecewise exponential dropout for the control group.
  - `w`: The number of time units per measurement visit in a period.
  - `N`: The number of measurement visits in a period.
  - `stdDev`: The standard deviation of the residual.
  - `G`: The covariance matrix for the random intercept and random slope.
  - `normalApproximation`: The type of computation of the p-values. If `TRUE`, the variance is assumed to be known, otherwise the calculations are performed with the t distribution.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

Daniel O. Scharfstein, Anastasios A. Tsiatis, and James M. Robins. Semiparametric efficiency and its implication on the design and analysis of group-sequential studies. Journal of the American Statistical Association 1997; 92:1342-1350.

Examples

```
(design1 <- getDesignSlopeDiffMMRM(
  beta = 0.2, slopeDiff = log(1.15)/52,
  stDev = sqrt(.182),
  stDevIntercept = sqrt(.238960),
  stDevSlope = sqrt(.000057),
  corrInterceptSlope = .003688/sqrt(.238960*.000057),
  w = 8,
  N = 10000,
  accrualIntensity = 15,
  gamma1 = 1/(4.48*52),
  gamma2 = 1/(4.48*52),
  accrualDuration = NA,
```

```
  followupTime = 8,
  alpha = 0.025))
```

`getDesignTwoMultinom` *Power and Sample Size for Difference in Two-Sample Multinomial Responses*

Description

Obtains the power given sample size or obtains the sample size given power for difference in two-sample multinomial responses.

Usage

```
getDesignTwoMultinom(
  beta = NA_real_,
  n = NA_real_,
  ncats = NA_integer_,
  pi1 = NA_real_,
  pi2 = NA_real_,
  allocationRatioPlanned = 1,
  rounding = TRUE,
  alpha = 0.05
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`ncats`	The number of categories of the multinomial response.
`pi1`	The prevalence of each category for the treatment group. Only need to specify the valued for the first `ncats-1` categories.
`pi2`	The prevalence of each category for the control group. Only need to specify the valued for the first `ncats-1` categories.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`alpha`	The two-sided significance level. Defaults to 0.05.

Details

A two-arm multinomial response design is used to test whether the prevalence of each category differs between two treatment arms. Let π_{gi} denote the prevalence of category i in group g, where $g = 1$ for the treatment group and $g = 2$ for the control group. The chi-square test statistic is given by

$$X^2 = \sum_{g=1}^{2} \sum_{i=1}^{C} \frac{(n_{gi} - n_{g+}n_{+i}/n)^2}{n_{g+}n_{+i}/n}$$

where n_{gi} is the number of subjects in category i for group g, n_{g+} is the total number of subjects in group g, and n_{+i} is the total number of subjects in category i across both groups, and n is the total sample size.

- Under the null hypothesis, X^2 follows a chi-square distribution with $C-1$ degrees of freedom.
- Under the alternative hypothesis, X^2 follows a non-central chi-square distribution with non-centrality parameter

  $$\lambda = nr(1-r) \sum_{i=1}^{C} \frac{(\pi_{1i} - \pi_{2i})^2}{r\pi_{1i} + (1-r)\pi_{2i}}$$

  where r is the randomization probability for the active treatment.

The sample size is chosen such that the power to reject the null hypothesis is at least $1 - \beta$ for a given significance level α.

Value

An S3 class `designTwoMultinom` object with the following components:

- `power`: The power to reject the null hypothesis.
- `alpha`: The two-sided significance level.
- `n`: The maximum number of subjects.
- `ncats`: The number of categories of the multinomial response.
- `pi1`: The prevalence of each category for the treatment group.
- `pi2`: The prevalence of each category for the control group.
- `effectsize`: The effect size for the chi-square test.
- `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
- `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignTwoMultinom(
  beta = 0.1, ncats = 3, pi1 = c(0.3, 0.35),
  pi2 = c(0.2, 0.3), alpha = 0.05))
```

`getDesignTwoOrdinal` *Power and Sample Size for the Wilcoxon Test for Two-Sample Ordinal Response*

Description

Obtains the power given sample size or obtains the sample size given power for the Wilcoxon test for two-sample ordinal response.

Usage

```
getDesignTwoOrdinal(
  beta = NA_real_,
  n = NA_real_,
  ncats = NA_integer_,
  pi1 = NA_real_,
  pi2 = NA_real_,
  allocationRatioPlanned = 1,
  rounding = TRUE,
  alpha = 0.05
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`ncats`	The number of categories of the ordinal response.
`pi1`	The prevalence of each category for the treatment group. Only need to specify the valued for the first `ncats-1` categories.
`pi2`	The prevalence of each category for the control group. Only need to specify the valued for the first `ncats-1` categories.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`alpha`	The significance level. Defaults to 0.025.

Details

A two-sample ordinal response design is used to test whether the ordinal response distributions differ between two treatment arms. Let π_{gi} denote the prevalence of category i in group g, where $g = 1$ represents the treatment group and $g = 2$ represents the control group.

The parameter of interest is

$$\theta = \sum_{i=1}^{C} w_i(\pi_{1i} - \pi_{2i})$$

where w_i is the midrank score for category i. The Z-test statistic is given by

$$Z = \hat{\theta}/\sqrt{Var(\hat{\theta})}$$

where $\hat{\theta}$ is the estimate of θ.

The midrank score w_i for category i is calculated as:

$$w_i = \sum_{j=1}^{i} \pi_j - 0.5\pi_i$$

where $\pi_i = r\pi_{1i} + (1-r)\pi_{2i}$ denotes the average prevalence of category i across both groups, and r is the randomization probability for the active treatment.

To understand the midrank score, consider $n\pi_i$ subjects in category i. The midrank score is the average rank of these subjects:

$$s_i = \frac{1}{n\pi_i} \sum_{j=1}^{n\pi_i} (n\pi_1 + \cdots + n\pi_{i-1} + j)$$

This simplifies to

$$s_i = n\left(\sum_{j=1}^{i} \pi_j - 0.5\pi_i\right) + \frac{1}{2}$$

By dividing by n and ignoring $\frac{1}{2n}$, we obtain the midrank score w_i.

The variance of $\hat{\theta}$ can be derived from the multinomial distributions and is given by

$$Var(\hat{\theta}) = \frac{1}{n} \sum_{g=1}^{2} \frac{1}{r_g} \left\{ \sum_{i=1}^{C} w_i^2 \pi_{gi} - \left(\sum_{i=1}^{C} w_i \pi_{gi} \right)^2 \right\}$$

where r_g is the randomization probability for group g.

The sample size is chosen such that the power to reject the null hypothesis is at least $1-\beta$ for a given significance level α.

Value

An S3 class `designTwoOrdinal` object with the following components:

- `power`: The power to reject the null hypothesis.
- `alpha`: The two-sided significance level.
- `n`: The maximum number of subjects.
- `ncats`: The number of categories of the ordinal response.
- `pi1`: The prevalence of each category for the treatment group.
- `pi2`: The prevalence of each category for the control group.
- `meanscore1`: The mean midrank score for the treatment group.
- `meanscore2`: The mean midrank score for the control group.
- `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
- `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
(design1 <- getDesignTwoOrdinal(
  beta = 0.1, ncats = 4, pi1 = c(0.55, 0.3, 0.1),
  pi2 = c(0.214, 0.344, 0.251), alpha = 0.025))
```

`getDesignTwoWayANOVA` *Power and Sample Size for Two-Way ANOVA*

Description

Obtains the power and sample size for two-way analysis of variance.

Usage

```
getDesignTwoWayANOVA(
  beta = NA_real_,
  n = NA_real_,
  nlevelsA = 2,
  nlevelsB = 2,
  means = NA_real_,
  stDev = 1,
  rounding = TRUE,
  alpha = 0.05
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`nlevelsA`	The number of groups for Factor A.
`nlevelsB`	The number of levels for Factor B.
`means`	The matrix of treatment means for Factors A and B combination.
`stDev`	The common standard deviation.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`alpha`	The two-sided significance level. Defaults to 0.05.

Value

An S3 class `designTwoWayANOVA` object with the following components:

- `alpha`: The two-sided significance level.
- `nlevelsA`: The number of levels for Factor A.
- `nlevelsB`: The number of levels for Factor B.
- `means`: The matrix of treatment group means.
- `stDev`: The common standard deviation.
- `effectsizeA`: The effect size for Factor A.
- `effectsizeB`: The effect size for Factor B.
- `effectsizeAB`: The effect size for Factor A and Factor B interaction.
- `rounding`: Whether to round up sample size.
- `powerdf`: The data frame containing the power and sample size results. It has the following variables:
  - `n`: The sample size.
  - `powerA`: The power to reject the null hypothesis that there is no difference among Factor A levels.
  - `powerB`: The power to reject the null hypothesis that there is no difference among Factor B levels.
  - `powerAB`: The power to reject the null hypothesis that there is no interaction between Factor A and Factor B.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignTwoWayANOVA(
  beta = 0.1, nlevelsA = 2, nlevelsB = 2,
  means = matrix(c(0.5, 4.7, 0.4, 6.9), 2, 2, byrow = TRUE),
  stDev = 2, alpha = 0.05))
```

`getDesignUnorderedBinom` *Power and Sample Size for Unordered Multi-Sample Binomial Response*

Description

Obtains the power given sample size or obtains the sample size given power for the chi-square test for unordered multi-sample binomial response.

Usage

```
getDesignUnorderedBinom(
  beta = NA_real_,
  n = NA_real_,
  ngroups = NA_integer_,
  pi = NA_real_,
  allocationRatioPlanned = NA_integer_,
  rounding = TRUE,
  alpha = 0.05
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`ngroups`	The number of treatment groups.
`pi`	The response probabilities for the treatment groups.
`allocationRatioPlanned`	Allocation ratio for the treatment groups.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`alpha`	The two-sided significance level. Defaults to 0.05.

Details

A multi-sample binomial response design is used to test whether the response probabilities differ among multiple treatment arms. Let π_g denote the response probability in group $g = 1, \ldots, G$, where G is the total number of treatment groups.

The chi-square test statistic is given by

$$X^2 = \sum_{g=1}^{G} \sum_{i=1}^{2} \frac{(n_{gi} - n_{g+}n_{+i}/n)^2}{n_{g+}n_{+i}/n}$$

where n_{gi} is the number of subjects in category i for group g, n_{g+} is the total number of subjects in group g, and n_{+i} is the total number of subjects in category i across all groups, and n is the total sample size.

Let r_g denote the randomization probability for group g, and define the weighted average response probability across all groups as

$$\bar{\pi} = \sum_{g=1}^{G} r_g \pi_g$$

- Under the null hypothesis, X^2 follows a chi-square distribution with $G-1$ degrees of freedom.
- Under the alternative hypothesis, X^2 follows a non-central chi-square distribution with non-centrality parameter

$$\lambda = n \sum_{g=1}^{G} \frac{r_g(\pi_g - \bar{\pi})^2}{\bar{\pi}(1 - \bar{\pi})}$$

The sample size is chosen such that the power to reject the null hypothesis is at least $1 - \beta$ for a given significance level α.

Value

An S3 class `designUnorderedBinom` object with the following components:

- `power`: The power to reject the null hypothesis.
- `alpha`: The two-sided significance level.
- `n`: The maximum number of subjects.
- `ngroups`: The number of treatment groups.
- `pi`: The response probabilities for the treatment groups.
- `effectsize`: The effect size for the chi-square test.
- `allocationRatioPlanned`: Allocation ratio for the treatment groups.
- `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(design1 <- getDesignUnorderedBinom(
  beta = 0.1, ngroups = 3, pi = c(0.1, 0.25, 0.5), alpha = 0.05))
```

`getDesignUnorderedMultinom`

Power and Sample Size for Unordered Multi-Sample Multinomial Response

Description

Obtains the power given sample size or obtains the sample size given power for the chi-square test for unordered multi-sample multinomial response.

Usage

```
getDesignUnorderedMultinom(
  beta = NA_real_,
  n = NA_real_,
  ngroups = NA_integer_,
  ncats = NA_integer_,
  pi = NA_real_,
  allocationRatioPlanned = NA_integer_,
  rounding = TRUE,
  alpha = 0.05
)
```

Arguments

`beta`	The type II error.
`n`	The total sample size.
`ngroups`	The number of treatment groups.
`ncats`	The number of categories of the multinomial response.
`pi`	The matrix of response probabilities for the treatment groups. It should have `ngroups` rows and `ncats-1` or `ncats` columns.
`allocationRatioPlanned`	Allocation ratio for the treatment groups.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`alpha`	The two-sided significance level. Defaults to 0.05.

Details

A multi-sample multinomial response design is used to test whether the response probabilities differ among multiple treatment arms. Let π_{gi} denote the response probability for category $i = 1, \ldots, C$ in group $g = 1, \ldots, G$, where G is the total number of treatment groups, and C is the total number of categories for the response variable.

The chi-square test statistic is given by

$$X^2 = \sum_{g=1}^G \sum_{i=1}^C \frac{(n_{gi} - n_{g+}n_{+i}/n)^2}{n_{g+}n_{+i}/n}$$

where n_{gi} is the number of subjects in category i for group g, n_{g+} is the total number of subjects in group g, and n_{+i} is the total number of subjects in category i across all groups, and n is the total sample size.

Let r_g denote the randomization probability for group g, and define the weighted average response probability for category i across all groups as

$$\bar{\pi}_i = \sum_{g=1}^G r_g \pi_{gi}$$

- Under the null hypothesis, X^2 follows a chi-square distribution with $(G-1)(C-1)$ degrees of freedom.
- Under the alternative hypothesis, X^2 follows a non-central chi-square distribution with non-centrality parameter

$$\lambda = n \sum_{g=1}^G \sum_{i=1}^C \frac{r_g(\pi_{gi} - \bar{\pi}_i)^2}{\bar{\pi}_i}$$

The sample size is chosen such that the power to reject the null hypothesis is at least $1 - \beta$ for a given significance level α.

Value

An S3 class `designUnorderedMultinom` object with the following components:

- `power`: The power to reject the null hypothesis.
- `alpha`: The two-sided significance level.
- `n`: The maximum number of subjects.
- `ngroups`: The number of treatment groups.
- `ncats`: The number of categories of the multinomial response.
- `pi`: The response probabilities for the treatment groups.
- `effectsize`: The effect size for the chi-square test.
- `allocationRatioPlanned`: Allocation ratio for the treatment groups.
- `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
(design1 <- getDesignUnorderedMultinom(
  beta = 0.1, ngroups = 3, ncats = 4,
  pi = matrix(c(0.230, 0.320, 0.272,
                0.358, 0.442, 0.154,
                0.142, 0.036, 0.039),
              3, 3, byrow = TRUE),
  allocationRatioPlanned = c(2, 2, 1),
  alpha = 0.05))
```

`getDesignWilcoxon` *Group Sequential Design for Two-Sample Wilcoxon Test*

Description

Obtains the power given sample size or obtains the sample size given power for a group sequential design for two-sample Wilcoxon test.

Usage

```
getDesignWilcoxon(
  beta = NA_real_,
  n = NA_real_,
  pLarger = 0.6,
  allocationRatioPlanned = 1,
  rounding = TRUE,
```

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

Arguments

`beta`	The type II error.
`n`	The total sample size.
`pLarger`	The probability that a randomly chosen sample from the treatment group is larger than a randomly chosen sample from the control group under the alternative hypothesis.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`rounding`	Whether to round up sample size. Defaults to 1 for sample size rounding.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Fixed prior to the trial. Defaults to `(1:kMax) / kMax` if left unspecified.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage. Defaults to true if left unspecified.
`futilityStopping`	Indicators of whether futility stopping is allowed at each stage. Defaults to true if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, "user" for user defined spending, and "none" for no early efficacy stopping. Defaults to "sfOF".

`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.

Details

The Mann-Whitney U test is a non-parametric test for the difference in location between two independent groups. It is also known as the Wilcoxon rank-sum test. The test is based on the ranks of the data rather than the actual values, making it robust to outliers and non-normal distributions. The test statistic is the number of times a randomly chosen sample from the treatment group is larger than a randomly chosen sample from the control group, i.e.,

$$W_{XY} = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} I(X_i > Y_j)$$

where X_i and Y_j are the samples from the treatment and control groups, respectively. The test is often used when the data do not meet the assumptions of the t-test, such as non-normality or unequal variances. The test is also applicable to ordinal data. The test is one-sided, meaning that it only tests whether the treatment group is larger than the control group. Asymptotically,

$$\frac{W_{XY} - n_1 n_2/2}{\sqrt{n_1 n_2 (n+1)/12}} \sim N(0,1) \quad \text{under} H_0$$

where n_1 and n_2 are the sample sizes of the treatment and control groups, respectively, and $n = n_1 + n_2$. Let $\theta = P(X > Y)$, and $\hat{\theta} = \frac{1}{nm} W_{XY}$. It follows that

$$\sqrt{n}(\hat{\theta} - 1/2) \sim N\left(0, \frac{1}{12r(1-r)}\right) \quad \text{under} H_0$$

where $r = n_1/(n_1 + n_2)$ is the randomization probability for the active treatment group. This formulation allows for group sequential testing with futility stopping and efficacy stopping.

Value

An S3 class `designWilcoxon` 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*sqrt(information)`.
    - `inflationFactor`: The inflation factor (relative to the fixed design).
    - `numberOfSubjects`: The maximum number of subjects.
    - `expectedNumberOfSubjectsH1`: The expected number of subjects under H1.
    - `expectedNumberOfSubjectsH0`: The expected number of subjects under H0.
    - `pLarger`: The probability that a randomly chosen sample from the treatment group is larger than a randomly chosen sample from the control group under the alternative hypothesis.
- `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.
    - `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.
    - `efficacyPLarger`: The efficacy boundaries on the proportion of pairs of samples from the two treatment groups with the sample from the treatment group greater than that from the control group.

  - `futilityPLarger`: The futility boundaries on the proportion of pairs of samples from the two treatment groups with the sample from the treatment group greater than that from the control group.
  - `numberOfSubjects`: The number of subjects.
- `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.
  - `allocationRatioPlanned`: Allocation ratio for the active treatment versus control.
  - `rounding`: Whether to round up sample size.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Example 1: fixed design
(design1 <- getDesignWilcoxon(
  beta = 0.1, n = NA,
  pLarger = pnorm((8 - 2)/sqrt(2*25^2)), alpha = 0.025))

# Example 2: group sequential design
(design2 <- getDesignWilcoxon(
  beta = 0.1, n = NA,
  pLarger = pnorm((8 - 2)/sqrt(2*25^2)), alpha = 0.025,
  kMax = 3, typeAlphaSpending = "sfOF"))
```

`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
)
```

Arguments

`nevents`	The target number of events.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.

`followupTime`	Follow-up time for the last enrolled subjects. Must be provided for fixed follow-up design.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`npoints`	The number of accrual duration time points. Defaults to 23.

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

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

`parameterAlphaSpending`	The parameter value for the alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`userAlphaSpending`	The user defined alpha spending. Cumulative alpha spent up to each stage.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. 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`.
`hazardRatioH0`	Hazard ratio under the null hypothesis for the active treatment versus control. Defaults to 1 for superiority test.
`hazardRatio`	Hazard ratio under the alternative hypothesis for the active treatment versus control.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`rounding`	Whether to round up the number of events. Defaults to 1 for rounding.

Value

The required number of events.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

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

`getRCI` *Repeated Confidence Interval for Group Sequential Design*

Description

Obtains the repeated confidence interval for a group sequential trial.

Usage

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

Arguments

`L`	The look of interest.
`zL`	The z-test statistic at the look.
`IMax`	The maximum information of the trial.
`informationRates`	The information rates up to look `L`.
`efficacyStopping`	Indicators of whether efficacy stopping is allowed at each stage up to look `L`. Defaults to true if left unspecified.
`criticalValues`	The upper boundaries on the z-test statistic scale for efficacy stopping up to look `L`.
`alpha`	The significance level. Defaults to 0.025.
`typeAlphaSpending`	The type of alpha spending. One of the following: "OF" for O'Brien-Fleming boundaries, "P" for Pocock boundaries, "WT" for Wang & Tsiatis boundaries, "sfOF" for O'Brien-Fleming type spending function, "sfP" for Pocock type spending function, "sfKD" for Kim & DeMets spending function, "sfHSD" for Hwang, Shi & DeCani spending function, and "none" for no early efficacy stopping. Defaults to "sfOF".
`parameterAlphaSpending`	The parameter value of alpha spending. Corresponds to Delta for "WT", rho for "sfKD", and gamma for "sfHSD".
`spendingTime`	The error spending time up to look `L`. Defaults to missing, in which case, it is the same as `informationRates`.

Value

A data frame with the following components:

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

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

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

Examples

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

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

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

`hazard_pd` *Hazard Function for Progressive Disease (PD)*

Description

Computes the hazard function of a piecewise exponential (pwexp) distribution for progressive disease (PD), such that the resulting hazard function for progression-free survival (PFS) closely matches a given pwexp hazard for PFS.

Usage

```
hazard_pd(
  piecewiseSurvivalTime = 0,
  hazard_pfs = 0.0578,
  hazard_os = 0.02,
  corr_pd_os = 0.5
)
```

Arguments

`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.
`hazard_pfs`	A scalar or numeric vector specifying the hazard(s) for PFS based on a pwexp distribution.
`hazard_os`	A scalar or numeric vector specifying the hazard(s) for overall survival (OS) based on a pwexp distribution.
`corr_pd_os`	A numeric value specifying the correlation between PD and OS times.

Details

This function determines the hazard vector λ_{pd} for the pwexp distribution of PD, so that the implied survival function for PFS time, $T_{\text{pfs}} = \min(T_{\text{pd}}, T_{\text{os}})$, closely matches the specified pwexp distribution for PFS with hazard vector λ_{pfs}.

To achieve this, we simulate $(Z_{\text{pd}}, Z_{\text{os}})$ from a standard bivariate normal distribution with correlation ρ. Then, $U_{\text{pd}} = \Phi(Z_{\text{pd}})$ and $U_{\text{os}} = \Phi(Z_{\text{os}})$ are generated, where Φ denotes the standard normal CDF.

The times to PD and OS are obtained via the inverse transform method using quantile functions of the pwexp distribution:

$$T_{\text{pd}} = \text{qpwexp}(U_{\text{pd}}, u, \lambda_{\text{pd}})$$

$$T_{\text{os}} = \text{qpwexp}(U_{\text{os}}, u, \lambda_{\text{os}})$$

where `u` = `piecewiseSurvivalTime`.

The function solves for λ_{pd} such that the survival function of T_{pfs} closely matches that of a pwexp distribution with hazard λ_{pfs}:

$$P(\min(T_{\text{pd}}, T_{\text{os}}) > t) = S_{\text{pfs}}(t)$$

Since

$$Z_{\text{pd}} = \Phi^{-1}(\text{ppwexp}(T_{\text{pd}}, u, \lambda_{\text{pd}}))$$

and

$$Z_{\text{os}} = \Phi^{-1}(\text{ppwexp}(T_{\text{os}}, u, \lambda_{\text{os}}))$$

we have

$$P(\min(T_{\text{pd}}, T_{\text{os}}) > t) = P(Z_{\text{pd}} > \Phi^{-1}(\text{ppwexp}(t, u, \lambda_{\text{pd}})), Z_{\text{os}} > \Phi^{-1}(\text{ppwexp}(t, u, \lambda_{\text{os}})))$$

while

$$S_{\text{pfs}}(t) = 1 - \text{ppwexp}(t, u, \lambda_{\text{pfs}})$$

Matching is performed sequentially at the internal cutpoints $u_2, ..., u_J$ and at the point $u_J + \log(2)/\lambda_{\text{pfs},J}$ for the final interval to solve for $\lambda_{\text{pd},1}, \ldots, \lambda_{\text{pd},J-1}$ and $\lambda_{\text{pd},J}$, respectively.

Value

A numeric vector representing the estimated hazard rates for the pwexp distribution of PD.

Author(s)

Kaifeng Lu (<`kaifenglu@gmail.com`>)

Examples

```
u <- c(0, 1, 3, 4)
lambda1 <- c(0.0151, 0.0403, 0.0501, 0.0558)
lambda2 <- 0.0145
rho <- 0.5
hazard_pd(u, lambda1, lambda2, rho)
```

`hazard_sub` *Hazard Function for Sub Population*

Description

Computes the hazard function of a piecewise exponential (pwexp) distribution for the biomarker negative sub population, such that the resulting survival function for the ITT population closely matches a given pwexp survival function.

Usage

```
hazard_sub(
  piecewiseSurvivalTime = 0,
  hazard_itt = 0.0578,
  hazard_pos = 0.02,
  p_pos = 0.5
)
```

Arguments

`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.
`hazard_itt`	A scalar or numeric vector specifying the hazard(s) for the ITT population based on a pwexp distribution.

`hazard_pos`	A scalar or numeric vector specifying the hazard(s) for the biomarker positive sub population based on a pwexp distribution.
`p_pos`	A numeric value specifying the prevalence of the biomarker positive sub population.

Details

This function determines the hazard vector λ_{neg} for the pwexp distribution of the biomarker negative sub population, so that the implied survival function for the ITT population closely matches the specified pwexp distribution with hazard vector λ_{itt}.

Let p_{pos} be the prevalence of the biomarker positive sub population, then the survival function for the ITT population is given by

$$S_{\text{itt}}(t) = p_{\text{pos}} S_{\text{pos}}(t) + (1 - p_{\text{pos}}) S_{\text{neg}}(t)$$

where $S_{\text{pos}}(t)$ and $S_{\text{neg}}(t)$ are the survival functions for the biomarker positive and biomarker negative sub populations, respectively.

Matching is performed sequentially at the internal cutpoints $u_2, ..., u_J$ and at the point $u_J + \log(2)/\lambda_{\text{itt},J}$ for the final interval to solve for $\lambda_{\text{neg},1}, \ldots, \lambda_{\text{neg},J-1}$ and $\lambda_{\text{neg},J}$, respectively.

Value

A numeric vector representing the estimated hazard rates for the pwexp distribution of the biomarker negative sub population.

Author(s)

Kaifeng Lu (<`kaifenglu@gmail.com`>)

Examples

```
u <- c(0, 1, 3, 4)
lambda_itt <- c(0.0151, 0.0403, 0.0501, 0.0558)
lambda_pos <- c(0.0115, 0.0302, 0.0351, 0.0404)
p_pos <- 0.3
hazard_sub(u, lambda_itt, lambda_pos, p_pos)
```

`heart` *Stanford heart transplant data from the survival package*

Description

Survival of patients on the waiting list for the Stanford heart transplant program.

`start, stop, event`	entry and exit time and status for the time interval
`age`	age-48 years

`year` year of acceptance (in years after Nov 1, 1967)

`surgery` prior bypass surgery 1=yes, 0=no

`transplant` received transplant 1=yes, 0=no

`id` patient id

Usage

```
heart
```

Format

An object of class `data.frame` with 172 rows and 8 columns.

`hedgesg` *Hedges' g Effect Size*

Description

Obtains Hedges' g estimate and confidence interval of effect size.

Usage

```
hedgesg(tstat, m, ntilde, cilevel = 0.95)
```

Arguments

`tstat`	The value of the t-test statistic for comparing two treatment conditions.
`m`	The degrees of freedom for the t-test.
`ntilde`	The normalizing sample size to convert the standardized treatment difference to the t-test statistic, i.e., `tstat` = `sqrt(ntilde)*meanDiff/stDev`.
`cilevel`	The confidence interval level. Defaults to 0.95.

Details

Hedges' g is an effect size measure commonly used in meta-analysis to quantify the difference between two groups. It's an improvement over Cohen's d, particularly when dealing with small sample sizes.

The formula for Hedges' g is

$$g = c(m)d$$

where d is Cohen's d effect size estimate, and $c(m)$ is the bias correction factor,

$$d = (\hat{\mu}_1 - \hat{\mu}_2)/\hat{\sigma}$$

$$c(m) = 1 - \frac{3}{4m - 1}.$$

Since $c(m) < 1$, Cohen's d overestimates the true effect size, $\delta = (\mu_1 - \mu_2)/\sigma$. Since

$$t = \sqrt{\tilde{n}} d$$

we have

$$g = \frac{c(m)}{\sqrt{\tilde{n}}} t$$

where t has a noncentral t distribution with m degrees of freedom and noncentrality parameter $\sqrt{\tilde{n}}\delta$.

The asymptotic variance of g can be approximated by

$$Var(g) = \frac{1}{\tilde{n}} + \frac{g^2}{2m}.$$

The confidence interval for δ can be constructed using normal approximation.

For two-sample mean difference with sample size n_1 for the treatment group and n_2 for the control group, we have $\tilde{n} = \frac{n_1 n_2}{n_1 + n_2}$ and $m = n_1 + n_2 - 2$ for pooled variance estimate.

Value

A data frame with the following variables:

- `tstat`: The value of the `t` test statistic.
- `m`: The degrees of freedom for the t-test.
- `ntilde`: The normalizing sample size to convert the standardized treatment difference to the t-test statistic.
- `g`: Hedges' g effect size estimate.
- `varg`: Variance of `g`.
- `lower`: The lower confidence limit for effect size.
- `upper`: The upper confidence limit for effect size.
- `cilevel`: The confidence interval level.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

Larry V. Hedges. Distribution theory for Glass's estimator of effect size and related estimators. Journal of Educational Statistics 1981; 6:107-128.

Examples

```
n1 = 7
n2 = 8
meanDiff = 0.444
stDev = 1.201
m = n1+n2-2
```

```
ntilde = n1*n2/(n1+n2)
tstat = sqrt(ntilde)*meanDiff/stDev

hedgesg(tstat, m, ntilde)
```

`immdef` *Simulated CONCORDE trial data from the rpsftm package*

Description

Patients were randomly assigned to receive treatment immediately or deferred, and those in the deferred arm could cross over and receive treatment. The primary endpoint was time to disease progression.

`id` Patient identification number

`def` Indicator that the participant was assigned to the deferred treatment arm

`imm` Indicator that the participant was assigned to the immediate treatment arm

`censyrs` The censoring time, in years, corresponding to the close of study minus the time of entry for each patient

`xo` Indicator that crossover occurred

`xoyrs` The time, in years, from entry to switching, or 0 for patients in the immediate arm

`prog` Indicator of disease progression (1), or censoring (0)

`progyrs` Time, in years, from entry to disease progression or censoring

`entry` The time of entry into the study, measured in years from the date of randomisation

Usage

```
immdef
```

Format

An object of class `data.frame` with 1000 rows and 9 columns.

`ingots` *The binary data from Cox and Snell (1989, pp. 10-11).*

Description

The dataset consits of the number of ingots not ready for rolling and the number of ingots ready for rolling for a number of combinations of heating time and soaking time.

Usage

```
ingots
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 25 rows and 4 columns.

Details

`Heat` The heating time

`Soak` The soaking time

`NotReady` Response indicator, with a value 1 for units not ready for rolling (event) and a value of 0 for units ready for rolling (nonevent)

`Freq` The frequency of occurrence of each combination of `Heat`, `Soak`, and `NotReady`

`kmdiff` *Estimate of Milestone Survival Difference*

Description

Obtains the estimate of milestone survival difference between two treatment groups.

Usage

```
kmdiff(
  data,
  rep = "",
  stratum = "",
  treat = "treat",
  time = "time",
  event = "event",
  milestone = NA_real_,
  survDiffH0 = 0,
  conflev = 0.95
)
```

Arguments

`data`	The input data frame that contains the following variables: • `rep`: The replication for by-group processing. • `stratum`: The stratum. • `treat`: The treatment. • `time`: The possibly right-censored survival time. • `event`: The event indicator.
`rep`	The name of the replication variable in the input data.
`stratum`	The name of the stratum variable in the input data.
`treat`	The name of the treatment variable 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.
`milestone`	The milestone time at which to calculate the survival probability.
`survDiffH0`	The difference in milestone survival probabilities under the null hypothesis. Defaults to 0 for superiority test.
`conflev`	The level of the two-sided confidence interval for the difference in milestone survival probabilities. Defaults to 0.95.

Value

A data frame with the following variables:

- `rep`: The replication.
- `milestone`: The milestone time relative to randomization.
- `survDiffH0`: The difference in milestone survival probabilities under the null hypothesis.
- `surv1`: The estimated milestone survival probability for the treatment group.
- `surv2`: The estimated milestone survival probability for the control group.
- `survDiff`: The estimated difference in milestone survival probabilities.
- `vsurv1`: The variance for surv1.
- `vsurv2`: The variance for surv2.
- `vsurvDiff`: The variance for survDiff.
- `survDiffZ`: The Z-statistic value.
- `survDiffPValue`: The one-sided p-value.
- `lower`: The lower bound of confidence interval.
- `upper`: The upper bound of confidence interval.
- `conflev`: The level of confidence interval.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
df <- kmdiff(data = rawdata, rep = "iterationNumber",
             stratum = "stratum", treat = "treatmentGroup",
             time = "timeUnderObservation", event = "event",
             milestone = 12)
head(df)
```

`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,
  keep_censor = 0L
)
```

Arguments

`data`	The input data frame that contains the following variables: • `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 asin(sqrt(survival)).
`conflev`	The level of the two-sided confidence interval for the survival probabilities. Defaults to 0.95.
`keep_censor`	Whether to retain the censoring time in the output data frame.

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.
- `ncensor`: The number of censored subjects.
- `survival`: The Kaplan-Meier estimate of the survival probability.
- `stderr`: The standard error of the estimated survival probability based on the Greendwood 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")
```

`kmpower` *Power for Difference in Milestone Survival Probabilities*

Description

Estimates the power for testing the difference in milestone survival probabilities in a two-sample survival design.

Usage

```
kmpower(
  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_,
  milestone = NA_real_,
  survDiffH0 = 0,
  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,
  spendingTime = NA_real_,
  studyDuration = NA_real_
)
```

Arguments

`kMax`	The maximum number of stages.
`informationRates`	The information rates. 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".
`milestone`	The milestone time at which to calculate the survival probability.
`survDiffH0`	The difference in milestone survival probabilities under the null hypothesis. Defaults to 0 for superiority test.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.

`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.
`studyDuration`	Study duration for fixed follow-up design. Defaults to missing, which is to be replaced with the sum of `accrualDuration` and `followupTime`. If provided, the value is allowed to be less than the sum of `accrualDuration` and `followupTime`.

Value

An S3 class `kmpower` 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.
  - `numbeOfSubjects`: The total number of subjects.
  - `studyDuration`: The total study duration.
  - `information`: The maximum information.
  - `expectedNumberOfEvents`: The expected number of events.
  - `expectedNumberOfSubjects`: The expected number of subjects.
  - `expectedStudyDuration`: The expected study duration.
  - `expectedInformation`: The expected information.
  - `accrualDuration`: The accrual duration.
  - `followupTime`: The follow-up duration.
  - `fixedFollowup`: Whether a fixed follow-up design is used.
  - `kMax`: The number of stages.
  - `milestone`: The milestone time relative to randomization.
  - `survDiffH0`: The difference in milestone survival probabilities under the null hypothesis.
  - `surv1`: The milestone survival probability for the treatment group.
  - `surv2`: The milestone survival probability for the control group.
  - `survDiff`: The difference in milestone survival probabilities, equal to `surv1 - surv2`.
- `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.

  - `numberOfMilestone`: The number of subjects reaching milestone.
  - `analysisTime`: The average time since trial start.
  - `efficacySurvDiff`: The efficacy boundaries on the survival difference scale.
  - `futilitySurvDiff`: The futility boundaries on the survival difference scale.
  - `efficacyP`: The efficacy boundaries on the p-value scale.
  - `futilityP`: The futility boundaries on the p-value scale.
  - `information`: The cumulative information.
  - `efficacyStopping`: Whether to allow efficacy stopping.
  - `futilityStopping`: Whether to allow futility stopping.
- `settings`: A list containing the following input parameters: `typeAlphaSpending`, `parameterAlphaSpending`, `userAlphaSpending`, `typeBetaSpending`, `parameterBetaSpending`, `allocationRatioPlanned`, `accrualTime`, `accuralIntensity`, `piecewiseSurvivalTime`, `stratumFraction`, `lambda1`, `lambda2`, `gamma1`, `gamma2`, and `spendingTime`.
- `byTreatmentCounts`: A list containing the following counts by treatment group:
  - `numberOfEvents1`: The number of events by stage for the treatment group.
  - `numberOfDropouts1`: The number of dropouts by stage for the treatment group.
  - `numberOfSubjects1`: The number of subjects by stage for the treatment group.
  - `numberOfMilestone1`: The number of subjects reaching milestone by stage for the active 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.
  - `numberOfMilestone2`: The number of subjects reaching milestone by stage for the control group.
  - `expectedNumberOfEvents1`: The expected number of events for the treatment group.
  - `expectedNumberOfDropouts1`: The expected number of dropouts for the active treatment group.
  - `expectedNumberOfSubjects1`: The expected number of subjects for the active treatment group.
  - `expectedNumberOfMilestone1`: The expected number of subjects reaching milestone for the active 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.
  - `expectedNumberOfMilestone2`: The expected number of subjects reaching milestone 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.

kmpower(kMax = 2, informationRates = c(0.8, 1),
        alpha = 0.025, typeAlphaSpending = "sfOF",
        milestone = 18,
        allocationRatioPlanned = 1, accrualTime = seq(0, 8),
        accrualIntensity = 26/9*seq(1, 9),
        piecewiseSurvivalTime = c(0, 6),
        stratumFraction = c(0.2, 0.8),
        lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
        lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
        gamma1 = -log(1-0.05)/12,
        gamma2 = -log(1-0.05)/12, accrualDuration = 22,
        followupTime = 18, fixedFollowup = FALSE)
```

`kmpower1s` *Power for One-Sample Milestone Survival Probability*

Description

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

Usage

```
kmpower1s(
  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_,
  milestone = NA_real_,
  survH0 = NA_real_,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  lambda = NA_real_,
```

```
  gamma = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  spendingTime = NA_real_,
  studyDuration = NA_real_
)
```

Arguments

`kMax`	The maximum number of stages.
`informationRates`	The information rates. 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".
`milestone`	The milestone time at which to calculate the survival probability.
`survH0`	The milestone survival probability under the null hypothesis.

`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda`	A vector of hazard rates for the event in each analysis time interval by stratum under the alternative hypothesis.
`gamma`	The hazard rate for exponential dropout or a vector of hazard rates for piecewise exponential dropout. Defaults to 0 for no dropout.
`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.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.
`studyDuration`	Study duration for fixed follow-up design. Defaults to missing, which is to be replaced with the sum of `accrualDuration` and `followupTime`. If provided, the value is allowed to be less than the sum of `accrualDuration` and `followupTime`.

Value

An S3 class `kmpower1s` object with 3 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.
    - `numbeOfSubjects`: The total number of subjects.
    - `numberOfMilestone`: The total number of subjects reaching milestone.
    - `studyDuration`: The total study duration.
    - `information`: The maximum information.
    - `expectedNumberOfEvents`: The expected number of events.
    - `expectedNumberOfSubjects`: The expected number of subjects.
    - `expectedNumberOfMilestone`: The expected number of subjects reaching milestone.
    - `expectedStudyDuration`: The expected study duration.
    - `expectedInformation`: The expected information.
    - `accrualDuration`: The accrual duration.

  - `followupTime`: The follow-up duration.
  - `fixedFollowup`: Whether a fixed follow-up design is used.
  - `kMax`: The number of stages.
  - `milestone`: The milestone time to calculate the survival probability.
  - `survH0`: The milestone survival probability under the null hypothesis.
  - `surv`: The milestone survival probability under the alternative hypothesis.
- `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.
  - `numberOfMilestone`: The number of subjects reaching milestone.
  - `analysisTime`: The average time since trial start.
  - `efficacySurv`: The efficacy boundaries on the milestone survival probability scale.
  - `futilitySurv`: The futility boundaries on the milestone survival probability scale.
  - `efficacyP`: The efficacy boundaries on the p-value scale.
  - `futilityP`: The futility boundaries on the p-value scale.
  - `information`: The cumulative information.
  - `efficacyStopping`: Whether to allow efficacy stopping.
  - `futilityStopping`: Whether to allow futility stopping.
- `settings`: A list containing the following input parameters: `typeAlphaSpending`, `parameterAlphaSpending`, `userAlphaSpending`, `typeBetaSpending`, `parameterBetaSpending`, `accrualTime`, `accuralIntensity`, `piecewiseSurvivalTime`, `stratumFraction`, `lambda`, `gamma`, and `spendingTime`.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

See Also

`kmstat`

Examples

```
kmpower1s(kMax = 2, informationRates = c(0.8, 1),
          alpha = 0.025, typeAlphaSpending = "sfOF",
          milestone = 18, survH0 = 0.30,
          accrualTime = seq(0, 8),
```

```
            accrualIntensity = 26/9*seq(1, 9),
            piecewiseSurvivalTime = c(0, 6),
            stratumFraction = c(0.2, 0.8),
            lambda = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
            gamma = -log(1-0.05)/12, accrualDuration = 22,
            followupTime = 18, fixedFollowup = FALSE)
```

`kmpowerequiv` *Power for Equivalence in Milestone Survival Probability Difference*

Description

Obtains the power for equivalence in milestone survival probability difference.

Usage

```
kmpowerequiv(
  kMax = 1L,
  informationRates = NA_real_,
  criticalValues = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  milestone = NA_real_,
  survDiffLower = NA_real_,
  survDiffUpper = 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,
  spendingTime = NA_real_,
  studyDuration = NA_real_
)
```

Arguments

`kMax`	The maximum number of stages.

`informationRates`	The information rates. Defaults to `(1:kMax) / kMax` if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`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.
`milestone`	The milestone time at which to calculate the survival probability.
`survDiffLower`	The lower equivalence limit of milestone survival probability difference.
`survDiffUpper`	The upper equivalence limit of milestone survival probability difference.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.

`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.
`studyDuration`	Study duration for fixed follow-up design. Defaults to missing, which is to be replaced with the sum of `accrualDuration` and `followupTime`. If provided, the value is allowed to be less than the sum of `accrualDuration` and `followupTime`.

Value

An S3 class `kmpowerequiv` 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.
  - `numberOfSubjects`: The total number of subjects.
  - `studyDuration`: The total study duration.
  - `information`: The maximum information.
  - `expectedNumberOfEvents`: The expected number of events.
  - `expectedNumberOfSubjects`: The expected number of subjects.
  - `expectedStudyDuration`: The expected study duration.
  - `expectedInformation`: The expected information.
  - `kMax`: The number of stages.
  - `milestone`: The milestone time relative to randomization.
  - `survDiffLower`: The lower equivalence limit of milestone survival probability difference.
  - `survDiffUpper`: The upper equivalence limit of milestone survival probability difference.
  - `surv1`: The milestone survival probability for the treatment group.
  - `surv2`: The milestone survival probability for the control group.
  - `survDiff`: The milestone survival probability difference.
  - `accrualDuration`: The accrual duration.
  - `followupTime`: The follow-up duration.
  - `fixedFollowup`: Whether a fixed follow-up design is used.
- `byStageResults`: A data frame containing the following variables:
  - `informationRates`: The information rates.
  - `efficacyBounds`: The efficacy boundaries on the Z-scale for each of the two one-sided tests.
  - `rejectPerStage`: The probability for efficacy stopping.
  - `cumulativeRejection`: The cumulative probability for efficacy stopping.

  - `cumulativeAlphaSpent`: The cumulative alpha for each of the two one-sided tests.
  - `cumulativeAttainedAlphaH10`: The cumulative alpha attained under `H10`.
  - `cumulativeAttainedAlphaH20`: The cumulative alpha attained under `H20`.
  - `numberOfEvents`: The number of events.
  - `numberOfDropouts`: The number of dropouts.
  - `numberOfSubjects`: The number of subjects.
  - `numberOfMilestone`: The number of subjects reaching milestone.
  - `analysisTime`: The average time since trial start.
  - `efficacySurvDiffLower`: The efficacy boundaries on the milestone survival probability difference scale for the one-sided null hypothesis at the lower equivalence limit.
  - `efficacySurvDiffUpper`: The efficacy boundaries on the milestone survival probability difference scale for the one-sided null hypothesis at the upper equivalence limit.
  - `efficacyP`: The efficacy bounds on the p-value scale for each of the two one-sided tests.
  - `information`: The cumulative information.
- `settings`: A list containing the following input parameters: `typeAlphaSpending`, `parameterAlphaSpending`, `userAlphaSpending`, `allocationRatioPlanned`, `accrualTime`, `accuralIntensity`, `piecewiseSurvivalTime`, `stratumFraction`, `lambda1`, `lambda2`, `gamma1`, `gamma2`, and `spendingTime`.
- `byTreatmentCounts`: A list containing the following counts by treatment group:
  - `numberOfEvents1`: The number of events by stage for the treatment group.
  - `numberOfDropouts1`: The number of dropouts by stage for the treatment group.
  - `numberOfSubjects1`: The number of subjects by stage for the treatment group.
  - `numberOfMilestone1`: The number of subjects reaching milestone by stage for the active 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.
  - `numberOfMilestone2`: The number of subjects reaching milestone by stage for the control group.
  - `expectedNumberOfEvents1`: The expected number of events for the treatment group.
  - `expectedNumberOfDropouts1`: The expected number of dropouts for the active treatment group.
  - `expectedNumberOfSubjects1`: The expected number of subjects for the active treatment group.
  - `expectedNumberOfMilestone1`: The expected number of subjects reaching milestone for the active 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.
  - `expectedNumberOfMilestone2`: The expected number of subjects reaching milestone for the control group.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

See Also

kmstat

Examples

```
kmpowerequiv(kMax = 2, informationRates = c(0.5, 1),
             alpha = 0.05, typeAlphaSpending = "sfOF",
             milestone = 18,
             survDiffLower = -0.13, survDiffUpper = 0.13,
             allocationRatioPlanned = 1, accrualTime = seq(0, 8),
             accrualIntensity = 26/9*seq(1, 9),
             piecewiseSurvivalTime = c(0, 6),
             stratumFraction = c(0.2, 0.8),
             lambda1 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
             lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
             gamma1 = -log(1-0.05)/12,
             gamma2 = -log(1-0.05)/12, accrualDuration = 22,
             followupTime = 18, fixedFollowup = FALSE)
```

`kmsamplesize` *Sample Size for Difference in Milestone Survival Probabilities*

Description

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

Usage

```
kmsamplesize(
  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_,
  milestone = NA_real_,
```

```
    survDiffH0 = 0,
    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,
    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. 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.
`milestone`	The milestone time at which to calculate the survival probability.
`survDiffH0`	The difference in milestone survival probabilities under the null hypothesis. Defaults to 0 for superiority test.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`interval`	The interval to search for the solution of accrualDuration, followupTime, or the proportionality constant of accrualIntensity. Defaults to `c(0.001, 240)`.

`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. Defaults to 1 for sample size rounding.

Value

A list of two components:

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

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

See Also

`kmpower`

Examples

```
# Example 1: Obtains follow-up time given power, accrual intensity,
# and accrual duration for variable follow-up. Of note, the power
# reaches the maximum when the follow-up time equals milestone.

kmsamplesize(beta = 0.25, kMax = 2, informationRates = c(0.8, 1),
             alpha = 0.025, typeAlphaSpending = "sfOF",
             milestone = 18,
             allocationRatioPlanned = 1, accrualTime = seq(0, 8),
             accrualIntensity = 26/9*seq(1, 9),
             piecewiseSurvivalTime = c(0, 6),
             stratumFraction = c(0.2, 0.8),
             lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
             lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
             gamma1 = -log(1-0.05)/12,
             gamma2 = -log(1-0.05)/12, accrualDuration = 22,
             followupTime = NA, fixedFollowup = FALSE)

# Example 2: Obtains accrual intensity given power, accrual duration, and
# follow-up time for variable follow-up

kmsamplesize(beta = 0.2, kMax = 2, informationRates = c(0.8, 1),
             alpha = 0.025, typeAlphaSpending = "sfOF",
             milestone = 18,
             allocationRatioPlanned = 1, accrualTime = seq(0, 8),
             accrualIntensity = 26/9*seq(1, 9),
             piecewiseSurvivalTime = c(0, 6),
             stratumFraction = c(0.2, 0.8),
             lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
             lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
             gamma1 = -log(1-0.05)/12,
             gamma2 = -log(1-0.05)/12, accrualDuration = 22,
```

```
             followupTime = 18, fixedFollowup = FALSE)


# Example 3: Obtains accrual duration given power, accrual intensity, and
# follow-up time for fixed follow-up

kmsamplesize(beta = 0.2, kMax = 2, informationRates = c(0.8, 1),
             alpha = 0.025, typeAlphaSpending = "sfOF",
             milestone = 18,
             allocationRatioPlanned = 1, accrualTime = seq(0, 8),
             accrualIntensity = 26/9*seq(1, 9),
             piecewiseSurvivalTime = c(0, 6),
             stratumFraction = c(0.2, 0.8),
             lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
             lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
             gamma1 = -log(1-0.05)/12,
             gamma2 = -log(1-0.05)/12, accrualDuration = NA,
             followupTime = 18, fixedFollowup = TRUE)
```

`kmsamplesize1s` *Sample Size for One-Sample Milestone Survival Probability*

Description

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

Usage

```
kmsamplesize1s(
  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_,
  milestone = NA_real_,
  survH0 = NA_real_,
```

```
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  lambda = NA_real_,
  gamma = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  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. 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.
`milestone`	The milestone time at which to calculate the survival probability.
`survH0`	The milestone survival probability under the null hypothesis.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda`	A vector of hazard rates for the event in each analysis time interval by stratum under the alternative hypothesis.
`gamma`	The hazard rate for exponential dropout or a vector of hazard rates for piecewise exponential dropout. Defaults to 0 for no dropout.
`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.
`interval`	The interval to search for the solution of accrualDuration, followupDuration, or the proportionality constant of accrualIntensity. Defaults to `c(0.001, 240)`.
`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. Defaults to 1 for sample size rounding.

Value

A list of two components:

- `resultsUnderH1`: An S3 class `kmpower1s` object under the alternative hypothesis.
- `resultsUnderH0`: An S3 class `kmpower1s` object under the null hypothesis.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

See Also

kmpower1s

Examples

```
# Example 1: Obtains follow-up duration given power, accrual intensity,
# and accrual duration for variable follow-up

kmsamplesize1s(beta = 0.2, kMax = 2,
               informationRates = c(0.8, 1),
               alpha = 0.025, typeAlphaSpending = "sfOF",
               milestone = 18, survH0 = 0.30,
               accrualTime = seq(0, 8),
               accrualIntensity = 26/9*seq(1, 9),
               piecewiseSurvivalTime = c(0, 6),
               stratumFraction = c(0.2, 0.8),
               lambda = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
               gamma = -log(1-0.05)/12, accrualDuration = 22,
               followupTime = NA, fixedFollowup = FALSE)

# Example 2: Obtains accrual intensity given power, accrual duration, and
# follow-up duration for variable follow-up

kmsamplesize1s(beta = 0.2, kMax = 2,
               informationRates = c(0.8, 1),
               alpha = 0.025, typeAlphaSpending = "sfOF",
               milestone = 18, survH0 = 0.30,
               accrualTime = seq(0, 8),
               accrualIntensity = 26/9*seq(1, 9),
               piecewiseSurvivalTime = c(0, 6),
               stratumFraction = c(0.2, 0.8),
               lambda = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
               gamma = -log(1-0.05)/12, accrualDuration = 22,
               followupTime = 18, fixedFollowup = FALSE)


# Example 3: Obtains accrual duration given power, accrual intensity, and
# follow-up duration for fixed follow-up

kmsamplesize1s(beta = 0.2, kMax = 2,
               informationRates = c(0.8, 1),
               alpha = 0.025, typeAlphaSpending = "sfOF",
               milestone = 18, survH0 = 0.30,
               accrualTime = seq(0, 8),
               accrualIntensity = 26/9*seq(1, 9),
               piecewiseSurvivalTime = c(0, 6),
               stratumFraction = c(0.2, 0.8),
               lambda = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
               gamma = -log(1-0.05)/12, accrualDuration = NA,
               followupTime = 18, fixedFollowup = TRUE)
```

`kmsamplesizeequiv` *Sample Size for Equivalence in Milestone Survival Probability Difference*

Description

Obtains the sample size for equivalence in milestone survival probability difference.

Usage

```
kmsamplesizeequiv(
  beta = 0.2,
  kMax = 1L,
  informationRates = NA_real_,
  criticalValues = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  milestone = NA_real_,
  survDiffLower = NA_real_,
  survDiffUpper = 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,
  interval = as.numeric(c(0.001, 240)),
  spendingTime = NA_real_,
  rounding = 1L
)
```

Arguments

`beta`	The type II error.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Defaults to `(1:kMax) / kMax` if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.

`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`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.
`milestone`	The milestone time at which to calculate the survival probability.
`survDiffLower`	The lower equivalence limit of milestone survival probability difference.
`survDiffUpper`	The upper equivalence limit of milestone survival probability difference.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.

`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`interval`	The interval to search for the solution of accrualDuration, followupDuration, or the proportionality constant of accrualIntensity. Defaults to `c(0.001, 240)`.
`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. Defaults to 1 for sample size rounding.

Value

An S3 class `kmpowerequiv` object

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

See Also

`kmpowerequiv`

Examples

```
kmsamplesizeequiv(beta = 0.1, kMax = 2, informationRates = c(0.5, 1),
                  alpha = 0.05, typeAlphaSpending = "sfOF",
                  milestone = 18,
                  survDiffLower = -0.13, survDiffUpper = 0.13,
                  allocationRatioPlanned = 1, accrualTime = seq(0, 8),
                  accrualIntensity = 26/9*seq(1, 9),
                  piecewiseSurvivalTime = c(0, 6),
                  stratumFraction = c(0.2, 0.8),
                  lambda1 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
                  lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
                  gamma1 = -log(1-0.05)/12,
                  gamma2 = -log(1-0.05)/12, accrualDuration = NA,
                  followupTime = 18, fixedFollowup = FALSE)
```

`kmstat` *Stratified Difference in Milestone Survival Probabilities*

Description

Obtains the stratified milestone survival probabilities and difference in milestone survival probabilities at given calendar times.

Usage

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

Arguments

`time`	A vector of calendar times for data cut.
`milestone`	The milestone time at which to calculate the survival probability.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.

`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.

Value

A data frame containing the following variables:

- `time`: The calendar 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.
- `milestone`: The milestone time relative to randomization.
- `nmilestone`: The total number of subjects reaching milestone.
- `nmilestone1`: The number of subjects reaching milestone in the active treatment group.
- `nmiletone2`: The number of subjects reaching milestone in the control group.
- `surv1`: The milestone survival probability for the treatment group.
- `surv2`: The milestone survival probability for the control group.
- `survDiff`: The difference in milestone survival probabilities, i.e., `surv1 - surv2`.
- `vsurv1`: The variance for `surv1`.
- `vsurv2`: The variance for `surv2`.
- `vsurvDiff`: The variance for `survDiff`.
- `information`: The information for `survDiff`, equal to `1/vsurvDiff`.
- `survDiffZ`: The Z-statistic value, i.e., `survDiff/sqrt(vsurvDiff)`.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

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

kmstat(time = c(22, 40),
       milestone = 18,
       allocationRatioPlanned = 1,
       accrualTime = seq(0, 8),
       accrualIntensity = 26/9*seq(1, 9),
```

```
        piecewiseSurvivalTime = c(0, 6),
        stratumFraction = c(0.2, 0.8),
        lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
        lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
        gamma1 = -log(1-0.05)/12,
        gamma2 = -log(1-0.05)/12,
        accrualDuration = 22,
        followupTime = 18, fixedFollowup = FALSE)
```

`liferegr` *Parametric Regression Models for Failure Time Data*

Description

Obtains the parameter estimates from parametric regression models with uncensored, right censored, left censored, or interval censored data.

Usage

```
liferegr(
  data,
  rep = "",
  stratum = "",
  time = "time",
  time2 = "",
  event = "event",
  covariates = "",
  weight = "",
  offset = "",
  id = "",
  dist = "weibull",
  init = NA_real_,
  robust = FALSE,
  plci = FALSE,
  alpha = 0.05,
  maxiter = 50,
  eps = 1e-09
)
```

Arguments

`data` The input data frame that contains the following variables:

- `rep`: The replication for by-group processing.
- `stratum`: The stratum.
- `time`: The follow-up time for right censored data, or the left end of each interval for interval censored data.

	• `time2`: The right end of each interval for interval censored data. • `event`: The event indicator, 1=event, 0=no event. • `covariates`: The values of baseline covariates. • `weight`: The weight for each observation. • `offset`: The offset for each observation. • `id`: The optional subject ID to group the score residuals in computing the robust sandwich variance.
`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 or the left end of each interval for interval censored data in the input data.
`time2`	The name of the right end of each interval for interval censored data in the input data.
`event`	The name of the event variable in the input data for right censored data.
`covariates`	The vector of names of baseline covariates in the input data.
`weight`	The name of the weight variable in the input data.
`offset`	The name of the offset variable in the input data.
`id`	The name of the id variable in the input data.
`dist`	The assumed distribution for time to event. Options include "exponential", "weibull", "lognormal", and "loglogistic" to be modeled on the log-scale, and "normal" and "logistic" to be modeled on the original scale.
`init`	A vector of initial values for the model parameters, including regression coefficients and the log scale parameter. By default, initial values are derived from an intercept-only model. If this approach fails, ordinary least squares (OLS) estimates, ignoring censoring, are used instead.
`robust`	Whether a robust sandwich variance estimate should be computed. In the presence of the id variable, the score residuals will be aggregated for each id when computing the robust sandwich variance estimate.
`plci`	Whether to obtain profile likelihood confidence interval.
`alpha`	The two-sided significance level.
`maxiter`	The maximum number of iterations.
`eps`	The tolerance to declare convergence.

Details

There are two ways to specify the model, one for right censored data through the time and event variables, and the other for interval censored data through the time (lower) and time2 (upper) variables. For the second form, we follow the convention used in SAS PROC LIFEREG:

- If lower is not missing, upper is not missing, and lower is equal to upper, then there is no censoring and the event occurred at time lower.
- If lower is not missing, upper is not missing, and lower < upper, then the event time is censored within the interval (lower, upper).

- If lower is missing, but upper is not missing, then upper will be used as the left censoring value.
- If lower is not missing, but upper is missing, then lower will be used as the right censoring value.
- If lower is not missing, upper is not missing, but lower > upper, or if both lower and upper are missing, then the observation will not be used.

Value

A list with the following components:

- `sumstat`: The data frame of summary statistics of model fit with the following variables:
  - `n`: The number of observations.
  - `nevents`: The number of events.
  - `loglik0`: The log-likelihood under null.
  - `loglik1`: The maximum log-likelihood.
  - `niter`: The number of Newton-Raphson iterations.
  - `dist`: The assumed distribution.
  - `p`: The number of parameters, including the intercept, regression coefficients associated with the covariates, and the log scale parameters for the strata.
  - `nvar`: The number of regression coefficients associated with the covariates (excluding the intercept).
  - `robust`: Whether the robust sandwich variance estimate is requested.
  - `fail`: Whether the model fails to converge.
  - `rep`: The replication.
- `parest`: The data frame of parameter estimates with the following variables:
  - `param`: The name of the covariate for the parameter estimate.
  - `beta`: The parameter estimate.
  - `sebeta`: The standard error of parameter estimate.
  - `z`: The Wald test statistic for the parameter.
  - `expbeta`: The exponentiated parameter estimate.
  - `vbeta`: The covariance matrix for parameter estimates.
  - `lower`: The lower limit of confidence interval.
  - `upper`: The upper limit of confidence interval.
  - `p`: The p-value from the chi-square test.
  - `method`: The method to compute the confidence interval and p-value.
  - `sebeta_naive`: The naive standard error of parameter estimate if robust variance is requested.
  - `vbeta_naive`: The naive covariance matrix for parameter estimates if robust variance is requested.
  - `rep`: The replication.
- `p`: The number of parameters.
- `nvar`: The number of columns of the design matrix excluding the intercept.

- `param`: The parameter names.
- `beta`: The parameter estimate.
- `vbeta`: The covariance matrix for parameter estimates.
- `vbeta_naive`: The naive covariance matrix for parameter estimates.
- `terms`: The terms object.
- `xlevels`: A record of the levels of the factors used in fitting.
- `data`: The input data.
- `rep`: The name(s) of the replication variable(s).
- `stratum`: The name(s) of the stratum variable(s).
- `time`: The name of the time variable.
- `time2`: The name of the time2 variable.
- `event`: The name of the event variable.
- `covariates`: The names of baseline covariates.
- `weight`: The name of the weight variable.
- `offset`: The name of the offset variable.
- `id`: The name of the id variable.
- `dist`: The assumed distribution for time to event.
- `robust`: Whether a robust sandwich variance estimate should be computed.
- `plci`: Whether to obtain profile likelihood confidence interval.
- `alpha`: The two-sided significance level.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

John D. Kalbfleisch and Ross L. Prentice. The Statistical Analysis of Failure Time Data. Wiley: New York, 1980.

Examples

```
library(dplyr)

# right censored data
(fit1 <- liferegr(
  data = rawdata %>% mutate(treat = 1*(treatmentGroup == 1)),
  rep = "iterationNumber", stratum = "stratum",
  time = "timeUnderObservation", event = "event",
  covariates = "treat", dist = "weibull"))

# tobit regression for left censored data
(fit2 <- liferegr(
  data = tobin %>% mutate(time = ifelse(durable>0, durable, NA)),
  time = "time", time2 = "durable",
  covariates = c("age", "quant"), dist = "normal"))
```

`logisregr` *Logistic Regression Models for Binary Data*

Description

Obtains the parameter estimates from logistic regression models with binary data.

Usage

```
logisregr(
  data,
  rep = "",
  event = "event",
  covariates = "",
  freq = "",
  weight = "",
  offset = "",
  id = "",
  link = "logit",
  init = NA_real_,
  robust = FALSE,
  firth = FALSE,
  flic = FALSE,
  plci = FALSE,
  alpha = 0.05,
  maxiter = 50,
  eps = 1e-09
)
```

Arguments

`data`	The input data frame that contains the following variables: • `rep`: The replication for by-group processing. • `event`: The event indicator, 1=event, 0=no event. • `covariates`: The values of baseline covariates. • `freq`: The frequency for each observation. • `weight`: The weight for each observation. • `offset`: The offset for each observation. • `id`: The optional subject ID to group the score residuals in computing the robust sandwich variance.
`rep`	The name(s) of the replication variable(s) in the input data.
`event`	The name of the event variable in the input data.
`covariates`	The vector of names of baseline covariates in the input data.

`freq`	The name of the frequency variable in the input data. The frequencies must be the same for all observations within each cluster as indicated by the id. Thus freq is the cluster frequency.
`weight`	The name of the weight variable in the input data.
`offset`	The name of the offset variable in the input data.
`id`	The name of the id variable in the input data.
`link`	The link function linking the response probabilities to the linear predictors. Options include "logit" (default), "probit", and "cloglog" (complementary log-log).
`init`	A vector of initial values for the model parameters. By default, initial values are derived from an intercept-only model.
`robust`	Whether a robust sandwich variance estimate should be computed. In the presence of the id variable, the score residuals will be aggregated for each id when computing the robust sandwich variance estimate.
`firth`	Whether the firth's bias reducing penalized likelihood should be used. The default is `FALSE`.
`flic`	Whether to apply intercept correction to obtain more accurate predicted probabilities. The default is `FALSE`.
`plci`	Whether to obtain profile likelihood confidence interval.
`alpha`	The two-sided significance level.
`maxiter`	The maximum number of iterations.
`eps`	The tolerance to declare convergence.

Details

Fitting a logistic regression model using Firth's bias reduction method is equivalent to penalization of the log-likelihood by the Jeffreys prior. Firth's penalized log-likelihood is given by

$$l(\beta) + \frac{1}{2}\log(\mathrm{det}(I(\beta)))$$

and the components of the gradient $g(\beta)$ are computed as

$$g(\beta_j) + \frac{1}{2}\mathrm{trace}\left(I(\beta)^{-1}\frac{\partial I(\beta)}{\partial \beta_j}\right)$$

The Hessian matrix is not modified by this penalty.

Firth's method reduces bias in maximum likelihood estimates of coefficients, but it introduces a bias toward one-half in the predicted probabilities.

A straightforward modification to Firth's logistic regression to achieve unbiased average predicted probabilities involves a post hoc adjustment of the intercept. This approach, known as Firth's logistic regression with intercept correction (FLIC), preserves the bias-corrected effect estimates. By excluding the intercept from penalization, it ensures that we don't sacrifice the accuracy of effect estimates to improve the predictions.

Value

A list with the following components:

- `sumstat`: The data frame of summary statistics of model fit with the following variables:
  - `n`: The number of subjects.
  - `nevents`: The number of events.
  - `loglik0`: The (penalized) log-likelihood under null.
  - `loglik1`: The maximum (penalized) log-likelihood.
  - `niter`: The number of Newton-Raphson iterations.
  - `p`: The number of parameters, including the intercept, and regression coefficients associated with the covariates.
  - `link`: The link function.
  - `robust`: Whether a robust sandwich variance estimate should be computed.
  - `firth`: Whether the firth's penalized likelihood is used.
  - `flic`: Whether to apply intercept correction.
  - `fail`: Whether the model fails to converge.
  - `loglik0_unpenalized`: The unpenalized log-likelihood under null.
  - `loglik1_unpenalized`: The maximum unpenalized log-likelihood.
  - `rep`: The replication.
- `parest`: The data frame of parameter estimates with the following variables:
  - `param`: The name of the covariate for the parameter estimate.
  - `beta`: The parameter estimate.
  - `sebeta`: The standard error of parameter estimate.
  - `z`: The Wald test statistic for the parameter.
  - `expbeta`: The exponentiated parameter estimate.
  - `vbeta`: The covariance matrix for parameter estimates.
  - `lower`: The lower limit of confidence interval.
  - `upper`: The upper limit of confidence interval.
  - `p`: The p-value from the chi-square test.
  - `method`: The method to compute the confidence interval and p-value.
  - `sebeta_naive`: The naive standard error of parameter estimate.
  - `vbeta_naive`: The naive covariance matrix of parameter estimates.
  - `rep`: The replication.
- `fitted`: The data frame with the following variables:
  - `linear_predictors`: The linear fit on the link function scale.
  - `fitted_values`: The fitted probabilities of having an event, obtained by transforming the linear predictors by the inverse of the link function.
  - `rep`: The replication.
- `p`: The number of parameters.
- `link`: The link function.
- `param`: The parameter names.

- `beta`: The parameter estimate.
- `vbeta`: The covariance matrix for parameter estimates.
- `vbeta_naive`: The naive covariance matrix for parameter estimates.
- `linear_predictors`: The linear fit on the link function scale.
- `fitted_values`: The fitted probabilities of having an event.
- `terms`: The terms object.
- `xlevels`: A record of the levels of the factors used in fitting.
- `data`: The input data.
- `rep`: The name(s) of the replication variable(s).
- `event`: The name of the event variable.
- `covariates`: The names of baseline covariates.
- `freq`: The name of the freq variable.
- `weight`: The name of the weight variable.
- `offset`: The name of the offset variable.
- `id`: The name of the id variable.
- `robust`: Whether a robust sandwich variance estimate should be computed.
- `firth`: Whether to use the firth's bias reducing penalized likelihood.
- `flic`: Whether to apply intercept correction.
- `plci`: Whether to obtain profile likelihood confidence interval.
- `alpha`: The two-sided significance level.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

David Firth. Bias Reduction of Maximum Likelihood Estimates. Biometrika 1993; 80:27–38.

Georg Heinze and Michael Schemper. A solution to the problem of separation in logistic regression. Statistics in Medicine 2002;21:2409–2419.

Rainer Puhr, Georg Heinze, Mariana Nold, Lara Lusa, and Angelika Geroldinger. Firth's logistic regression with rare events: accurate effect estimates and predictions? Statistics in Medicine 2017; 36:2302-2317.

Examples

```
(fit1 <- logisregr(
  ingots, event = "NotReady", covariates = "Heat*Soak", freq = "Freq"))
```

`lrpower` *Log-Rank Test Power*

Description

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

Usage

```
lrpower(
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  hazardRatioH0 = 1,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = 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,
  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.

`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`rho1`	The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
`rho2`	The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
`estimateHazardRatio`	Whether to estimate the hazard ratio from weighted Cox regression model and report the stopping boundaries on the hazard ratio scale.
`typeOfComputation`	The type of computation, either "direct" for the direct approximation method, or "schoenfeld" for the Schoenfeld method. Defaults to "direct". Can use "Schoenfeld" under proportional hazards and conventional log-rank test.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.
`studyDuration`	Study duration for fixed follow-up design. Defaults to missing, which is to be replaced with the sum of `accrualDuration` and `followupTime`. If provided, the value is allowed to be less than the sum of `accrualDuration` and `followupTime`.

Value

An S3 class `lrpower` object with 4 components:

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

  - `alpha`: The overall significance level.
  - `numberOfEvents`: The total number of events.
  - `numberOfDropouts`: The total number of dropouts.
  - `numbeOfSubjects`: The total number of subjects.
  - `studyDuration`: The total study duration.
  - `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`, `accuralIntensity`, `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.

lrpower(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)
```

`lrpowerequiv` *Power for Equivalence in Hazard Ratio*

Description

Obtains the power for equivalence in hazard ratio.

Usage

```
lrpowerequiv(
  kMax = 1L,
  informationRates = NA_real_,
  criticalValues = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  hazardRatioLower = NA_real_,
  hazardRatioUpper = 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,
  typeOfComputation = "direct",
  spendingTime = NA_real_,
  studyDuration = NA_real_
)
```

Arguments

`kMax`	The maximum number of stages.
`informationRates`	The information rates. Defaults to `(1:kMax) / kMax` if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`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.

`hazardRatioLower`	The lower equivalence limit of hazard ratio.
`hazardRatioUpper`	The upper equivalence limit of hazard ratio.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`typeOfComputation`	The type of computation, either "direct" for the direct approximation method, or "schoenfeld" for the Schoenfeld method. Defaults to "direct". Can use "Schoenfeld" under proportional hazards and conventional log-rank test.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.
`studyDuration`	Study duration for fixed follow-up design. Defaults to missing, which is to be replaced with the sum of `accrualDuration` and `followupTime`. If provided, the value is allowed to be less than the sum of `accrualDuration` and `followupTime`.

Value

An S3 class `lrpowerequiv` 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.
  - `kMax`: The number of stages.
  - `hazardRatioLower`: The lower equivalence limit of hazard ratio.
  - `hazardRatioUpper`: The upper equivalence limit of hazard ratio.
  - `accrualDuration`: The accrual duration.
  - `followupTime`: The follow-up time.
  - `fixedFollowup`: Whether a fixed follow-up design is used.
- `byStageResults`: A data frame containing the following variables:
  - `informationRates`: The information rates.
  - `efficacyBounds`: The efficacy boundaries on the Z-scale for each of the two one-sided tests.
  - `rejectPerStage`: The probability for efficacy stopping.
  - `cumulativeRejection`: The cumulative probability for efficacy stopping.
  - `cumulativeAlphaSpent`: The cumulative alpha for each of the two one-sided tests.
  - `cumulativeAttainedAlphaH10`: The cumulative alpha attained under `H10`.
  - `cumulativeAttainedAlphaH20`: The cumulative alpha attained under `H20`.
  - `numberOfEvents`: The number of events.
  - `numberOfDropouts`: The number of dropouts.
  - `numberOfSubjects`: The number of subjects.
  - `analysisTime`: The average time since trial start.
  - `efficacyHRLower`: The efficacy boundaries on the hazard ratio scale for the one-sided null hypothesis at the lower equivalence limit.
  - `efficacyHRUpper`: The efficacy boundaries on the hazard ratio scale for the one-sided null hypothesis at the upper equivalence limit.
  - `efficacyP`: The efficacy bounds on the p-value scale for each of the two one-sided tests.
  - `information`: The cumulative information.
  - `HR`: The average hazard ratio.

- `settings`: A list containing the following input parameters: `typeAlphaSpending`, `parameterAlphaSpending`, `userAlphaSpending`, `allocationRatioPlanned`, `accrualTime`, `accuralIntensity`, `piecewiseSurvivalTime`, `stratumFraction`, `lambda1`, `lambda2`, `gamma1`, `gamma2`, `typeOfComputation`, 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>`

See Also

`rmstat`

Examples

```
lrpowerequiv(kMax = 2, informationRates = c(0.5, 1),
             alpha = 0.05, typeAlphaSpending = "sfOF",
             hazardRatioLower = 0.71, hazardRatioUpper = 1.4,
             allocationRatioPlanned = 1, accrualTime = seq(0, 8),
             accrualIntensity = 100/9*seq(1, 9),
             piecewiseSurvivalTime = c(0, 6),
             lambda1 = c(0.0533, 0.0533),
             lambda2 = c(0.0533, 0.0533),
             gamma1 = -log(1-0.05)/12,
             gamma2 = -log(1-0.05)/12, accrualDuration = 22,
             followupTime = 18, fixedFollowup = FALSE)
```

`lrsamplesize` *Log-Rank Test Sample Size*

Description

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

Usage

```
lrsamplesize(
  beta = 0.2,
  kMax = 1L,
  informationRates = NA_real_,
  efficacyStopping = NA_integer_,
  futilityStopping = NA_integer_,
  criticalValues = NA_real_,
  alpha = 0.025,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  futilityBounds = NA_real_,
  typeBetaSpending = "none",
  parameterBetaSpending = NA_real_,
  userBetaSpending = NA_real_,
  hazardRatioH0 = 1,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = 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,
  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, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`rho1`	The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
`rho2`	The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
`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

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 = NA,
             followupTime = 18, fixedFollowup = FALSE)


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

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 = 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)
```

`lrsamplesizeequiv` *Sample Size for Equivalence in Hazard Ratio*

Description

Obtains the sample size for equivalence in hazard ratio.

Usage

```
lrsamplesizeequiv(
  beta = 0.2,
  kMax = 1L,
  informationRates = NA_real_,
  criticalValues = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  hazardRatioLower = NA_real_,
  hazardRatioUpper = 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,
  typeOfComputation = "direct",
  interval = as.numeric(c(0.001, 240)),
  spendingTime = NA_real_,
  rounding = 1L
)
```

Arguments

`beta`	The type II error.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Defaults to `(1:kMax) / kMax` if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`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.
`hazardRatioLower`	The lower equivalence limit of hazard ratio.
`hazardRatioUpper`	The upper equivalence limit of hazard ratio.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.

`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`typeOfComputation`	The type of computation, either "direct" for the direct approximation method, or "schoenfeld" for the Schoenfeld method. Defaults to "direct". Can use "Schoenfeld" under proportional hazards and conventional log-rank test.
`interval`	The interval to search for the solution of accrualDuration, followupDuration, or the proportionality constant of accrualIntensity. Defaults to `c(0.001, 240)`.
`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. Defaults to 1 for sample size rounding.

Value

An S3 class `lrpowerequiv` object

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

See Also

`lrpowerequiv`

Examples

```
lrsamplesizeequiv(kMax = 2, informationRates = c(0.5, 1),
                  alpha = 0.05, typeAlphaSpending = "sfOF",
                  hazardRatioLower = 0.71, hazardRatioUpper = 1.4,
                  allocationRatioPlanned = 1, accrualTime = seq(0, 8),
                  accrualIntensity = 26/9*seq(1, 9),
                  piecewiseSurvivalTime = c(0, 6),
```

```
                   lambda1 = c(0.0533, 0.0533),
                   lambda2 = c(0.0533, 0.0533),
                   gamma1 = -log(1-0.05)/12,
                   gamma2 = -log(1-0.05)/12, accrualDuration = NA,
                   followupTime = 18, fixedFollowup = FALSE)
```

`lrschoenfeld` *Schoenfeld Method for Log-Rank Test Sample Size Calculation*

Description

Obtains the sample size and study duration by calibrating the number of events calculated using the Schoenfeld formula under the proportional hazards assumption.

Usage

```
lrschoenfeld(
  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,
  hazardRatio = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  interval = as.numeric(c(0.001, 240)),
  spendingTime = NA_real_,
  rounding = 1L,
```

```
  calibrate = 1L,
  maxNumberOfIterations = 10000L,
  maxNumberOfRawDatasetsPerStage = 0L,
  seed = NA_integer_
)
```

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, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`hazardRatio`	Hazard ratio under the alternative hypothesis for the active treatment versus control.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`interval`	The interval to search for the solution of followupTime. Defaults to `c(0.001, 240)`.
`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.
`calibrate`	Whether to use simulations to calibrate the number of events calculated using the Schoenfeld formula.
`maxNumberOfIterations`	The number of simulation iterations. Defaults to 10000.
`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.

Details

This function calculates the sample size and study duration by calibrating the number of events estimated using the Schoenfeld formula under the proportional hazards assumption, particularly when the hazard ratio is far away from one and/or the allocation between groups is unequal.

For a fixed design, the Schoenfeld formula for the required number of events is

$$D = \frac{(\Phi^{-1}(1-\alpha) + \Phi^{-1}(1-\beta))^2}{(\theta - \theta_0)^2 r(1-r)}$$

where D is the total number of events required, α is the type I error rate, β is the type II error rate, r is the randomization probability for the active treatment group, θ_0 and θ are the log hazard ratios under the null and alternative hypotheses, respectively.

The function first computes the number of events using the Schoenfeld formula. If `calibrate` is set to 1, the function uses simulations to calibrate the number of events, accounting for scenarios where the Schoenfeld formula may be inaccurate (e.g., when allocation is unequal or the hazard ratio is extreme).

Let $D_{schoenfeld}$ be the number of events calculated by the Schoenfeld formula, and $D_{calibrated}$ be the calibrated number of events. The calibrated number of events is calculated as #'

$$D_{\text{calibrated}} = \frac{\left\{\Phi^{-1}(1-\alpha) + \Phi^{-1}(1-\beta)\right\}^2}{\left\{\Phi^{-1}(1-\alpha) + \Phi^{-1}(1-\beta_{\text{schoenfeld}})\right\}^2} D_{\text{schoenfeld}}$$

where $\beta_{schoenfeld}$ is the empirical type II error estimated via simulation.

A second round of simulation is performed to obtain the empirical power using the calibrated number of events.

Value

A list of two components:

- `analyticalResults`: An S3 class `lrpower` object for the asymptotic power.
- `simulationResults`: An S3 class `lrsim` object for the empirical power.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
(lr1 <- lrschoenfeld(
  beta = 0.1, kMax = 2, alpha = 0.025,
  hazardRatioH0 = 1, allocationRatioPlanned = 1,
  accrualIntensity = 20, hazardRatio = 0.3,
  lambda2 = 1.9/12,
  gamma1 = -log(1-0.1)/24, gamma2 = -log(1-0.1)/24,
  fixedFollowup = 0, rounding = 1,
  calibrate = 0, maxNumberOfIterations = 1000,
  seed = 12345))
```

```
(lr2 <- lrschoenfeld(
  beta = 0.1, kMax = 2, alpha = 0.025,
  hazardRatioH0 = 1, allocationRatioPlanned = 1,
  accrualIntensity = 20, hazardRatio = 0.3,
  lambda2 = 1.9/12,
  gamma1 = -log(1-0.1)/24, gamma2 = -log(1-0.1)/24,
  fixedFollowup = 0, rounding = 1,
  calibrate = 1, maxNumberOfIterations = 1000,
  seed = 12345))
```

`lrsim` *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

`kMax`	The maximum number of stages.
`informationRates`	The information rates in terms of number of events for the conventional log-rank test and in terms of the actual information for weighted log-rank tests. Fixed prior to the trial. If left unspecified, it defaults to `plannedEvents / plannedEvents[kMax]` when `plannedEvents` is provided and to `plannedTime / plannedTime[kMax]` otherwise.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`futilityBounds`	Lower boundaries on the z-test statistic scale for stopping for futility at stages 1, ..., `kMax-1`. Defaults to `rep(-6, kMax-1)` if left unspecified. The futility bounds are non-binding for the calculation of critical values.
`hazardRatioH0`	Hazard ratio under the null hypothesis for the active treatment versus control. Defaults to 1 for superiority test.
`allocation1`	Number of subjects in the active treatment group in a randomization block. Defaults to 1 for equal randomization.
`allocation2`	Number of subjects in the control group in a randomization block. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.

`rho1`	The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
`rho2`	The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
`plannedEvents`	The planned cumulative total number of events at each stage.
`plannedTime`	The calendar times for the analyses. To use calendar time to plan the analyses, `plannedEvents` should be missing.
`maxNumberOfIterations`	The number of simulation iterations. Defaults to 1000.
`maxNumberOfRawDatasetsPerStage`	The number of raw datasets per stage to extract.
`seed`	The seed to reproduce the simulation results. The seed from the environment will be used if left unspecified.

Value

An S3 class `lrsim` object with 3 components:

- `overview`: A list containing the following information:
  - `rejectPerStage`: The efficacy stopping probability by stage.
  - `futilityPerStage`: The futility stopping probability by stage.
  - `cumulativeRejection`: Cumulative efficacy stopping probability by stage.
  - `cumulativeFutility`: The cumulative futility stopping probability by stage.
  - `numberOfEvents`: The average number of events by stage.
  - `numberOfDropouts`: The average number of dropouts by stage.
  - `numberOfSubjects`: The average number of subjects by stage.
  - `analysisTime`: The average analysis time by stage.
  - `overallReject`: The overall rejection probability.
  - `expectedNumberOfEvents`: The expected number of events for the overall study.
  - `expectedNumberOfDropouts`: The expected number of dropouts for the overall study.
  - `expectedNumberOfSubjects`: The expected number of subjects for the overall study.
  - `expectedStudyDuration`: The expected study duration.
  - `hazardRatioH0`: Hazard ratio under the null hypothesis for the active treatment versus control.
  - `useEvents`: whether the analyses are planned based on the number of events or calendar time.
  - `accrualDuration`: Duration of the enrollment period.
  - `fixedFollowup`: Whether a fixed follow-up design is used.
  - `rho1`: The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
  - `rho2`: The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
  - `kMax`: The maximum number of stages.
- `sumdata`: A data frame of summary data by iteration and stage:

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

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

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

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

sim1 = lrsim(kMax = 2, informationRates = c(0.5, 1),
             criticalValues = c(2.797, 1.977),
             accrualIntensity = 11,
             lambda1 = 0.018, lambda2 = 0.030,
             accrualDuration = 12,
             plannedEvents = c(60, 120),
             maxNumberOfIterations = 1000,
             maxNumberOfRawDatasetsPerStage = 1,
             seed = 314159)

# summary statistics
sim1

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

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


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

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

# summary statistics
sim2

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

`lrsim2e` *Log-Rank Test Simulation for Two Endpoints*

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 = 1L,
  kMaxe1 = 1L,
  hazardRatioH0e1 = 1,
  hazardRatioH0e2 = 1,
  allocation1 = 1L,
  allocation2 = 1L,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  rho = 0,
  lambda1e1 = NA_real_,
  lambda2e1 = NA_real_,
  lambda1e2 = NA_real_,
  lambda2e2 = NA_real_,
  gamma1e1 = 0L,
  gamma2e1 = 0L,
  gamma1e2 = 0L,
  gamma2e2 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  rho1 = 0,
  rho2 = 0,
  plannedEvents = NA_integer_,
  plannedTime = NA_real_,
  maxNumberOfIterations = 1000L,
  maxNumberOfRawDatasetsPerStage = 0L,
  seed = NA_integer_
)
```

Arguments

`kMax`	The maximum number of stages.
`kMaxe1`	Number of stages with timing determined by PFS events. Ranges from 0 (none) to `kMax`.
`hazardRatioH0e1`	Hazard ratio under the null hypothesis for the active treatment vs control for endpoint 1 (PFS). Defaults to 1 for superiority test.
`hazardRatioH0e2`	Hazard ratio under the null hypothesis for the active treatment vs control for endpoint 2 (OS). Defaults to 1 for superiority test.
`allocation1`	Number of subjects in the treatment group in a randomization block. Defaults to 1 for equal randomization.
`allocation2`	Number of subjects in the control group in a randomization block. Defaults to 1 for equal randomization.

`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`rho`	The correlation coefficient for the standard bivariate normal random variables used to generate time to disease progression and time to death using the inverse CDF method.
`lambda1e1`	A vector of hazard rates for the event in each analysis time interval by stratum for the treatment group and endpoint 1 (PFS).
`lambda2e1`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group and endpoint 1 (PFS).
`lambda1e2`	A vector of hazard rates for the event in each analysis time interval by stratum for the treatment group and endpoint 2 (OS).
`lambda2e2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group and endpoint 2 (OS).
`gamma1e1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the treatment group and endpoint 1 (PFS).
`gamma2e1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group and endpoint 1 (PFS).
`gamma1e2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the treatment group and endpoint 2 (OS).
`gamma2e2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group and endpoint 2 (OS).
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`rho1`	The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

`rho2`	The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
`plannedEvents`	The planned cumulative total number of PFS events at Look 1 to Look `kMaxe1` and the planned cumulative total number of OS events at Look `kMaxe1+1` to Look `kMax`.
`plannedTime`	The calendar times for the analyses. To use calendar time to plan the analyses, `plannedEvents` should be missing.
`maxNumberOfIterations`	The number of simulation iterations. Defaults to 1000.
`maxNumberOfRawDatasetsPerStage`	The number of raw datasets per stage to extract.
`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 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)
```

`lrsim2e3a` *Log-Rank Test Simulation for Two Endpoints and Three Arms*

Description

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

Usage

```
lrsim2e3a(
  kMax = 1L,
  kMaxe1 = 1L,
  hazardRatioH013e1 = 1,
  hazardRatioH023e1 = 1,
  hazardRatioH012e1 = 1,
  hazardRatioH013e2 = 1,
  hazardRatioH023e2 = 1,
  hazardRatioH012e2 = 1,
  allocation1 = 1L,
  allocation2 = 1L,
  allocation3 = 1L,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  rho = 0,
  lambda1e1 = NA_real_,
  lambda2e1 = NA_real_,
  lambda3e1 = NA_real_,
  lambda1e2 = NA_real_,
  lambda2e2 = NA_real_,
  lambda3e2 = NA_real_,
  gamma1e1 = 0L,
  gamma2e1 = 0L,
  gamma3e1 = 0L,
  gamma1e2 = 0L,
  gamma2e2 = 0L,
  gamma3e2 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  rho1 = 0,
  rho2 = 0,
  plannedEvents = NA_integer_,
  plannedTime = NA_real_,
  maxNumberOfIterations = 1000L,
  maxNumberOfRawDatasetsPerStage = 0L,
  seed = NA_integer_
)
```

Arguments

`kMax`	The maximum number of stages.
`kMaxe1`	Number of stages with timing determined by PFS events. Ranges from 0 (none) to `kMax`.
`hazardRatioH013e1`	Hazard ratio under the null hypothesis for arm 1 vs arm 3 for endpoint 1 (PFS). Defaults to 1 for superiority test.
`hazardRatioH023e1`	Hazard ratio under the null hypothesis for arm 2 vs arm 3 for endpoint 1 (PFS). Defaults to 1 for superiority test.
`hazardRatioH012e1`	Hazard ratio under the null hypothesis for arm 1 vs arm 2 for endpoint 1 (PFS). Defaults to 1 for superiority test.
`hazardRatioH013e2`	Hazard ratio under the null hypothesis for arm 1 vs arm 3 for endpoint 2 (OS). Defaults to 1 for superiority test.
`hazardRatioH023e2`	Hazard ratio under the null hypothesis for arm 2 vs arm 3 for endpoint 2 (OS). Defaults to 1 for superiority test.
`hazardRatioH012e2`	Hazard ratio under the null hypothesis for arm 1 vs arm 2 for endpoint 2 (OS). Defaults to 1 for superiority test.
`allocation1`	Number of subjects in Arm A in a randomization block. Defaults to 1 for equal randomization.
`allocation2`	Number of subjects in Arm B in a randomization block. Defaults to 1 for equal randomization.
`allocation3`	Number of subjects in Arm C in a randomization block. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`rho`	The correlation coefficient for the standard bivariate normal random variables used to generate time to disease progression and time to death using the inverse CDF method.
`lambda1e1`	A vector of hazard rates for the event in each analysis time interval by stratum for arm 1 and endpoint 1 (PFS).

`lambda2e1`	A vector of hazard rates for the event in each analysis time interval by stratum for arm 2 and endpoint 1 (PFS).
`lambda3e1`	A vector of hazard rates for the event in each analysis time interval by stratum for arm 3 and endpoint 1 (PFS).
`lambda1e2`	A vector of hazard rates for the event in each analysis time interval by stratum for arm 1 and endpoint 2 (OS).
`lambda2e2`	A vector of hazard rates for the event in each analysis time interval by stratum for arm 2 and endpoint 2 (OS).
`lambda3e2`	A vector of hazard rates for the event in each analysis time interval by stratum for arm 3 and endpoint 2 (OS).
`gamma1e1`	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 1 and endpoint 1 (PFS).
`gamma2e1`	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 2 and endpoint 1 (PFS).
`gamma3e1`	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 3 and endpoint 1 (PFS).
`gamma1e2`	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 1 and endpoint 2 (OS).
`gamma2e2`	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 2 and endpoint 2 (OS).
`gamma3e2`	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 3 and endpoint 2 (OS).
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`rho1`	The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
`rho2`	The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
`plannedEvents`	The planned cumulative total number of PFS events at Look 1 to Look `kMaxe1` for Arms A and C combined and the planned cumulative total number of OS events at Look `kMaxe1+1` to Look `kMax` for Arms A and C combined.
`plannedTime`	The calendar times for the analyses. To use calendar time to plan the analyses, `plannedEvents` should be missing.

`maxNumberOfIterations`	The number of simulation iterations. Defaults to 1000.
`maxNumberOfRawDatasetsPerStage`	The number of raw datasets per stage to extract.
`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.
  - `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 = 1L,
  hazardRatioH013 = 1,
  hazardRatioH023 = 1,
  hazardRatioH012 = 1,
  allocation1 = 1L,
  allocation2 = 1L,
  allocation3 = 1L,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  lambda3 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  gamma3 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  rho1 = 0,
  rho2 = 0,
  plannedEvents = NA_integer_,
  plannedTime = NA_real_,
  maxNumberOfIterations = 1000L,
  maxNumberOfRawDatasetsPerStage = 0L,
  seed = NA_integer_
)
```

Arguments

`kMax`	The maximum number of stages.
`hazardRatioH013`	Hazard ratio under the null hypothesis for arm 1 versus arm 3. Defaults to 1 for superiority test.

`hazardRatioH023`	Hazard ratio under the null hypothesis for arm 2 versus arm 3. Defaults to 1 for superiority test.
`hazardRatioH012`	Hazard ratio under the null hypothesis for arm 1 versus arm 2. Defaults to 1 for superiority test.
`allocation1`	Number of subjects in Arm A in a randomization block. Defaults to 1 for equal randomization.
`allocation2`	Number of subjects in Arm B in a randomization block. Defaults to 1 for equal randomization.
`allocation3`	Number of subjects in Arm C in a randomization block. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for arm 1.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for arm 2.
`lambda3`	A vector of hazard rates for the event in each analysis time interval by stratum for arm 3.
`gamma1`	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 1.
`gamma2`	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 2.
`gamma3`	The hazard rate for exponential dropout. A vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for arm 3.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`rho1`	The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

`rho2`	The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
`plannedEvents`	The planned cumulative total number of events at Look 1 to Look `kMax` for Arms A and C combined.
`plannedTime`	The calendar times for the analyses. To use calendar time to plan the analyses, `plannedEvents` should be missing.
`maxNumberOfIterations`	The number of simulation iterations. Defaults to 1000.
`maxNumberOfRawDatasetsPerStage`	The number of raw datasets per stage to extract.
`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, <`kaifenglu@gmail.com`>

Examples

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

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

`lrsimsub` *Log-Rank Test Simulation for Enrichment Design*

Description

Performs simulation for two-arm group sequential trials based on weighted log-rank test for a biomarker enrichment design. The looks are either driven by the total number of events in the ITT population or the biomarker positive sub population. Alternatively, the analyses can be planned to occur at specified calendar times.

Usage

```
lrsimsub(
  kMax = 1L,
  kMaxitt = 1L,
  hazardRatioH0itt = 1,
  hazardRatioH0pos = 1,
  hazardRatioH0neg = 1,
  allocation1 = 1L,
  allocation2 = 1L,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  p_pos = NA_real_,
  lambda1itt = NA_real_,
  lambda2itt = NA_real_,
  lambda1pos = NA_real_,
  lambda2pos = NA_real_,
  gamma1itt = 0L,
  gamma2itt = 0L,
  gamma1pos = 0L,
  gamma2pos = 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.
`kMaxitt`	Number of stages with timing determined by events in the ITT population. Ranges from 0 (none) to `kMax`.
`hazardRatioH0itt`	Hazard ratio under the null hypothesis for the ITT population. Defaults to 1 for

	superiority test.
`hazardRatioH0pos`	Hazard ratio under the null hypothesis for the biomarker positive sub population. Defaults to 1 for superiority test.
`hazardRatioH0neg`	Hazard ratio under the null hypothesis for the biomarker negative sub population. Defaults to 1 for superiority test.
`allocation1`	Number of subjects in the treatment group in a randomization block. Defaults to 1 for equal randomization.
`allocation2`	Number of subjects in the control group in a randomization block. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`p_pos`	The prevalence of the biomarker positive sub population in each stratum.
`lambda1itt`	A vector of hazard rates for the event in each analysis time interval by stratum for the treatment group in the ITT population.
`lambda2itt`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group in the ITT population.
`lambda1pos`	A vector of hazard rates for the event in each analysis time interval by stratum for the treatment group in the biomarker positive sub population.
`lambda2pos`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group in the biomarker positive sub population.
`gamma1itt`	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 in the ITT population.
`gamma2itt`	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 in the ITT population.
`gamma1pos`	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 in the biomarker positive sub population.

`gamma2pos`	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 in the biomarker positive sub population.
`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 events in the ITT population at Look 1 to Look `kMaxitt` and the planned cumulative total number of events at Look `kMaxitt+1` to Look `kMax` in the biomarker positive sub population.
`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.
  - `population`: The population ("ITT", "Biomarker Positive", "Biomarker Negative") under consideration.
  - `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.
  - `logRankStatistic`: The log-rank test Z-statistic for the population.
- `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.
  - `biomarker`: The biomarker status for the subject (1 for positive, 0 for negative).
  - `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 for the subject.
  - `event`: Whether the subject experienced an event.
  - `dropoutEvent`: Whether the subject dropped out.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
sim1 = lrsimsub(
  kMax = 2,
  kMaxitt = 2,
  allocation1 = 1,
  allocation2 = 1,
  accrualTime = seq(0,9),
  accrualIntensity = c(seq(10,70,10),rep(70,3)),
  piecewiseSurvivalTime = c(0,12,24),
  p_pos = 0.6,
  lambda1itt = c(0.00256, 0.00383, 0.00700),
  lambda2itt = c(0.00427, 0.00638, 0.01167),
  lambda1pos = c(0.00299, 0.00430, 0.01064),
  lambda2pos = c(0.00516, 0.00741, 0.01835),
  gamma1itt = -log(1-0.04)/12,
  gamma2itt = -log(1-0.04)/12,
  gamma1pos = -log(1-0.04)/12,
  gamma2pos = -log(1-0.04)/12,
  accrualDuration = 10.14,
  plannedEvents = c(108,144),
  maxNumberOfIterations = 1000,
  maxNumberOfRawDatasetsPerStage = 1,
  seed = 314159)
```

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

`lrstat` *Number of Subjects Having an Event and Log-Rank Statistics*

Description

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

Usage

```
lrstat(
  time = NA_real_,
  hazardRatioH0 = 1,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  rho1 = 0,
  rho2 = 0,
  predictTarget = 2L
)
```

Arguments

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

`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`rho1`	The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
`rho2`	The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.
`predictTarget`	The target of prediction. Set `predictTarget = 1` to predict the number of events only. Set `predictTarget = 2` (default) to predict the number of events and log-rank score statistic mean and variance. Set `predictTarget = 3` to predict the number of events, log-rank score statistic mean and variance, and hazard ratio and variance of log hazard ratio.

Value

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

- `time`: The analysis time since trial start.
- `subjects`: The number of enrolled subjects.
- `nevents`: The total number of events.
- `nevents1`: The number of events in the active treatment group.
- `nevents2`: The number of events in the control group.

- `ndropouts`: The total number of dropouts.
- `ndropouts1`: The number of 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)
```

`lrtest` *Log-Rank Test of Survival Curve Difference*

Description

Obtains the log-rank test using the Fleming-Harrington family of weights.

Usage

```
lrtest(
  data,
  rep = "",
  stratum = "",
  treat = "treat",
  time = "time",
  event = "event",
  rho1 = 0,
  rho2 = 0
)
```

Arguments

`data`	The input data frame that contains the following variables: • `rep`: The replication for by-group processing. • `stratum`: The stratum. • `treat`: The treatment. • `time`: The possibly right-censored survival time. • `event`: The event indicator.
`rep`	The name of the replication variable in the input data.
`stratum`	The name of the stratum variable in the input data.
`treat`	The name of the treatment variable 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.
`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.

Value

A data frame with the following variables:

- `uscore`: The numerator of the log-rank test statistic.
- `vscore`: The variance of the log-rank score test statistic.

- `logRankZ`: The Z-statistic value.
- `logRankPValue`: The one-sided p-value.
- `rho1`: The first parameter of the Fleming-Harrington weights.
- `rho2`: The second parameter of the Fleming-Harrington weights.
- `rep`: The replication.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
df <- lrtest(data = rawdata, rep = "iterationNumber",
             stratum = "stratum", treat = "treatmentGroup",
             time = "timeUnderObservation", event = "event",
             rho1 = 0.5, rho2 = 0)
head(df)
```

`mnOddsRatioCI` *Miettinen-Nurminen Score Confidence Interval for Two-Sample Odds Ratio*

Description

Obtains the Miettinen-Nurminen score confidence interval for two-sample odds ratio possibly with stratification.

Usage

```
mnOddsRatioCI(
  n1 = NA_real_,
  y1 = NA_real_,
  n2 = NA_real_,
  y2 = NA_real_,
  cilevel = 0.95
)
```

Arguments

`n1`	The sample size for the active treatment group.
`y1`	The number of responses for the active treatment group.
`n2`	The sample size for the control group.
`y2`	The number of responses for the control group.
`cilevel`	The confidence interval level.

Details

The Mantel-Haenszel sample size weights are used for stratified samples.

Value

A list with two components:

- `data` A data frame containing the input sample size and number of responses for each treatment group. It has the following variables:
  - `n1`: The sample size for the active treatment group.
  - `y1`: The number of responses for the active treatment group.
  - `n2`: The sample size for the control group.
  - `y2`: The number of responses for the control group.
- `estimates`: A data frame containing the point estimate and confidence interval for odds ratio. It has the following variables:
  - `scale`: The scale of treatment effect.
  - `estimate`: The point estimate.
  - `lower`: The lower limit of the confidence interval.
  - `upper`: The upper limit of the confidence interval.
  - `cilevel`: The confidence interval level.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
mnOddsRatioCI(n1 = c(10,10), y1 = c(4,3), n2 = c(20,10), y2 = c(2,0))
```

`mnRateDiffCI` *Miettinen-Nurminen Score Confidence Interval for Two-Sample Rate Difference*

Description

Obtains the Miettinen-Nurminen score confidence interval for two-sample rate difference possibly with stratification.

Usage

```
mnRateDiffCI(
  t1 = NA_real_,
  y1 = NA_real_,
  t2 = NA_real_,
  y2 = NA_real_,
  cilevel = 0.95
)
```

Arguments

`t1`	The exposure for the active treatment group.
`y1`	The number of events for the active treatment group.
`t2`	The exposure for the control group.
`y2`	The number of events for the control group.
`cilevel`	The confidence interval level.

Details

The Mantel-Haenszel weights are used for stratified samples.

Value

A list with two components:

- `data` A data frame containing the input exposure and number of events for each treatment group. It has the following variables:
  - `t1`: The exposure for the active treatment group.
  - `y1`: The number of events for the active treatment group.
  - `t2`: The exposure for the control group.
  - `y2`: The number of events for the control group.
- `estimates`: A data frame containing the point estimate and confidence interval for rate difference. It has the following variables:
  - `scale`: The scale of treatment effect.
  - `estimate`: The point estimate.
  - `lower`: The lower limit of the confidence interval.
  - `upper`: The upper limit of the confidence interval.
  - `cilevel`: The confidence interval level.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
mnRateDiffCI(t1 = c(10,10), y1 = c(4,3), t2 = c(20,10), y2 = c(2,0))
```

`mnRateRatioCI` *Miettinen-Nurminen Score Confidence Interval for Two-Sample Rate Ratio*

Description

Obtains the Miettinen-Nurminen score confidence interval for two-sample rate ratio possibly with stratification.

Usage

```
mnRateRatioCI(t1, y1, t2, y2, cilevel = 0.95)
```

Arguments

`t1`	The exposure for the active treatment group.
`y1`	The number of events for the active treatment group.
`t2`	The exposure for the control group.
`y2`	The number of events for the control group.
`cilevel`	The confidence interval level.

Details

The Mantel-Haenszel weights are used for stratified samples.

Value

A list with two components:

- `data` A data frame containing the input exposure and number of events for each treatment group. It has the following variables:
  - `t1`: The exposure for the active treatment group.
  - `y1`: The number of events for the active treatment group.
  - `t2`: The exposure for the control group.
  - `y2`: The number of events for the control group.
- `estimates`: A data frame containing the point estimate and confidence interval for rate ratio. It has the following variables:
  - `scale`: The scale of treatment effect.
  - `estimate`: The point estimate.
  - `lower`: The lower limit of the confidence interval.
  - `upper`: The upper limit of the confidence interval.
  - `cilevel`: The confidence interval level.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
mnRateRatioCI(t1 = c(10,10), y1 = c(4,3), t2 = c(20,10), y2 = c(2,0))
```

---

mnRiskDiffCI *Miettinen-Nurminen Score Confidence Interval for Two-Sample Risk Difference*

Description

Obtains the Miettinen-Nurminen score confidence interval for two-sample risk difference possibly with stratification.

Usage

```
mnRiskDiffCI(
  n1 = NA_real_,
  y1 = NA_real_,
  n2 = NA_real_,
  y2 = NA_real_,
  cilevel = 0.95
)
```

Arguments

`n1`	The sample size for the active treatment group.
`y1`	The number of responses for the active treatment group.
`n2`	The sample size for the control group.
`y2`	The number of responses for the control group.
`cilevel`	The confidence interval level.

Details

The Mantel-Haenszel sample size weights are used for stratified samples.

Value

A list with two components:

- `data` A data frame containing the input sample size and number of responses for each treatment group. It has the following variables:
  - `n1`: The sample size for the active treatment group.
  - `y1`: The number of responses for the active treatment group.
  - `n2`: The sample size for the control group.
  - `y2`: The number of responses for the control group.

- `estimates`: A data frame containing the point estimate and confidence interval for risk difference. It has the following variables:
  - `scale`: The scale of treatment effect.
  - `estimate`: The point estimate.
  - `lower`: The lower limit of the confidence interval.
  - `upper`: The upper limit of the confidence interval.
  - `cilevel`: The confidence interval level.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
mnRiskDiffCI(n1 = c(10,10), y1 = c(4,3), n2 = c(20,10), y2 = c(2,0))
```

`mnRiskRatioCI` *Miettinen-Nurminen Score Confidence Interval for Two-Sample Risk Ratio*

Description

Obtains the Miettinen-Nurminen score confidence interval for two-sample risk ratio possibly with stratification.

Usage

```
mnRiskRatioCI(
  n1 = NA_real_,
  y1 = NA_real_,
  n2 = NA_real_,
  y2 = NA_real_,
  cilevel = 0.95
)
```

Arguments

`n1`	The sample size for the active treatment group.
`y1`	The number of responses for the active treatment group.
`n2`	The sample size for the control group.
`y2`	The number of responses for the control group.
`cilevel`	The confidence interval level.

Details

The Mantel-Haenszel sample size weights are used for stratified samples.

Value

A list with two components:

- `data` A data frame containing the input sample size and number of responses for each treatment group. It has the following variables:
  - `n1`: The sample size for the active treatment group.
  - `y1`: The number of responses for the active treatment group.
  - `n2`: The sample size for the control group.
  - `y2`: The number of responses for the control group.
- `estimates`: A data frame containing the point estimate and confidence interval for risk ratio. It has the following variables:
  - `scale`: The scale of treatment effect.
  - `estimate`: The point estimate.
  - `lower`: The lower limit of the confidence interval.
  - `upper`: The upper limit of the confidence interval.
  - `cilevel`: The confidence interval level.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
mnRiskRatioCI(n1 = c(10,10), y1 = c(4,3), n2 = c(20,10), y2 = c(2,0))
```

`mTPI2Table` *mTPI-2 Decision Table*

Description

Obtains the decision table for the modified toxicity probability interval-2 (mTPI-2) design.

Usage

```
mTPI2Table(
  nMax = NA_integer_,
  pT = 0.3,
  epsilon1 = 0.05,
  epsilon2 = 0.05,
  a = 1,
  b = 1,
  pExcessTox = 0.95
)
```

Arguments

`nMax`	The maximum number of subjects in a dose cohort.
`pT`	The target toxicity probability. Defaults to 0.3.
`epsilon1`	The lower equivalence margin from the target. Defaults to 0.05.
`epsilon2`	The upper equivalence margin from the target. Defaults to 0.05.
`a`	The prior toxicity parameter for the beta prior.
`b`	The prior non-toxicity parameter for the beta prior.
`pExcessTox`	The threshold for excessive toxicity, i.e., if Prob(p > pT | Data) > pExcessTox, then the current and all higher doses will be excluded and never be used again in the remainder of the trial to avoid any other subjects receiving treatment at those doses. Defaults to 0.95.

Value

An S3 class `mTPI2Table` object with the following components:

- `settings`: The input settings data frame with the following variables:
  - `nMax`: The maximum number of subjects in a dose cohort.
  - `pT`: The target toxicity probability.
  - `epsilon1`: The lower equivalence margin from the target.
  - `epsilon2`: The upper equivalence margin from the target.
  - `a`: The prior toxicity parameter for the beta prior.
  - `b`: The prior non-toxicity parameter for the beta prior.
  - `pExcessTox`: The threshold for excessive toxicity.
- `subintervals`: The subintervals of equal length in the mTPI-2 design. It includes the following variables:
  - `lower`: The lower bound of the subinterval.
  - `upper`: The upper bound of the subinterval.
  - `decision`: The dosing decision for the subinterval.
- `decisionDataFrame`: The decision data frame for the mTPI-2 design. It includes the following variables:
  - `n`: The sample size.
  - `y`: The number of toxicities.
  - `decision`: The dosing decision.
- `decisionMatrix`: The decision matrix corresponding to the decision data frame.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

Guo, W., Wang, S. J., Yang, S., Lynn, H., & Ji, Y. (2017). A Bayesian interval dose-finding design addressing Ockham's razor: mTPI-2. Contemporary Clinical Trials, 58, 23-33.

Examples

```
mTPI2Table(nMax = 18, pT = 0.3, epsilon1 = 0.05, epsilon2 = 0.05)
```

---

natrisk *Number of Subjects at Risk*

Description

Obtains the number of subjects at risk at given analysis times for each treatment group.

Usage

```
natrisk(
  time = NA_real_,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = NA_real_,
  minFollowupTime = NA_real_,
  maxFollowupTime = NA_real_
)
```

Arguments

`time`	A vector of analysis times at which to calculate the number of patients at risk.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`lambda1`	A vector of hazard rates for the event for the active treatment group. One for each analysis time interval.

`lambda2`	A vector of hazard rates for the event for the control group. One for each analysis time interval.
`gamma1`	The hazard rate for exponential dropout, or a vector of hazard rates for piecewise exponential dropout for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, or a vector of hazard rates for piecewise exponential dropout for the control group.
`accrualDuration`	Duration of the enrollment period.
`minFollowupTime`	Follow-up time for the last enrolled subject.
`maxFollowupTime`	Follow-up time for the first enrolled subject. For fixed follow-up, `maxFollowupTime` = `minFollowupTime`. For variable follow-up, `maxFollowupTime` = `accrualDuration` + `minFollowupTime`.

Value

A matrix of the number of patients at risk at the specified analysis times (row) for each treatment group (column).

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

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

natrisk(time = c(9, 24), allocationRatioPlanned = 1,
        accrualTime = c(0, 3), accrualIntensity = c(10, 20),
        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 = 12, minFollowupTime = 18,
        maxFollowupTime = 30)
```

`nbpower` *Power for Negative Binomial Rate Ratio*

Description

Estimates the power for negative binomial rate ratio test.

Usage

```
nbpower(
  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_,
  rateRatioH0 = 1,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  kappa1 = NA_real_,
  kappa2 = NA_real_,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  spendingTime = NA_real_,
  studyDuration = NA_real_,
  nullVariance = 0L
)
```

Arguments

`kMax`	The maximum number of stages.
`informationRates`	The information rates. 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".
`rateRatioH0`	Rate ratio under the null hypothesis.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`kappa1`	The dispersion parameter (reciprocal of the shape parameter of the gamma mixing distribution) for the active treatment group by stratum.
`kappa2`	The dispersion parameter (reciprocal of the shape parameter of the gamma mixing distribution) for the control group by stratum.
`lambda1`	The rate parameter of the negative binomial distribution for the active treatment group by stratum.
`lambda2`	The rate parameter of the negative binomial distribution for the control group by stratum.

`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.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.
`studyDuration`	Study duration for fixed follow-up design. Defaults to missing, which is to be replaced with the sum of `accrualDuration` and `followupTime`. If provided, the value is allowed to be less than the sum of `accrualDuration` and `followupTime`.
`nullVariance`	Whether to calculate the variance for log rate ratio under the null hypothesis.

Value

An S3 class `nbpower` 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.
  - `exposure`: The total exposure.
  - `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.
  - `expectedExposure`: The expected exposure.
  - `expectedStudyDuration`: The expected study duration.
  - `expectedInformation`: The expected information.
  - `accrualDuration`: The accrual duration.
  - `followupTime`: The follow-up duration.
  - `fixedFollowup`: Whether a fixed follow-up design is used.
  - `kMax`: The number of stages.
  - `rateRatioH0`: The rate ratio under the null hypothesis.
  - `rateRatio`: The rate ratio.

- `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.
    - `exposure`: The exposure.
    - `analysisTime`: The average time since trial start.
    - `efficacyRateRatio`: The efficacy boundaries on the rate ratio scale.
    - `futilityRateRatio`: The futility boundaries on the rate ratio scale.
    - `efficacyP`: The efficacy boundaries on the p-value scale.
    - `futilityP`: The futility boundaries on the p-value scale.
    - `information`: The cumulative information.
    - `efficacyStopping`: Whether to allow efficacy stopping.
    - `futilityStopping`: Whether to allow futility stopping.
- `settings`: A list containing the following input parameters: `typeAlphaSpending`, `parameterAlphaSpending`, `userAlphaSpending`, `typeBetaSpending`, `parameterBetaSpending`, `allocationRatioPlanned`, `accrualTime`, `accuralIntensity`, `piecewiseSurvivalTime`, `kappa1`, `kappa2`, `lambda1`, `lambda2`, `gamma1`, `gamma2`, `spendingTime`, and `nullVariance`.
- `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.
    - `exposure1`: The exposure 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.
    - `exposure2`: The exposure 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.
    - `expectedExposure1`: The expected exposure 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.
    - `expectedExposure2`: The expected exposure for the control group.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

See Also

`nbstat`

Examples

```
# Example 1: Variable follow-up design

nbpower(kMax = 2, informationRates = c(0.5, 1),
        alpha = 0.025, typeAlphaSpending = "sfOF",
        accrualIntensity = 1956/1.25,
        stratumFraction = c(0.2, 0.8),
        kappa1 = 5, kappa2 = 5,
        lambda1 = c(0.7*0.125, 0.75*0.25),
        lambda2 = c(0.125, 0.25),
        gamma1 = 0, gamma2 = 0,
        accrualDuration = 1.25,
        followupTime = 2.75, fixedFollowup = FALSE,
        nullVariance = 1)

# Example 2: Fixed follow-up design

nbpower(kMax = 2, informationRates = c(0.5, 1),
        alpha = 0.025, typeAlphaSpending = "sfOF",
        accrualIntensity = 220/1.5,
        kappa1 = 3, kappa2 = 3,
        lambda1 = 0.5*8.4, lambda2 = 8.4,
        gamma1 = 0, gamma2 = 0,
        accrualDuration = 1.5,
        followupTime = 0.5, fixedFollowup = TRUE)
```

`nbpower1s` *Power for One-Sample Negative Binomial Rate*

Description

Estimates the power, stopping probabilities, and expected sample size in a one-group negative binomial design.

Usage

```
nbpower1s(
  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_,
  lambdaH0 = NA_real_,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  kappa = NA_real_,
  lambda = NA_real_,
  gamma = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  spendingTime = NA_real_,
  studyDuration = NA_real_
)
```

Arguments

`kMax`	The maximum number of stages.
`informationRates`	The information rates. 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".
`lambdaH0`	The rate parameter of the negative binomial distribution under the null hypothesis.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`kappa`	The dispersion parameter (reciprocal of the shape parameter of the gamma mixing distribution) of the negative binomial distribution by stratum.
`lambda`	The rate parameter of the negative binomial distribution under the alternative hypothesis by stratum.
`gamma`	The hazard rate for exponential dropout or a vector of hazard rates for piecewise exponential dropout by stratum. Defaults to 0 for no dropout.
`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.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.
`studyDuration`	Study duration for fixed follow-up design. Defaults to missing, which is to be replaced with the sum of `accrualDuration` and `followupTime`. If provided, the value is allowed to be less than the sum of `accrualDuration` and `followupTime`.

Value

An S3 class `nbpower1s` object with 3 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.
  - `exposure`: The total exposure.
  - `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.
  - `expectedExposure`: The expected exposure.
  - `expectedStudyDuration`: The expected study duration.
  - `expectedInformation`: The expected information.
  - `accrualDuration`: The accrual duration.
  - `followupTime`: The follow-up duration.
  - `fixedFollowup`: Whether a fixed follow-up design is used.
  - `kMax`: The number of stages.
  - `lambdaH0`: The rate parameter of the negative binomial distribution under the null hypothesis.
  - `lambda`: The overall rate parameter of the negative binomial distribution under the alternative hypothesis.
- `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.
  - `exposure`: The exposure.
  - `analysisTime`: The average time since trial start.
  - `efficacyRate`: The efficacy boundaries on the rate scale.
  - `futilityRate`: The futility boundaries on the rate scale.

  - `efficacyP`: The efficacy boundaries on the p-value scale.
  - `futilityP`: The futility boundaries on the p-value scale.
  - `information`: The cumulative information.
  - `efficacyStopping`: Whether to allow efficacy stopping.
  - `futilityStopping`: Whether to allow futility stopping.
- `settings`: A list containing the following input parameters: `typeAlphaSpending`, `parameterAlphaSpending`, `userAlphaSpending`, `typeBetaSpending`, `parameterBetaSpending`, `accrualTime`, `accuralIntensity`, `piecewiseSurvivalTime`, `stratumFraction`, `kappa`, `lambda`, `gamma`, and `spendingTime`.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

See Also

`nbstat`

Examples

```
# Example 1: Variable follow-up design

nbpower1s(kMax = 2, informationRates = c(0.5, 1),
          alpha = 0.025, typeAlphaSpending = "sfOF",
          lambdaH0 = 0.125, accrualIntensity = 500,
          stratumFraction = c(0.2, 0.8),
          kappa = c(3, 5), lambda = c(0.0875, 0.085),
          gamma = 0, accrualDuration = 1.25,
          followupTime = 2.75, fixedFollowup = FALSE)

# Example 2: Fixed follow-up design

nbpower1s(kMax = 2, informationRates = c(0.5, 1),
          alpha = 0.025, typeAlphaSpending = "sfOF",
          lambdaH0 = 8.4, accrualIntensity = 40,
          kappa = 3, lambda = 0.5*8.4,
          gamma = 0, accrualDuration = 1.5,
          followupTime = 0.5, fixedFollowup = TRUE)
```

`nbpowerequiv` *Power for Equivalence in Negative Binomial Rate Ratio*

Description

Obtains the power for equivalence in negative binomial rate ratio.

Usage

```
nbpowerequiv(
  kMax = 1L,
  informationRates = NA_real_,
  criticalValues = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  rateRatioLower = NA_real_,
  rateRatioUpper = NA_real_,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  kappa1 = NA_real_,
  kappa2 = NA_real_,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  spendingTime = NA_real_,
  studyDuration = NA_real_
)
```

Arguments

`kMax`	The maximum number of stages.
`informationRates`	The information rates. Defaults to `(1:kMax) / kMax` if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`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.
`rateRatioLower`	The lower equivalence limit of rate ratio.
`rateRatioUpper`	The upper equivalence limit of rate ratio.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`kappa1`	The dispersion parameter (reciprocal of the shape parameter of the gamma mixing distribution) for the active treatment group by stratum.
`kappa2`	The dispersion parameter (reciprocal of the shape parameter of the gamma mixing distribution) for the control group by stratum.
`lambda1`	The rate parameter of the negative binomial distribution for the active treatment group by stratum.
`lambda2`	The rate parameter of the negative binomial distribution for the control group by stratum.
`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.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.
`studyDuration`	Study duration for fixed follow-up design. Defaults to missing, which is to be replaced with the sum of `accrualDuration` and `followupTime`. If provided, the value is allowed to be less than the sum of `accrualDuration` and `followupTime`.

Value

An S3 class `nbpowerequiv` 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.
  - `numbeOfSubjects`: The total number of subjects.
  - `exposure`: The total exposure.
  - `studyDuration`: The total study duration.
  - `information`: The maximum information.
  - `expectedNumberOfEvents`: The expected number of events.
  - `expectedNumberOfSubjects`: The expected number of subjects.
  - `expectedExposure`: The expected exposure.
  - `expectedStudyDuration`: The expected study duration.
  - `expectedInformation`: The expected information.
  - `kMax`: The number of stages.
  - `rateRatioLower`: The lower equivalence limit of rate ratio.
  - `rateRatioUpper`: The upper equivalence limit of rate ratio.
  - `rateRatio`: The rate ratio.
  - `accrualDuration`: The accrual duration.
  - `followupTime`: The follow-up duration.
  - `fixedFollowup`: Whether a fixed follow-up design is used.
- `byStageResults`: A data frame containing the following variables:
  - `informationRates`: The information rates.
  - `efficacyBounds`: The efficacy boundaries on the Z-scale for each of the two one-sided tests.
  - `rejectPerStage`: The probability for efficacy stopping.
  - `cumulativeRejection`: The cumulative probability for efficacy stopping.
  - `cumulativeAlphaSpent`: The cumulative alpha for each of the two one-sided tests.
  - `cumulativeAttainedAlphaH10`: The cumulative alpha attained under `H10`.
  - `cumulativeAttainedAlphaH20`: The cumulative alpha attained under `H20`.
  - `numberOfEvents`: The number of events.
  - `numberOfDropouts`: The number of dropouts.
  - `numberOfSubjects`: The number of subjects.
  - `exposure`: The exposure.
  - `analysisTime`: The average time since trial start.
  - `efficacyRateRatioLower`: The efficacy boundaries on the rate ratio scale for the one-sided null hypothesis at the lower equivalence limit.
  - `efficacyRateRatioUpper`: The efficacy boundaries on the rate ratio scale for the one-sided null hypothesis at the upper equivalence limit.
  - `efficacyP`: The efficacy bounds on the p-value scale for each of the two one-sided tests.

  - `information`: The cumulative information.
- `settings`: A list containing the following input parameters: `typeAlphaSpending`, `parameterAlphaSpending`, `userAlphaSpending`, `allocationRatioPlanned`, `accrualTime`, `accuralIntensity`, `piecewiseSurvivalTime`, `stratumFraction`, `kappa1`, `kappa2`, `lambda1`, `lambda2`, `gamma1`, `gamma2`, `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.
  - `exposure1`: The exposure 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.
  - `exposure2`: The exposure 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.
  - `expectedExposure1`: The expected exposure 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.
  - `expectedExposure2`: The expected exposure for the control group.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

See Also

`nbstat`

Examples

```
# Example 1: Variable follow-up design
nbpowerequiv(kMax = 2, informationRates = c(0.5, 1),
             alpha = 0.05, typeAlphaSpending = "sfOF",
             rateRatioLower = 2/3, rateRatioUpper = 3/2,
             accrualIntensity = 1956/1.25,
             kappa1 = 5, kappa2 = 5,
             lambda1 = 0.125, lambda2 = 0.125,
             gamma1 = 0, gamma2 = 0,
             accrualDuration = 1.25,
             followupTime = 2.75, fixedFollowup = FALSE)

# Example 2: Fixed follow-up design
nbpowerequiv(kMax = 2, informationRates = c(0.5, 1),
             alpha = 0.05, typeAlphaSpending = "sfOF",
             rateRatioLower = 0.5, rateRatioUpper = 2,
```

```
               accrualIntensity = 220/1.5,
               stratumFraction = c(0.2, 0.8),
               kappa1 = 3, kappa2 = 3,
               lambda1 = c(8.4, 10.2),
               lambda2 = c(8.0, 11.5),
               gamma1 = 0, gamma2 = 0,
               accrualDuration = 1.5,
               followupTime = 0.5, fixedFollowup = TRUE)
```

`nbsamplesize` *Sample Size for Negative Binomial Rate Ratio*

Description

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

Usage

```
nbsamplesize(
  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_,
  rateRatioH0 = 1,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  kappa1 = NA_real_,
  kappa2 = NA_real_,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
```

```
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  interval = as.numeric(c(0.001, 240)),
  spendingTime = NA_real_,
  rounding = 1L,
  nullVariance = 0L
)
```

Arguments

`beta`	Type II error. Defaults to 0.2.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. 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.
`rateRatioH0`	Rate ratio under the null hypothesis.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`kappa1`	The dispersion parameter (reciprocal of the shape parameter of the gamma mixing distribution) for the active treatment group by stratum.
`kappa2`	The dispersion parameter (reciprocal of the shape parameter of the gamma mixing distribution) for the control group by stratum.
`lambda1`	The rate parameter of the negative binomial distribution for the active treatment group by stratum.
`lambda2`	The rate parameter of the negative binomial distribution for the control group by stratum.
`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.
`interval`	The interval to search for the solution of accrualDuration, followupDuration, or the proportionality constant of accrualIntensity. Defaults to `c(0.001, 240)`.
`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. Defaults to 1 for sample size rounding.
`nullVariance`	Whether to calculate the variance for log rate ratio under the null hypothesis.

Value

A list of two components:

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

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

See Also

`nbpower`

Examples

```
# Example 1: Obtains follow-up duration given power, accrual intensity,
# and accrual duration for variable follow-up

nbsamplesize(beta = 0.2, kMax = 2,
             informationRates = c(0.5, 1),
             alpha = 0.025, typeAlphaSpending = "sfOF",
             accrualIntensity = 1956/1.25,
             kappa1 = 5, kappa2 = 5,
             lambda1 = 0.0875, lambda2 = 0.125,
             gamma1 = 0, gamma2 = 0,
             accrualDuration = 1.25,
             followupTime = NA, fixedFollowup = FALSE)

# Example 2: Obtains accrual intensity given power, accrual duration, and
# follow-up duration for variable follow-up

nbsamplesize(beta = 0.2, kMax = 2,
             informationRates = c(0.5, 1),
             alpha = 0.025, typeAlphaSpending = "sfOF",
             accrualIntensity = 100,
             kappa1 = 5, kappa2 = 5,
             lambda1 = 0.0875, lambda2 = 0.125,
             gamma1 = 0, gamma2 = 0,
             accrualDuration = 1.25,
             followupTime = 2.25, fixedFollowup = FALSE)


# Example 3: Obtains accrual duration given power, accrual intensity, and
# follow-up duration for fixed follow-up

nbsamplesize(beta = 0.2, kMax = 2,
             informationRates = c(0.5, 1),
             alpha = 0.025, typeAlphaSpending = "sfOF",
             accrualIntensity = 1667,
             stratumFraction = c(0.2, 0.8),
             kappa1 = 5, kappa2 = 5,
```

```
              lambda1 = c(0.7*0.125, 0.75*0.25),
              lambda2 = c(0.125, 0.25),
              gamma1 = 0, gamma2 = 0,
              accrualDuration = NA,
              followupTime = 0.5, fixedFollowup = TRUE)
```

`nbsamplesize1s` *Sample Size for One-Sample Negative Binomial Rate*

Description

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

Usage

```
nbsamplesize1s(
  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_,
  lambdaH0 = NA_real_,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  kappa = NA_real_,
  lambda = NA_real_,
  gamma = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  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. 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.
`lambdaH0`	The rate parameter of the negative binomial distribution under the null hypothesis.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).

`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`kappa`	The dispersion parameter (reciprocal of the shape parameter of the gamma mixing distribution) of the negative binomial distribution by stratum.
`lambda`	The rate parameter of the negative binomial distribution under the alternative hypothesis by stratum.
`gamma`	The hazard rate for exponential dropout or a vector of hazard rates for piecewise exponential dropout by stratum. Defaults to 0 for no dropout.
`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.
`interval`	The interval to search for the solution of accrualDuration, followupDuration, or the proportionality constant of accrualIntensity. Defaults to `c(0.001, 240)`.
`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. Defaults to 1 for sample size rounding.

Value

A list of two components:

- `resultsUnderH1`: An S3 class `nbpower1s` object under the alternative hypothesis.
- `resultsUnderH0`: An S3 class `nbpower1s` object under the null hypothesis.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

See Also

`nbpower1s`

Examples

```
# Example 1: Obtains follow-up duration given power, accrual intensity,
# and accrual duration for variable follow-up

nbsamplesize1s(beta = 0.2, kMax = 2,
               informationRates = c(0.5, 1),
               alpha = 0.025, typeAlphaSpending = "sfOF",
```

```
               lambdaH0 = 0.125, accrualIntensity = 500,
               stratumFraction = c(0.2, 0.8),
               kappa = c(3, 5), lambda = c(0.0875, 0.085),
               gamma = 0, accrualDuration = 1.25,
               followupTime = NA, fixedFollowup = FALSE)

# Example 2: Obtains accrual intensity given power, accrual duration, and
# follow-up duration for variable follow-up

nbsamplesize1s(beta = 0.2, kMax = 2,
               informationRates = c(0.5, 1),
               alpha = 0.025, typeAlphaSpending = "sfOF",
               lambdaH0 = 0.125, accrualIntensity = 100,
               kappa = 5, lambda = 0.0875,
               gamma = 0, accrualDuration = 1.25,
               followupTime = 2.25, fixedFollowup = FALSE)


# Example 3: Obtains accrual duration given power, accrual intensity, and
# follow-up duration for fixed follow-up

nbsamplesize1s(beta = 0.2, kMax = 2,
               informationRates = c(0.5, 1),
               alpha = 0.025, typeAlphaSpending = "sfOF",
               lambdaH0 = 8.4, accrualIntensity = 40,
               kappa = 3, lambda = 4.2,
               gamma = 0, accrualDuration = NA,
               followupTime = 0.5, fixedFollowup = TRUE)
```

`nbsamplesizeequiv` *Sample Size for Equivalence in Negative Binomial Rate Ratio*

Description

Obtains the sample size for equivalence in negative binomial rate ratio.

Usage

```
nbsamplesizeequiv(
  beta = 0.2,
  kMax = 1L,
  informationRates = NA_real_,
  criticalValues = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  rateRatioLower = NA_real_,
```

```
  rateRatioUpper = NA_real_,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  kappa1 = NA_real_,
  kappa2 = NA_real_,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  interval = as.numeric(c(0.001, 240)),
  spendingTime = NA_real_,
  rounding = 1L
)
```

Arguments

`beta`	The type II error.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Defaults to `(1:kMax) / kMax` if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`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.
`rateRatioLower`	The lower equivalence limit of rate ratio.
`rateRatioUpper`	The upper equivalence limit of rate ratio.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.

`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`kappa1`	The dispersion parameter (reciprocal of the shape parameter of the gamma mixing distribution) for the active treatment group by stratum.
`kappa2`	The dispersion parameter (reciprocal of the shape parameter of the gamma mixing distribution) for the control group by stratum.
`lambda1`	The rate parameter of the negative binomial distribution for the active treatment group by stratum.
`lambda2`	The rate parameter of the negative binomial distribution for the control group by stratum.
`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.
`interval`	The interval to search for the solution of accrualDuration, followupDuration, or the proportionality constant of accrualIntensity. Defaults to `c(0.001, 240)`.
`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. Defaults to 1 for sample size rounding.

Value

An S3 class `nbpowerequiv` object

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

See Also

nbpowerequiv

Examples

```
# Example 1: Variable follow-up design and solve for follow-up time
nbsamplesizeequiv(beta = 0.1, kMax = 2, informationRates = c(0.5, 1),
                  alpha = 0.05, typeAlphaSpending = "sfOF",
                  rateRatioLower = 2/3, rateRatioUpper = 3/2,
                  accrualIntensity = 1956/1.25,
                  stratumFraction = c(0.2, 0.8),
                  kappa1 = c(3, 5),
                  kappa2 = c(2, 3),
                  lambda1 = c(0.125, 0.165),
                  lambda2 = c(0.135, 0.175),
                  gamma1 = -log(1-0.05),
                  gamma2 = -log(1-0.10),
                  accrualDuration = 1.25,
                  followupTime = NA, fixedFollowup = FALSE)

# Example 2: Fixed follow-up design and solve for accrual duration
nbsamplesizeequiv(beta = 0.2, kMax = 2, informationRates = c(0.5, 1),
                  alpha = 0.05, typeAlphaSpending = "sfOF",
                  rateRatioLower = 0.5, rateRatioUpper = 2,
                  accrualIntensity = 220/1.5,
                  kappa1 = 3, kappa2 = 3,
                  lambda1 = 8.4, lambda2 = 8.4,
                  gamma1 = 0, gamma2 = 0,
                  accrualDuration = NA,
                  followupTime = 0.5, fixedFollowup = TRUE)
```

`nbstat` *Negative Binomial Rate Ratio*

Description

Obtains the number of subjects accrued, number of events, number of dropouts, number of subjects reaching the maximum follow-up, total exposure, and variance for log rate in each group, rate ratio, variance, and Wald test statistic of log rate ratio at given calendar times.

Usage

```
nbstat(
  time = NA_real_,
  rateRatioH0 = 1,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
```

```
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  kappa1 = NA_real_,
  kappa2 = NA_real_,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  nullVariance = 0L
)
```

Arguments

`time`	A vector of calendar times for data cut.
`rateRatioH0`	Rate ratio under the null hypothesis.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`kappa1`	The dispersion parameter (reciprocal of the shape parameter of the gamma mixing distribution) for the active treatment group by stratum.
`kappa2`	The dispersion parameter (reciprocal of the shape parameter of the gamma mixing distribution) for the control group by stratum.
`lambda1`	The rate parameter of the negative binomial distribution for the active treatment group by stratum.
`lambda2`	The rate parameter of the negative binomial distribution for the control group by stratum.
`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.
`nullVariance`	Whether to calculate the variance for log rate ratio under the null hypothesis.

Details

The probability mass function for a negative binomial distribution with dispersion parameter κ_i and rate parameter λ_i is given by

$$P(Y_{ij} = y) = \frac{\Gamma(y + 1/\kappa_i)}{\Gamma(1/\kappa_i) y!} \left(\frac{1}{1 + \kappa_i \lambda_i t_{ij}}\right)^{1/\kappa_i} \left(\frac{\kappa_i \lambda_i t_{ij}}{1 + \kappa_i \lambda_i t_{ij}}\right)^{y},$$

where Y_{ij} is the event count for subject j in treatment group i, and t_{ij} is the exposure time for the subject. If $\kappa_i = 0$, the negative binomial distribution reduces to the Poisson distribution.

For treatment group i, let $\beta_i = \log(\lambda_i)$. The log-likelihood for $\{(\kappa_i, \beta_i) : i = 1, 2\}$ can be written as

$$l = \sum_{i=1}^{2} \sum_{j=1}^{n_i} \{\log \Gamma(y_{ij} + 1/\kappa_i) - \log \Gamma(1/\kappa_i) + y_{ij} (\log(\kappa_i) + \beta_i) - (y_{ij} + 1/\kappa_i) \log(1 + \kappa_i \exp(\beta_i) t_{ij})\}.$$

It follows that

$$\frac{\partial l}{\partial \beta_i} = \sum_{j=1}^{n_i} \left\{ y_{ij} - (y_{ij} + 1/\kappa_i) \frac{\kappa_i \exp(\beta_i) t_{ij}}{1 + \kappa_i \exp(\beta_i) t_{ij}} \right\},$$

and

$$-\frac{\partial^2 l}{\partial \beta_i^2} = \sum_{j=1}^{n_i} (y_{ij} + 1/\kappa_i) \frac{\kappa_i \lambda_i t_{ij}}{(1 + \kappa_i \lambda_i t_{ij})^2}.$$

The Fisher information for β_i is

$$E\left(-\frac{\partial^2 l}{\partial \beta_i^2}\right) = n_i E\left(\frac{\lambda_i t_{ij}}{1 + \kappa_i \lambda_i t_{ij}}\right).$$

In addition, we can show that

$$E\left(-\frac{\partial^2 l}{\partial \beta_i \partial \kappa_i}\right) = 0.$$

Therefore, the variance of $\hat{\beta}_i$ is

$$Var(\hat{\beta}_i) = \frac{1}{n_i} \left\{ E\left(\frac{\lambda_i t_{ij}}{1 + \kappa_i \lambda_i t_{ij}}\right) \right\}^{-1}.$$

To evaluate the integral, we need to obtain the distribution of the exposure time,

$$t_{ij} = \min(\tau - W_{ij}, C_{ij}, T_{fmax}),$$

where τ denotes the calendar time since trial start, W_{ij} denotes the enrollment time for subject j in treatment group i, C_{ij} denotes the time to dropout after enrollment for subject j in treatment group i, and T_{fmax} denotes the maximum follow-up time for all subjects. Therefore,

$$P(t_{ij} \geq t) = P(W_{ij} \leq \tau - t)P(C_{ij} \geq t)I(t \leq T_{fmax}).$$

Let H denote the distribution function of the enrollment time, and G_i denote the survival function of the dropout time for treatment group i. By the change of variables, we have

$$E\left(\frac{\lambda_i t_{ij}}{1 + \kappa_i \lambda_i t_{ij}}\right) = \int_0^{\tau \wedge T_{fmax}} \frac{\lambda_i}{(1 + \kappa_i \lambda_i t)^2} H(\tau - t) G_i(t) dt.$$

A numerical integration algorithm for a univariate function can be used to evaluate the above integral.

For the restricted maximum likelihood (reml) estimate of (β_1, β_2) subject to the constraint that $\beta_1 - \beta_2 = \Delta$, we express the log-likelihood in terms of $(\beta_2, \Delta, \kappa_1, \kappa_2)$, and takes the derivative of the log-likelihood function with respect to β_2. The resulting score equation has asymptotic limit

$$E\left(\frac{\partial l}{\partial \beta_2}\right) = s_1 + s_2,$$

where

$$s_1 = nrE\left\{\lambda_1 t_{1j} - \left(\lambda_1 t_{1j} + \frac{1}{\kappa_1}\right) \frac{\kappa_1 e^{\tilde{\beta}_2 + \Delta} t_{1j}}{1 + \kappa_1 e^{\tilde{\beta}_2 + \Delta} t_{1j}}\right\},$$

and

$$s_2 = n(1 - r)E\left\{\lambda_2 t_{2j} - \left(\lambda_2 t_{2j} + \frac{1}{\kappa_2}\right) \frac{\kappa_2 e^{\tilde{\beta}_2} t_{2j}}{1 + \kappa_2 e^{\tilde{\beta}_2} t_{2j}}\right\}.$$

Here r is the randomization probability for the active treatment group. The asymptotic limit of the reml of β_2 is the solution $\tilde{\beta}_2$ to $E\left(\frac{\partial l}{\partial \beta_2}\right) = 0$.

Value

A list with two components:

- `resultsUnderH1`: A data frame containing the following variables:
  - `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.
  - `exposure`: The total exposure time.
  - `exposure1`: The exposure time for the active treatment group.
  - `exposure2`: The exposure time for the control group.
  - `rateRatio`: The rate ratio of the active treatment group versus the control group.
  - `vlogRate1`: The variance for the log rate parameter for the active treatment group.
  - `vlogRate2`: The variance for the log rate parameter for the control group.
  - `vlogRR`: The variance of log rate ratio.
  - `information`: The information of log rate ratio.
  - `zlogRR`: The Z-statistic for log rate ratio.
- `resultsUnderH0` when `nullVariance = TRUE`: A data frame with the following variables:
  - `time`: The analysis time since trial start.
  - `lambda1H0`: The restricted maximum likelihood estimate of the event rate for the active treatment group.
  - `lambda2H0`: The restricted maximum likelihood estimate of the event rate for the control group.
  - `rateRatioH0`: The rate ratio under H0.
  - `vlogRate1H0`: The variance for the log rate parameter for the active treatment group under H0.
  - `vlogRate2H0`: The variance for the log rate parameter for the control group under H0.
  - `vlogRRH0`: The variance of log rate ratio under H0.
  - `informationH0`: The information of log rate ratio under H0.
  - `zlogRRH0`: The Z-statistic for log rate ratio with variance evaluated under H0.
  - `varianceRatio`: The ratio of the variance under H0 versus the variance under H1.
  - `lambda1`: The true event rate for the active treatment group.
  - `lambda2`: The true event rate for the control group.
  - `rateRatio`: The true rate ratio.
- `resultsUnderH0` when `nullVariance = FALSE`: A data frame with the following variables:
  - `time`: The analysis time since trial start.
  - `rateRatioH0`: The rate ratio under H0.
  - `varianceRatio`: Equal to 1.
  - `lambda1`: The true event rate for the active treatment group.
  - `lambda2`: The true event rate for the control group.
  - `rateRatio`: The true rate ratio.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Example 1: Variable follow-up design

nbstat(time = c(1, 1.25, 2, 3, 4),
```

```
        accrualIntensity = 1956/1.25,
        kappa1 = 5,
        kappa2 = 5,
        lambda1 = 0.7*0.125,
        lambda2 = 0.125,
        gamma1 = 0,
        gamma2 = 0,
        accrualDuration = 1.25,
        followupTime = 2.75)

# Example 2: Fixed follow-up design

nbstat(time = c(0.5, 1, 1.5, 2),
        accrualIntensity = 220/1.5,
        stratumFraction = c(0.2, 0.8),
        kappa1 = 3,
        kappa2 = 3,
        lambda1 = c(0.5*8.4, 0.6*10.5),
        lambda2 = c(8.4, 10.5),
        gamma1 = 0,
        gamma2 = 0,
        accrualDuration = 1.5,
        followupTime = 0.5,
        fixedFollowup = 1,
        nullVariance = 1)
```

`nevent` *Number of Subjects Having an Event*

Description

Obtains the number of subjects having an event by given analysis times for each treatment group.

Usage

```
nevent(
  time = NA_real_,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = NA_real_,
  minFollowupTime = NA_real_,
  maxFollowupTime = NA_real_
)
```

Arguments

`time`	A vector of analysis times at which to calculate the number of patients having an event.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`lambda1`	A vector of hazard rates for the event for the active treatment group. One for each analysis time interval.
`lambda2`	A vector of hazard rates for the event for the control group. One for each analysis time interval.
`gamma1`	The hazard rate for exponential dropout, or a vector of hazard rates for piecewise exponential dropout for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, or a vector of hazard rates for piecewise exponential dropout for the control group.
`accrualDuration`	Duration of the enrollment period.
`minFollowupTime`	Follow-up time for the last enrolled subject.
`maxFollowupTime`	Follow-up time for the first enrolled subject. For fixed follow-up, `maxFollowupTime` = `minFollowupTime`. For variable follow-up, `maxFollowupTime` = `accrualDuration` + `minFollowupTime`.

Value

A matrix of the number of patients having an event at the specified analysis times (row) for each treatment group (column).

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

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

```
nevent(time = c(9, 24), allocationRatioPlanned = 1,
       accrualTime = c(0, 3), accrualIntensity = c(10, 20),
       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 = 12, minFollowupTime = 18,
       maxFollowupTime = 30)
```

---

nevent2 *Number of Subjects Having an Event by Calendar Time*

Description

Obtains the number of subjects having an event by given calendar times for each treatment group.

Usage

```
nevent2(
  time = NA_real_,
  allocationRatioPlanned = 1,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  lambda1 = NA_real_,
  lambda2 = NA_real_,
  gamma1 = 0L,
  gamma2 = 0L,
  accrualDuration = NA_real_,
  minFollowupTime = NA_real_,
  maxFollowupTime = NA_real_
)
```

Arguments

`time`	A vector of calendar times at which to calculate the number of patients having an event.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.

`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`lambda1`	A vector of hazard rates for the event for the active treatment group. One for each analysis time interval.
`lambda2`	A vector of hazard rates for the event for the control group. One for each analysis time interval.
`gamma1`	The hazard rate for exponential dropout, or a vector of hazard rates for piecewise exponential dropout for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, or a vector of hazard rates for piecewise exponential dropout for the control group.
`accrualDuration`	Duration of the enrollment period.
`minFollowupTime`	Follow-up time for the last enrolled subject.
`maxFollowupTime`	Follow-up time for the first enrolled subject. For fixed follow-up, `maxFollowupTime` = `minFollowupTime`. For variable follow-up, `maxFollowupTime` = `accrualDuration` + `minFollowupTime`.

Value

A matrix of the number of patients having an event at the specified calendar times (row) for each treatment group (column).

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
# Piecewise accrual, piecewise exponential survivals, and 5% dropout by
# the end of 1 year.
nevent2(time = c(9, 24), allocationRatioPlanned = 1,
        accrualTime = c(0, 3), accrualIntensity = c(10, 20),
        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 = 12, minFollowupTime = 18,
        maxFollowupTime = 30)
```

---

patrisk *Probability of Being at Risk*

Description

Obtains the probability of being at risk at given analysis times.

Usage

```
patrisk(
  time = NA_real_,
  piecewiseSurvivalTime = 0L,
  lambda = NA_real_,
  gamma = 0L
)
```

Arguments

`time`	A vector of analysis times at which to calculate the probability of being at risk.
`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.
`gamma`	The hazard rate for exponential dropout, or a vector of hazard rates for piecewise exponential dropout.

Value

A vector of probabilities of being at risk at the specified analysis times after enrollment for a patient in a treatment group with specified piecewise exponential survival and dropout distributions.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Piecewise exponential survival with hazard 0.0533 in the first 6
# months, and hazard 0.0309 thereafter, and 5% dropout by the end of
# 1 year.

patrisk(time = c(3, 9), piecewiseSurvivalTime = c(0, 6),
        lambda = c(0.0533, 0.0309), gamma = -log(1-0.05)/12)
```

`pbvnorm` *Distribution Function of the Standard Bivariate Normal*

Description

Computes the cumulative distribution function (CDF) of the standard bivariate normal distribution with specified lower and upper integration limits and correlation coefficient.

Usage

```
pbvnorm(lower = c(-Inf, Inf), upper = c(Inf, Inf), corr = 0)
```

Arguments

`lower`	A numeric vector of length 2 specifying the lower limits of integration.
`upper`	A numeric vector of length 2 specifying the upper limits of integration.
`corr`	A numeric value specifying the correlation coefficient of the standard bivariate normal distribution.

Details

This function evaluates the probability $P(\texttt{lower[1]} < X < \texttt{upper[1]}, \texttt{lower[2]} < Y < \texttt{upper[2]})$ where (X, Y) follows a standard bivariate normal distribution with correlation `corr`.

Value

A numeric value representing the probability that a standard bivariate normal vector falls within the specified rectangular region.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
pbvnorm(c(-1, -1), c(1, 1), 0.5)
```

`pevent` *Probability of Having an Event*

Description

Obtains the probability of having an event at given analysis times.

Usage

```
pevent(
  time = NA_real_,
  piecewiseSurvivalTime = 0L,
  lambda = NA_real_,
  gamma = 0L
)
```

Arguments

`time`	A vector of analysis times at which to calculate the probability of having an event.
`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.
`gamma`	The hazard rate for exponential dropout, or a vector of hazard rates for piecewise exponential dropout.

Value

A vector of probabilities of having an event at the specified analysis times after enrollment for a patient in a treatment group with specified piecewise exponential survival and dropout distributions.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Piecewise exponential survival with hazard 0.0533 in the first 6
# months, and hazard 0.0309 thereafter, and 5% dropout by the end of
# 1 year.

pevent(time = c(3, 9), piecewiseSurvivalTime = c(0, 6),
       lambda = c(0.0533, 0.0309), gamma = -log(1-0.05)/12)
```

`phregr` *Proportional Hazards Regression Models*

Description

Obtains the hazard ratio estimates from the proportional hazards regression model with right censored or counting process data.

Usage

```
phregr(
  data,
  rep = "",
  stratum = "",
  time = "time",
  time2 = "",
  event = "event",
  covariates = "",
  weight = "",
  offset = "",
  id = "",
  ties = "efron",
  init = NA_real_,
  robust = FALSE,
  est_basehaz = TRUE,
  est_resid = TRUE,
  firth = FALSE,
  plci = FALSE,
  alpha = 0.05,
  maxiter = 50,
  eps = 1e-09
)
```

Arguments

`data` The input data frame that contains the following variables:

- `rep`: The replication for by-group processing.
- `stratum`: The stratum.
- `time`: The follow-up time for right censored data, or the left end of each interval for counting process data.
- `time2`: The right end of each interval for counting process data. Intervals are assumed to be open on the left and closed on the right, and event indicates whether an event occurred at the right end of each interval.
- `event`: The event indicator, 1=event, 0=no event.
- `covariates`: The values of baseline covariates (and time-dependent covariates in each interval for counting process data).

- `weight`: The weight for each observation.
- `offset`: The offset for each observation.
- `id`: The optional subject ID for counting process data with time-dependent covariates.

`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 or the left end of each interval for counting process data in the input data.
`time2`	The name of the right end of each interval for counting process data in the input data.
`event`	The name of the event variable in the input data.
`covariates`	The vector of names of baseline and time-dependent covariates in the input data.
`weight`	The name of the weight variable in the input data.
`offset`	The name of the offset variable in the input data.
`id`	The name of the id variable in the input data.
`ties`	The method for handling ties, either "breslow" or "efron" (default).
`init`	The vector of initial values. Defaults to zero for all variables.
`robust`	Whether a robust sandwich variance estimate should be computed. In the presence of the id variable, the score residuals will be aggregated for each id when computing the robust sandwich variance estimate.
`est_basehaz`	Whether to estimate the baseline hazards. Defaults to `TRUE`.
`est_resid`	Whether to estimate the martingale residuals. Defaults to `TRUE`.
`firth`	Whether to use Firth's penalized likelihood method. Defaults to `FALSE`.
`plci`	Whether to obtain profile likelihood confidence interval.
`alpha`	The two-sided significance level.
`maxiter`	The maximum number of iterations.
`eps`	The tolerance to declare convergence.

Value

A list with the following components:

- `sumstat`: The data frame of summary statistics of model fit with the following variables:
  - `n`: The number of observations.
  - `nevents`: The number of events.
  - `loglik0`: The (penalized) log-likelihood under null.
  - `loglik1`: The maximum (penalized) log-likelihood.
  - `scoretest`: The score test statistic.
  - `niter`: The number of Newton-Raphson iterations.
  - `ties`: The method for handling ties, either "breslow" or "efron".
  - `p`: The number of columns of the Cox model design matrix.

  - `robust`: Whether to use the robust variance estimate.
  - `firth`: Whether to use Firth's penalized likelihood method.
  - `fail`: Whether the model fails to converge.
  - `loglik0_unpenalized`: The unpenalized log-likelihood under null.
  - `loglik1_unpenalized`: The maximum unpenalized log-likelihood.
  - `rep`: The replication.
- `parest`: The data frame of parameter estimates with the following variables:
  - `param`: The name of the covariate for the parameter estimate.
  - `beta`: The log hazard ratio estimate.
  - `sebeta`: The standard error of log hazard ratio estimate.
  - `z`: The Wald test statistic for log hazard ratio.
  - `expbeta`: The hazard ratio estimate.
  - `vbeta`: The covariance matrix for parameter estimates.
  - `lower`: The lower limit of confidence interval.
  - `upper`: The upper limit of confidence interval.
  - `p`: The p-value from the chi-square test.
  - `method`: The method to compute the confidence interval and p-value.
  - `sebeta_naive`: The naive standard error of log hazard ratio estimate if robust variance is requested.
  - `vbeta_naive`: The naive covariance matrix for parameter estimates if robust variance is requested.
  - `rep`: The replication.
- `basehaz`: The data frame of baseline hazards with the following variables (if est_basehaz is TRUE):
  - `time`: The observed event time.
  - `nrisk`: The number of patients at risk at the time point.
  - `nevent`: The number of events at the time point.
  - `haz`: The baseline hazard at the time point.
  - `varhaz`: The variance of the baseline hazard at the time point assuming the parameter beta is known.
  - `gradhaz`: The gradient of the baseline hazard with respect to beta at the time point (in the presence of covariates).
  - `stratum`: The stratum.
  - `rep`: The replication.
- `residuals`: The martingale residuals.
- `p`: The number of parameters.
- `param`: The parameter names.
- `beta`: The parameter estimate.
- `vbeta`: The covariance matrix for parameter estimates.
- `vbeta_naive`: The naive covariance matrix for parameter estimates.
- `terms`: The terms object.

- `xlevels`: A record of the levels of the factors used in fitting.
- `data`: The input data.
- `rep`: The name(s) of the replication variable(s).
- `stratum`: The name(s) of the stratum variable(s).
- `time`: The name of the time varaible.
- `time2`: The name of the time2 variable.
- `event`: The name of the event variable.
- `covariates`: The names of baseline covariates.
- `weight`: The name of the weight variable.
- `offset`: The name of the offset variable.
- `id`: The name of the id variable.
- `ties`: The method for handling ties.
- `robust`: Whether a robust sandwich variance estimate should be computed.
- `est_basehaz`: Whether to estimate the baseline hazards.
- `est_resid`: Whether to estimate the martingale residuals.
- `firth`: Whether to use Firth's penalized likelihood method.
- `plci`: Whether to obtain profile likelihood confidence interval.
- `alpha`: The two-sided significance level.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

Per K. Anderson and Richard D. Gill. Cox's regression model for counting processes, a large sample study. Annals of Statistics 1982; 10:1100-1120.

Terry M. Therneau and Patricia M. Grambsch. Modeling Survival Data: Extending the Cox Model. Springer-Verlag, 2000.

Examples

```
library(dplyr)

# Example 1 with right-censored data
(fit1 <- phregr(
  data = rawdata %>% mutate(treat = 1*(treatmentGroup == 1)),
  rep = "iterationNumber", stratum = "stratum",
  time = "timeUnderObservation", event = "event",
  covariates = "treat", est_basehaz = FALSE, est_resid = FALSE))

# Example 2 with counting process data and robust variance estimate
(fit2 <- phregr(
  data = heart %>% mutate(rx = as.numeric(transplant) - 1),
  time = "start", time2 = "stop", event = "event",
```

```
  covariates = c("rx", "age"), id = "id",
  robust = TRUE, est_basehaz = TRUE, est_resid = TRUE))
```

---

ptpwexp *Distribution Function of Truncated Piecewise Exponential Distribution*

Description

Obtains the probability of a truncated piecewise exponential distribution.

Usage

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

Arguments

`q`	The vector of quantiles.
`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 <= x), otherwise, P(X > x).
`log.p`	Logical; if TRUE, probabilities p are given as log(p).

Value

The probability p such that P(X > q | X > lowerBound) = 1 - p.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
ptpwexp(q = c(8, 18), piecewiseSurvivalTime = c(0, 6, 9, 15),
        lambda = c(0.025, 0.04, 0.015, 0.007))
```

`pwexpcuts` *Piecewise Exponential Approximation to a Survival Distribution*

Description

Obtains the piecewise exponential distribution that approximates a survival distribution.

Usage

```
pwexpcuts(S, ..., tol = 1e-04)
```

Arguments

`S`	The survival function of a univariate survival time.
`...`	Additional arguments to be passed to S.
`tol`	The tolerance for convergence of the profile log-likelihood. Defaults to 0.0001.

Details

This function computes the piecewise exponential approximation to a survival distribution. The piecewise exponential model divides the time axis into J intervals defined by the change points, where each interval $[t_j, t_{j+1})$ has a constant hazard rate λ_j. The time intervals are specified as:

$$[t_1, t_2), [t_2, t_3), \ldots, [t_J, t_{J+1})$$

where $t_1 = 0$, $t_{J+1} = \infty$, and $t_j = \tau_{j-1}$ for $j = 2, \ldots, J$. The function starts with $J = 2$ (1 change point) and gradually increases J by adding one change point at a time until the maximized profile log-likelihood for change points stabilizes, i.e., the relative increase in the maximum of the profile log-likelihood function is less than `tol`. If the relative change in the hazard rate is also less than `tol`, the function stops and returns the results.

Value

A list with three components:

- `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).
- `lambda`: A vector of hazard rates for the event. One for each analysis time interval.
- `loglik`: The sequence of the asymptotic limit of the piecewise exponential log-likelihood for an increasing number of change points.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Example 1: Piecewise exponential
pwexpcuts(ptpwexp, piecewiseSurvivalTime = c(0, 3.4, 5.5),
          lambda = c(0.0168, 0.0833, 0.0431), lowerBound = 0,
          lower.tail = FALSE)

# Example 2: Weibull
pwexpcuts(pweibull, shape = 1.37, scale = 1/0.818, lower.tail = FALSE)
```

`pwexploglik` *Profile Log-Likelihood Function for Change Points in Piecewise Exponential Approximation*

Description

Obtains the profile log-likelihood function for change points in the piecewise exponential approximation to a survival function.

Usage

```
pwexploglik(tau, S, ...)
```

Arguments

`tau`	The numeric vector of change points.
`S`	The survival function of a univariate survival time.
`...`	Additional arguments to be passed to S.

Details

This function computes the profile log-likelihood for change points in a piecewise exponential survival model.

Let $S(t)$ denote the survival function of a univariate survival time, and τ be a vector of $J-1$ change points. The piecewise exponential survival model divides the time axis into J intervals defined by the change points τ, where each interval $[t_j, t_{j+1})$ has a constant hazard rate λ_j. The time intervals are specified as:

$$[t_1, t_2), [t_2, t_3), \ldots, [t_J, t_{J+1})$$

where $t_1 = 0$, $t_{J+1} = \infty$, and $t_j = \tau_{j-1}$ for $j = 2, \ldots, J$.

For each subject, the expected number of events occurring in the j-th interval is

$$d_j = E\{I(t_j < Y \leq t_{j+1})\} = S(t_j) - S(t_{j+1})$$

The expected exposure in the j-th interval is:

$$e_j = E\{(Y - t_j)I(t_j < Y \leq t_{j+1}) + (t_{j+1} - t_j)I(Y > t_{j+1})\}$$

which can be shown to be equivalent to

$$e_j = \int_{t_j}^{t_{j+1}} S(t) dt$$

The log-likelihood for the piecewise exponential model is:

$$\ell(\tau, \lambda) = \sum_{j=1}^{J} \{d_j \log(\lambda_j) - e_j \lambda_j\}$$

The profile log-likelihood for τ is obtained by maximizing $\ell(\tau, \lambda)$ with respect to λ for fixed τ. The maximum likelihood estimate of the hazard rate in the j-th interval is

$$\lambda_j = \frac{d_j}{e_j}$$

Substituting back, the profile log-likelihood is

$$\ell(\tau) = \sum_{j=1}^{J} d_j \log(d_j/e_j) - 1$$

where we use the fact that $\sum_{j=1}^{J} d_j = 1$.

Value

A list with the following three components:

- `piecewiseSurvivalTime`: A vector that specifies the starting time of piecewise exponential survival time intervals.
- `lambda`: A vector of hazard rates for the event. One for each analysis time interval.
- `loglik`: The value of the profile log-likelihood.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
pwexploglik(tau = c(0.5, 1.2, 2.8), pweibull,
            shape = 1.37, scale = 1/0.818, lower.tail = FALSE)
```

`qrcpp` *QR Decomposition of a Matrix*

Description

Computes the QR decomposition of a matrix.

Usage

```
qrcpp(X, tol = 1e-12)
```

Arguments

`X`	A numeric matrix whose QR decomposition is to be computed.
`tol`	The tolerance for detecting linear dependencies in the columns of `X`.

Details

This function performs Householder QR with column pivoting: Given an m-by-n matrix A with $m \geq n$, the following algorithm computes $r = \text{rank}(A)$ and the factorization $Q^T AP$ equal to

$$\begin{array}{ccccc} | & R_{11} & R_{12} & | & r \\ | & 0 & 0 & | & m-r \\ & r & n-r & & \end{array}$$

with $Q = H_1 \cdots H_r$ and $P = P_1 \cdots P_r$. The upper triangular part of A is overwritten by the upper triangular part of R and components $(j+1) : m$ of the jth Householder vector are stored in $A((j+1) : m, j)$. The permutation P is encoded in an integer vector `pivot`.

Value

A list with the following components:

- `qr`: A matrix with the same dimensions as `X`. The upper triangle contains the `R` of the decomposition and the lower triangle contains Householder vectors (stored in compact form).
- `rank`: The rank of `X` as computed by the decomposition.
- `pivot`: The column permutation for the pivoting strategy used during the decomposition.
- `Q`: The complete m-by-m orthogonal matrix Q.
- `R`: The complete m-by-n upper triangular matrix R.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

Gene N. Golub and Charles F. Van Loan. Matrix Computations, second edition. Baltimore, Maryland: The John Hopkins University Press, 1989, p.235.

Examples

```
hilbert <- function(n) { i <- 1:n; 1 / outer(i - 1, i, `+`) }
h9 <- hilbert(9)
qrcpp(h9)
```

---

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 <= 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 | X > 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))
```

`rawdata` *A simulated time-to-event data set with 10 replications*

Description

A simulated data set with stratification and delayed treatment effect:

`iterationNumber`	The iteration number
`arrivalTime`	The enrollment time for the subject
`stratum`	The stratum for the subject
`treatmentGroup`	The treatment group for the subject
`timeUnderObservation`	The time under observation since randomization
`event`	Whether the subject experienced the event
`dropoutEvent`	Whether the subject dropped out

Usage

```
rawdata
```

Format

An object of class `data.frame` with 4910 rows and 7 columns.

`remlOddsRatio` *REML Estimates of Individual Proportions With Specified Odds Ratio*

Description

Obtains the restricted maximum likelihood estimates of individual proportions with specified odds ratio.

Usage

```
remlOddsRatio(
  oddsRatioH0 = 1,
  n1 = NA_real_,
  y1 = NA_real_,
  n2 = NA_real_,
  y2 = NA_real_
)
```

Arguments

`oddsRatioH0`	The specified odds ratio.
`n1`	The sample size for the active treatment group.
`y1`	The number of responses for the active treatment group.
`n2`	The sample size for the control group.
`y2`	The number of responses for the control group.

Value

A vector of the restricted maximum likelihood estimates of the response probabilities for the two treatment groups.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
remlOddsRatio(oddsRatioH0 = 1.25, n1 = 10, y1 = 4, n2 = 20, y2 = 2)
```

`remlRateDiff` *REML Estimates of Individual Rates With Specified Rate Difference*

Description

Obtains the restricted maximum likelihood estimates of individual proportions with specified rate difference.

Usage

```
remlRateDiff(
  rateDiffH0 = 0,
  t1 = NA_real_,
  y1 = NA_real_,
  t2 = NA_real_,
  y2 = NA_real_
)
```

Arguments

`rateDiffH0`	The specified rate difference.
`t1`	The exposure for the active treatment group.
`y1`	The number of events for the active treatment group.
`t2`	The exposure for the control group.
`y2`	The number of events for the control group.

Value

A vector of the restricted maximum likelihood estimates of the incidence rates for the two treatment groups.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
remlRateDiff(rateDiffH0 = 0.1, t1 = 10, y1 = 4, t2 = 20, y2 = 2)
```

`remlRateRatio` *REML Estimates of Individual Rates With Specified Rate Ratio*

Description

Obtains the restricted maximum likelihood estimates of individual proportions with specified rate ratio.

Usage

```
remlRateRatio(
  rateRatioH0 = 1,
  t1 = NA_real_,
  y1 = NA_real_,
  t2 = NA_real_,
  y2 = NA_real_
)
```

Arguments

`rateRatioH0`	The specified rate ratio.
`t1`	The exposure for the active treatment group.
`y1`	The number of events for the active treatment group.
`t2`	The exposure for the control group.
`y2`	The number of events for the control group.

Value

A vector of the restricted maximum likelihood estimates of the incidence rates for the two treatment groups.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
remlRateRatio(rateRatioH0 = 1.1, t1 = 10, y1 = 4, t2 = 20, y2 = 2)
```

`remlRiskDiff` *REML Estimates of Individual Proportions With Specified Risk difference*

Description

Obtains the restricted maximum likelihood estimates of individual proportions with specified risk difference.

Usage

```
remlRiskDiff(
  riskDiffH0 = 0,
  n1 = NA_real_,
  y1 = NA_real_,
  n2 = NA_real_,
  y2 = NA_real_
)
```

Arguments

`riskDiffH0`	The specified risk difference.
`n1`	The sample size for the active treatment group.
`y1`	The number of responses for the active treatment group.
`n2`	The sample size for the control group.
`y2`	The number of responses for the control group.

Value

A vector of the restricted maximum likelihood estimates of the response probabilities for the two treatment groups.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
remlRiskDiff(riskDiffH0 = 0.1, n1 = 10, y1 = 4, n2 = 20, y2 = 0)
```

`remlRiskRatio` *REML Estimates of Individual Proportions With Specified Risk Ratio*

Description

Obtains the restricted maximum likelihood estimates of individual proportions with specified risk ratio.

Usage

```
remlRiskRatio(
  riskRatioH0 = 1,
  n1 = NA_real_,
  y1 = NA_real_,
  n2 = NA_real_,
  y2 = NA_real_
)
```

Arguments

`riskRatioH0`	The specified risk ratio.
`n1`	The sample size for the active treatment group.
`y1`	The number of responses for the active treatment group.
`n2`	The sample size for the control group.
`y2`	The number of responses for the control group.

Value

A vector of the restricted maximum likelihood estimates of the response probabilities for the two treatment groups.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
remlRiskRatio(riskRatioH0 = 1.2, n1 = 10, y1 = 4, n2 = 20, y2 = 2)
```

`repeatedPValue` *Repeated p-Values for Group Sequential Design*

Description

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

Usage

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

Arguments

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

Value

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

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

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

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

`residuals_liferegr` *Residuals for Parametric Regression Models for Failure Time Data*

Description

Obtains the response, deviance, dfbeta, and likelihood displacement residuals for a parametric regression model for failure time data.

Usage

```
residuals_liferegr(
  object,
 type = c("response", "deviance", "dfbeta", "dfbetas", "working", "ldcase", "ldresp",
    "ldshape", "matrix"),
  collapse = FALSE,
  weighted = (type %in% c("dfbeta", "dfbetas"))
)
```

Arguments

`object`	The output from the `phregr` call.
`type`	The type of residuals desired, with options including `"response"`, `"deviance"`, `"dfbeta"`, `"dfbetas"`, `"working"`, `"ldcase"`, `"ldresp"`, `"ldshape"`, and `"matrix"`.
`collapse`	Whether to collapse the residuals by `id`.
`weighted`	Whether to compute weighted residuals.

Details

The algorithms follow the `residuals.survreg` function in the `survival` package.

Value

Either a vector or a matrix of residuals, depending on the specified type:

- `response` residuals are on the scale of the original data.
- `working` residuals are on the scale of the linear predictor.
- `deviance` residuals are on the log-likelihood scale.
- `dfbeta` residuals are returned as a matrix, where the i-th row represents the approximate change in the model coefficients resulting from the inclusion of subject i.
- `dfbetas` residuals are similar to `dfbeta` residuals, but each column is scaled by the standard deviation of the corresponding coefficient.
- `matrix` residuals are a matrix of derivatives of the log-likelihood function. Let L be the log-likelihood, p be the linear predictor ($X\beta$), and s be $log(\sigma)$. Then the resulting matrix contains six columns: L, $\partial L/\partial p$, $\partial^2 L/\partial p^2$, $\partial L/\partial s$, $\partial^2 L/\partial s^2$, and $\partial L^2/\partial p\partial s$.
- `ldcase` residulas are likelihood displacement for case weight perturbation.
- `ldresp` residuals are likelihood displacement for response value perturbation.
- `ldshape` residuals are likelihood displacement related to the shape parameter.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

Escobar, L. A. and Meeker, W. Q. Assessing influence in regression analysis with censored data. Biometrics 1992; 48:507-528.

Examples

```
library(dplyr)

fit1 <- liferegr(
  data = tobin %>% mutate(time = ifelse(durable>0, durable, NA)),
  time = "time", time2 = "durable",
  covariates = c("age", "quant"), dist = "normal")

 resid <- residuals_liferegr(fit1, type = "response")
```

`residuals_phregr` *Residuals for Proportional Hazards Regression Models*

Description

Obtains the martingale, deviance, score, or Schoenfeld residuals for a proportional hazards regression model.

Usage

```
residuals_phregr(
  object,
  type = c("martingale", "deviance", "score", "schoenfeld", "dfbeta", "dfbetas",
    "scaledsch"),
  collapse = FALSE,
  weighted = (type %in% c("dfbeta", "dfbetas"))
)
```

Arguments

`object`	The output from the `phregr` call.
`type`	The type of residuals desired, with options including `"martingale"`, `"deviance"`, `"score"`, `"schoenfeld"`, `"dfbeta"`, `"dfbetas"`, and `"scaledsch"`.
`collapse`	Whether to collapse the residuals by `id`. This is not applicable for Schoenfeld type residuals.
`weighted`	Whether to compute weighted residuals.

Details

For score and Schoenfeld type residuals, the proportional hazards model must include at least one covariate. The algorithms for `deviance`, `dfbeta`, `dfbetas`, and `scaledsch` residuals follow the `residuals.coxph` function in the `survival` package.

Value

For martingale and deviance residuals, the result is a vector with one element corresponding to each subject (without `collapse`). For score residuals, the result is a matrix where each row represents a subject and each column corresponds to a variable. The row order aligns with the input data used in the original fit. For Schoenfeld residuals, the result is a matrix with one row for each event and one column per variable. These rows are sorted by time within strata, with the attributes `stratum` and `time` included.

Score residuals represent each individual's contribution to the score vector. Two commonly used transformations of this are `dfbeta`, which represents the approximate change in the coefficient vector if the observation is excluded, and `dfbetas`, which gives the approximate change in the coefficients scaled by the standard error of the coefficients.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

Terry M. Therneau, Patricia M. Grambsch, and Thomas M. Fleming. Martingale based residuals for survival models. Biometrika 1990; 77:147-160.

Patricia M. Grambsch and Terry M. Therneau. Proportional hazards tests and diagnostics based on weighted residuals. Biometrika 1994; 81:515-26.

Examples

```
library(dplyr)

# Example 1 with right-censored data
fit1 <- phregr(data = rawdata %>% filter(iterationNumber == 1) %>%
                 mutate(treat = 1*(treatmentGroup == 1)),
               stratum = "stratum",
               time = "timeUnderObservation", event = "event",
               covariates = "treat")

ressco <- residuals_phregr(fit1, type = "score")

# Example 2 with counting process data
fit2 <- phregr(data = heart %>% mutate(rx = as.numeric(transplant) - 1),
               time = "start", time2 = "stop", event = "event",
               covariates = c("rx", "age"), id = "id", robust = TRUE)

resssch <- residuals_phregr(fit2, type = "scaledsch")
```

`riskDiffExactCI` *Exact Unconditional Confidence Interval for Risk Difference*

Description

Obtains the exact unconditional confidence interval for risk difference based on the standardized score statistic.

Usage

```
riskDiffExactCI(
  n1 = NA_integer_,
  y1 = NA_integer_,
  n2 = NA_integer_,
  y2 = NA_integer_,
  cilevel = 0.95
)
```

Arguments

`n1`	The sample size for the active treatment group.
`y1`	The number of responses for the active treatment group.
`n2`	The sample size for the control group.
`y2`	The number of responses for the control group.
`cilevel`	The confidence interval level.

Value

A data frame containing the following variables:

- `scale`: The scale of treatment effect.
- `estimate`: The point estimate.
- `lower`: The lower limit of the confidence interval.
- `upper`: The upper limit of the confidence interval.
- `cilevel`: The confidence interval level.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
riskDiffExactCI(n1 = 30, y1 = 2, n2 = 30, y2 = 1, cilevel = 0.95)
```

`riskDiffExactPValue` *P-Value for Exact Unconditional Test of Risk Difference*

Description

Obtains the p-value for exact unconditional test of risk difference.

Usage

```
riskDiffExactPValue(
  riskDiffH0 = 0,
  directionUpper = 1L,
  n1 = NA_integer_,
  y1 = NA_integer_,
  n2 = NA_integer_,
  y2 = NA_integer_
)
```

Arguments

`riskDiffH0`	The risk difference under the null hypothesis. Defaults to 0.
`directionUpper`	Whether larger values represent better responses.
`n1`	The sample size for the active treatment group.
`y1`	The number of responses for the active treatment group.
`n2`	The sample size for the control group.
`y2`	The number of responses for the control group.

Value

A data frame containing the following variables:

- `riskDiffH0`: The risk difference under the null hypothesis.
- `directionUpper`: Whether larger values represent better responses.
- `riskDiff`: The observed risk difference.
- `zstat`: The observed value of the Z test statistic.
- `pvalue`: The one-sided p-value for the unconditional exact test.
- `pi2star`: The value of pi2 that yields the p-value.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
riskDiffExactPValue(riskDiffH0 = 0, directionUpper = 1,
                    n1 = 68, y1 = 2, n2 = 65, y2 = 1)
```

`riskRatioExactCI` *Exact Unconditional Confidence Interval for Risk Ratio*

Description

Obtains the exact unconditional confidence interval for risk ratio based on the standardized score statistic.

Usage

```
riskRatioExactCI(
  n1 = NA_integer_,
  y1 = NA_integer_,
  n2 = NA_integer_,
  y2 = NA_integer_,
  cilevel = 0.95
)
```

Arguments

`n1`	The sample size for the active treatment group.
`y1`	The number of responses for the active treatment group.
`n2`	The sample size for the control group.
`y2`	The number of responses for the control group.
`cilevel`	The confidence interval level.

Value

A data frame containing the following variables:

- `scale`: The scale of treatment effect.
- `estimate`: The point estimate.
- `lower`: The lower limit of the confidence interval.
- `upper`: The upper limit of the confidence interval.
- `cilevel`: The confidence interval level.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
riskRatioExactCI(n1 = 30, y1 = 2, n2 = 30, y2 = 1, cilevel = 0.95)
```

`riskRatioExactPValue` *P-Value for Exact Unconditional Test of Risk Ratio*

Description

Obtains the p-value for exact unconditional test of risk ratio.

Usage

```
riskRatioExactPValue(
  riskRatioH0 = 1,
  directionUpper = 1L,
  n1 = NA_integer_,
  y1 = NA_integer_,
  n2 = NA_integer_,
  y2 = NA_integer_
)
```

Arguments

`riskRatioH0`	The risk ratio under the null hypothesis. Defaults to 1.
`directionUpper`	Whether larger values represent better responses.
`n1`	The sample size for the active treatment group.
`y1`	The number of responses for the active treatment group.
`n2`	The sample size for the control group.
`y2`	The number of responses for the control group.

Value

A data frame containing the following variables:

- `riskRatioH0`: The risk ratio under the null hypothesis.
- `directionUpper`: Whether larger values represent better responses.
- `riskRatio`: The observed risk ratio.
- `zstat`: The observed value of the Z test statistic.
- `pvalue`: The one-sided p-value for the unconditional exact test.
- `pi2star`: The value of pi2 that yields the p-value.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
riskRatioExactPValue(riskRatioH0 = 1, directionUpper = 1,
                     n1 = 68, y1 = 2, n2 = 65, y2 = 1)
```

`rmdiff` *Estimate of Restricted Mean Survival Time Difference*

Description

Obtains the estimate of restricted mean survival time difference between two treatment groups.

Usage

```
rmdiff(
  data,
  rep = "",
  stratum = "",
  treat = "treat",
  time = "time",
  event = "event",
```

```
  milestone = NA_real_,
  rmstDiffH0 = 0,
  conflev = 0.95,
  biascorrection = 0L
)
```

Arguments

`data`	The input data frame that contains the following variables: • `rep`: The replication for by-group processing. • `stratum`: The stratum. • `treat`: The treatment. • `time`: The possibly right-censored survival time. • `event`: The event indicator.
`rep`	The name of the replication variable in the input data.
`stratum`	The name of the stratum variable in the input data.
`treat`	The name of the treatment variable 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.
`milestone`	The milestone time at which to calculate the restricted mean survival time.
`rmstDiffH0`	The difference in restricted mean survival times under the null hypothesis. Defaults to 0 for superiority test.
`conflev`	The level of the two-sided confidence interval for the difference in restricted mean survival times. Defaults to 0.95.
`biascorrection`	Whether to apply bias correction for the variance estimate of individual restricted mean survival times. Defaults to no bias correction.

Value

A data frame with the following variables:

- `rep`: The replication number.
- `milestone`: The milestone time relative to randomization.
- `rmstDiffH0`: The difference in restricted mean survival times under the null hypothesis.
- `rmst1`: The estimated restricted mean survival time for the treatment group.
- `rmst2`: The estimated restricted mean survival time for the control group.
- `rmstDiff`: The estimated difference in restricted mean survival times.
- `vrmst1`: The variance for rmst1.
- `vrmst2`: The variance for rmst2.
- `vrmstDiff`: The variance for rmstDiff.
- `rmstDiffZ`: The Z-statistic value.
- `rmstDiffPValue`: The one-sided p-value.

- `lower`: The lower bound of confidence interval.
- `upper`: The upper bound of confidence interval.
- `conflev`: The level of confidence interval.
- `biascorrection`: Whether to apply bias correction for the variance estimate of individual restricted mean survival times.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
df <- rmdiff(data = rawdata, rep = "iterationNumber",
             stratum = "stratum", treat = "treatmentGroup",
             time = "timeUnderObservation", event = "event",
             milestone = 12)
head(df)
```

`rmest` *Estimate of Restricted Mean Survival Time*

Description

Obtains the estimate of restricted means survival time for each stratum.

Usage

```
rmest(
  data,
  rep = "",
  stratum = "",
  time = "time",
  event = "event",
  milestone = NA_real_,
  conflev = 0.95,
  biascorrection = 0L
)
```

Arguments

`data`	The input data frame that contains the following variables: • `rep`: The replication for by-group processing. • `stratum`: The stratum. • `time`: The possibly right-censored survival time. • `event`: The event indicator.

`rep`	The name of the replication variable in the input data.
`stratum`	The name of the stratum variable 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.
`milestone`	The milestone time at which to calculate the restricted mean survival time.
`conflev`	The level of the two-sided confidence interval for the survival probabilities. Defaults to 0.95.
`biascorrection`	Whether to apply bias correction for the variance estimate. Defaults to no bias correction.

Value

A data frame with the following variables:

- `rep`: The replication.
- `stratum`: The stratum variable.
- `size`: The number of subjects in the stratum.
- `milestone`: The milestone time relative to randomization.
- `rmst`: The estimate of restricted mean survival time.
- `stderr`: The standard error of the estimated rmst.
- `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.
- `biascorrection`: Whether to apply bias correction for the variance estimate.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
rmest(data = aml, stratum = "x",
      time = "time", event = "status", milestone = 24)
```

`rmpower` *Power for Difference in Restricted Mean Survival Times*

Description

Estimates the power for testing the difference in restricted mean survival times in a two-sample survival design.

Usage

```
rmpower(
  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_,
  milestone = NA_real_,
  rmstDiffH0 = 0,
  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,
  spendingTime = NA_real_,
  studyDuration = NA_real_
)
```

Arguments

`kMax`	The maximum number of stages.
`informationRates`	The information rates. 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".
`milestone`	The milestone time at which to calculate the restricted mean survival time.
`rmstDiffH0`	The difference in restricted mean survival times under the null hypothesis. Defaults to 0 for superiority test.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.

`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.
`studyDuration`	Study duration for fixed follow-up design. Defaults to missing, which is to be replaced with the sum of `accrualDuration` and `followupTime`. If provided, the value is allowed to be less than the sum of `accrualDuration` and `followupTime`.

Value

An S3 class `rmpower` 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.
    - `numbeOfSubjects`: The total number of subjects.
    - `studyDuration`: The total study duration.
    - `information`: The maximum information.
    - `expectedNumberOfEvents`: The expected number of events.
    - `expectedNumberOfSubjects`: The expected number of subjects.
    - `expectedStudyDuration`: The expected study duration.
    - `expectedInformation`: The expected information.
    - `accrualDuration`: The accrual duration.
    - `followupTime`: The follow-up duration.
    - `fixedFollowup`: Whether a fixed follow-up design is used.
    - `kMax`: The number of stages.
    - `milestone`: The milestone time relative to randomization.

  - `rmstDiffH0`: The difference in restricted mean survival times under the null hypothesis.
  - `rmst1`: The restricted mean survival time for the treatment group.
  - `rmst2`: The restricted mean survival time for the control group.
  - `rmstDiff`: The difference in restricted mean survival times, equal to `rmst1 - rmst2`.
- `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.
  - `numberOfMilestone`: The number of subjects reaching milestone.
  - `analysisTime`: The average time since trial start.
  - `efficacyRmstDiff`: The efficacy boundaries on the restricted mean survival time difference scale.
  - `futilityRmstDiff`: The futility boundaries on the restricted mean survival time difference scale.
  - `efficacyP`: The efficacy boundaries on the p-value scale.
  - `futilityP`: The futility boundaries on the p-value scale.
  - `information`: The cumulative information.
  - `efficacyStopping`: Whether to allow efficacy stopping.
  - `futilityStopping`: Whether to allow futility stopping.
- `settings`: A list containing the following input parameters: `typeAlphaSpending`, `parameterAlphaSpending`, `userAlphaSpending`, `typeBetaSpending`, `parameterBetaSpending`, `allocationRatioPlanned`, `accrualTime`, `accuralIntensity`, `piecewiseSurvivalTime`, `stratumFraction`, `lambda1`, `lambda2`, `gamma1`, `gamma2`, and `spendingTime`.
- `byTreatmentCounts`: A list containing the following counts by treatment group:
  - `numberOfEvents1`: The number of events by stage for the treatment group.
  - `numberOfDropouts1`: The number of dropouts by stage for the treatment group.
  - `numberOfSubjects1`: The number of subjects by stage for the treatment group.
  - `numberOfMilestone1`: The number of subjects reaching milestone by stage for the active 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.
  - `numberOfMilestone2`: The number of subjects reaching milestone by stage for the control group.

- `expectedNumberOfEvents1`: The expected number of events for the treatment group.
- `expectedNumberOfDropouts1`: The expected number of dropouts for the active treatment group.
- `expectedNumberOfSubjects1`: The expected number of subjects for the active treatment group.
- `expectedNumberOfMilestone1`: The expected number of subjects reaching milestone for the active 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.
- `expectedNumberOfMilestone2`: The expected number of subjects reaching milestone 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.

rmpower(kMax = 2, informationRates = c(0.8, 1),
        alpha = 0.025, typeAlphaSpending = "sfOF",
        milestone = 18,
        allocationRatioPlanned = 1, accrualTime = seq(0, 8),
        accrualIntensity = 100/9*seq(1, 9),
        piecewiseSurvivalTime = c(0, 6),
        stratumFraction = c(0.2, 0.8),
        lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
        lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
        gamma1 = -log(1-0.05)/12,
        gamma2 = -log(1-0.05)/12, accrualDuration = 22,
        followupTime = 18, fixedFollowup = FALSE)
```

`rmpower1s` *Power for One-Sample Restricted Mean Survival Time*

Description

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

Usage

```
rmpower1s(
  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_,
  milestone = NA_real_,
  rmstH0 = NA_real_,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  lambda = NA_real_,
  gamma = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  spendingTime = NA_real_,
  studyDuration = NA_real_
)
```

Arguments

`kMax`	The maximum number of stages.
`informationRates`	The information rates. 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".
`milestone`	The milestone time at which to calculate the restricted mean survival time.
`rmstH0`	The restricted mean survival time under the null hypothesis.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda`	A vector of hazard rates for the event in each analysis time interval by stratum under the alternative hypothesis.
`gamma`	The hazard rate for exponential dropout or a vector of hazard rates for piecewise exponential dropout. Defaults to 0 for no dropout.
`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.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.
`studyDuration`	Study duration for fixed follow-up design. Defaults to missing, which is to be replaced with the sum of `accrualDuration` and `followupTime`. If provided, the value is allowed to be less than the sum of `accrualDuration` and `followupTime`.

Value

An S3 class `rmpower1s` object with 3 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.
    - `numbeOfSubjects`: The total number of subjects.
    - `numberOfMilestone`: The total number of subjects reaching milestone.
    - `studyDuration`: The total study duration.
    - `information`: The maximum information.
    - `expectedNumberOfEvents`: The expected number of events.
    - `expectedNumberOfSubjects`: The expected number of subjects.
    - `expectedNumberOfMilestone`: The expected number of subjects reaching milestone.
    - `expectedStudyDuration`: The expected study duration.
    - `expectedInformation`: The expected information.
    - `accrualDuration`: The accrual duration.
    - `followupTime`: The follow-up duration.
    - `fixedFollowup`: Whether a fixed follow-up design is used.
    - `kMax`: The number of stages.
    - `milestone`: The milestone time to calculate the restricted mean survival time.
    - `rmstH0`: The restricted mean survival time under the null hypothesis.
    - `rmst`: The restricted mean survival time under the alternative hypothesis.
- `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.
    - `numberOfMilestone`: The number of subjects reaching milestone.
    - `analysisTime`: The average time since trial start.
    - `efficacyRmst`: The efficacy boundaries on the restricted mean survival time.
    - `futilityRmst`: The futility boundaries on the restricted mean survival time.
    - `efficacyP`: The efficacy boundaries on the p-value scale.
    - `futilityP`: The futility boundaries on the p-value scale.
    - `information`: The cumulative information.

  - `efficacyStopping`: Whether to allow efficacy stopping.
  - `futilityStopping`: Whether to allow futility stopping.
- `settings`: A list containing the following input parameters: `typeAlphaSpending`, `parameterAlphaSpending`, `userAlphaSpending`, `typeBetaSpending`, `parameterBetaSpending`, `accrualTime`, `accuralIntensity`, `piecewiseSurvivalTime`, `stratumFraction`, `lambda`, `gamma`, and `spendingTime`.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

See Also

`rmstat`

Examples

```
rmpower1s(kMax = 2, informationRates = c(0.8, 1),
          alpha = 0.025, typeAlphaSpending = "sfOF",
          milestone = 18, rmstH0 = 10,
          accrualTime = seq(0, 8),
          accrualIntensity = 26/9*seq(1, 9),
          piecewiseSurvivalTime = c(0, 6),
          stratumFraction = c(0.2, 0.8),
          lambda = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
          gamma = -log(1-0.05)/12, accrualDuration = 22,
          followupTime = 18, fixedFollowup = FALSE)
```

`rmpowerequiv` *Power for Equivalence in Restricted Mean Survival Time Difference*

Description

Obtains the power for equivalence in restricted mean survival time difference.

Usage

```
rmpowerequiv(
  kMax = 1L,
  informationRates = NA_real_,
  criticalValues = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  milestone = NA_real_,
  rmstDiffLower = NA_real_,
  rmstDiffUpper = 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,
  spendingTime = NA_real_,
  studyDuration = NA_real_
)
```

Arguments

`kMax`	The maximum number of stages.
`informationRates`	The information rates. Defaults to `(1:kMax) / kMax` if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`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.
`milestone`	The milestone time at which to calculate the restricted mean survival time.
`rmstDiffLower`	The lower equivalence limit of restricted mean survival time difference.
`rmstDiffUpper`	The upper equivalence limit of restricted mean survival time difference.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.

`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`spendingTime`	A vector of length `kMax` for the error spending time at each analysis. Defaults to missing, in which case, it is the same as `informationRates`.
`studyDuration`	Study duration for fixed follow-up design. Defaults to missing, which is to be replaced with the sum of `accrualDuration` and `followupTime`. If provided, the value is allowed to be less than the sum of `accrualDuration` and `followupTime`.

Value

An S3 class `rmpowerequiv` 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.
  - `numberOfSubjects`: The total number of subjects.
  - `studyDuration`: The total study duration.
  - `information`: The maximum information.
  - `expectedNumberOfEvents`: The expected number of events.
  - `expectedNumberOfSubjects`: The expected number of subjects.
  - `expectedStudyDuration`: The expected study duration.
  - `expectedInformation`: The expected information.
  - `kMax`: The number of stages.

  - `milestone`: The milestone time relative to randomization.
  - `rmstDiffLower`: The lower equivalence limit of restricted mean survival time difference.
  - `rmstDiffUpper`: The upper equivalence limit of restricted mean survival time difference.
  - `rmst1`: The restricted mean survival time for the treatment group.
  - `rmst2`: The restricted mean survival time for the control group.
  - `rmstDiff`: The restricted mean survival time difference.
  - `accrualDuration`: The accrual duration.
  - `followupTime`: The follow-up duration.
  - `fixedFollowup`: Whether a fixed follow-up design is used.
- `byStageResults`: A data frame containing the following variables:
  - `informationRates`: The information rates.
  - `efficacyBounds`: The efficacy boundaries on the Z-scale for each of the two one-sided tests.
  - `rejectPerStage`: The probability for efficacy stopping.
  - `cumulativeRejection`: The cumulative probability for efficacy stopping.
  - `cumulativeAlphaSpent`: The cumulative alpha for each of the two one-sided tests.
  - `cumulativeAttainedAlphaH10`: The cumulative alpha attained under `H10`.
  - `cumulativeAttainedAlphaH20`: The cumulative alpha attained under `H20`.
  - `numberOfEvents`: The number of events.
  - `numberOfDropouts`: The number of dropouts.
  - `numberOfSubjects`: The number of subjects.
  - `numberOfMilestone`: The number of subjects reaching milestone.
  - `analysisTime`: The average time since trial start.
  - `efficacyRmstDiffLower`: The efficacy boundaries on the restricted mean survival time difference scale for the one-sided null hypothesis at the lower equivalence limit.
  - `efficacyRmstDiffUpper`: The efficacy boundaries on the restricted mean survival time difference scale for the one-sided null hypothesis at the upper equivalence limit.
  - `efficacyP`: The efficacy bounds on the p-value scale for each of the two one-sided tests.
  - `information`: The cumulative information.
- `settings`: A list containing the following input parameters: `typeAlphaSpending`, `parameterAlphaSpending`, `userAlphaSpending`, `allocationRatioPlanned`, `accrualTime`, `accuralIntensity`, `piecewiseSurvivalTime`, `stratumFraction`, `lambda1`, `lambda2`, `gamma1`, `gamma2`, and `spendingTime`.
- `byTreatmentCounts`: A list containing the following counts by treatment group:
  - `numberOfEvents1`: The number of events by stage for the treatment group.
  - `numberOfDropouts1`: The number of dropouts by stage for the treatment group.
  - `numberOfSubjects1`: The number of subjects by stage for the treatment group.
  - `numberOfMilestone1`: The number of subjects reaching milestone by stage for the active 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.
  - `numberOfMilestone2`: The number of subjects reaching milestone by stage for the control group.

  - `expectedNumberOfEvents1`: The expected number of events for the treatment group.
  - `expectedNumberOfDropouts1`: The expected number of dropouts for the active treatment group.
  - `expectedNumberOfSubjects1`: The expected number of subjects for the active treatment group.
  - `expectedNumberOfMilestone1`: The expected number of subjects reaching milestone for the active 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.
  - `expectedNumberOfMilestone2`: The expected number of subjects reaching milestone for the control group.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

See Also

`rmstat`

Examples

```
rmpowerequiv(kMax = 2, informationRates = c(0.5, 1),
             alpha = 0.05, typeAlphaSpending = "sfOF",
             milestone = 18,
             rmstDiffLower = -2, rmstDiffUpper = 2,
             allocationRatioPlanned = 1, accrualTime = seq(0, 8),
             accrualIntensity = 100/9*seq(1, 9),
             piecewiseSurvivalTime = c(0, 6),
             stratumFraction = c(0.2, 0.8),
             lambda1 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
             lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
             gamma1 = -log(1-0.05)/12,
             gamma2 = -log(1-0.05)/12, accrualDuration = 22,
             followupTime = 18, fixedFollowup = FALSE)
```

`rmsamplesize` *Sample Size for Difference in Restricted Mean Survival Times*

Description

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

Usage

```
rmsamplesize(
  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_,
  milestone = NA_real_,
  rmstDiffH0 = 0,
  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,
  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. 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.
`milestone`	The milestone time at which to calculate the restricted mean survival time.
`rmstDiffH0`	The difference in restricted mean survival times under the null hypothesis. Defaults to 0 for superiority test.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.

`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`interval`	The interval to search for the solution of accrualDuration, followupTime, or the proportionality constant of accrualIntensity. Defaults to `c(0.001, 240)`.
`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. Defaults to 1 for sample size rounding.

Value

A list of two components:

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

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

See Also

`rmpower`

Examples

```
# Example 1: Obtains follow-up time given power, accrual intensity,
# and accrual duration for variable follow-up. Of note, the power
# reaches the maximum when the follow-up time equals milestone.

rmsamplesize(beta = 0.2, kMax = 2, informationRates = c(0.8, 1),
             alpha = 0.025, typeAlphaSpending = "sfOF",
             milestone = 18,
             allocationRatioPlanned = 1, accrualTime = seq(0, 8),
             accrualIntensity = 100/9*seq(1, 9),
             piecewiseSurvivalTime = c(0, 6),
```

```
             stratumFraction = c(0.2, 0.8),
             lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
             lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
             gamma1 = -log(1-0.05)/12,
             gamma2 = -log(1-0.05)/12, accrualDuration = 22,
             followupTime = NA, fixedFollowup = FALSE)

# Example 2: Obtains accrual intensity given power, accrual duration, and
# follow-up time for variable follow-up

rmsamplesize(beta = 0.2, kMax = 2, informationRates = c(0.8, 1),
             alpha = 0.025, typeAlphaSpending = "sfOF",
             milestone = 18,
             allocationRatioPlanned = 1, accrualTime = seq(0, 8),
             accrualIntensity = 100/9*seq(1, 9),
             piecewiseSurvivalTime = c(0, 6),
             stratumFraction = c(0.2, 0.8),
             lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
             lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
             gamma1 = -log(1-0.05)/12,
             gamma2 = -log(1-0.05)/12, accrualDuration = 22,
             followupTime = 18, fixedFollowup = FALSE)


# Example 3: Obtains accrual duration given power, accrual intensity, and
# follow-up time for fixed follow-up

rmsamplesize(beta = 0.2, kMax = 2, informationRates = c(0.8, 1),
             alpha = 0.025, typeAlphaSpending = "sfOF",
             milestone = 18,
             allocationRatioPlanned = 1, accrualTime = seq(0, 8),
             accrualIntensity = 100/9*seq(1, 9),
             piecewiseSurvivalTime = c(0, 6),
             stratumFraction = c(0.2, 0.8),
             lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
             lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
             gamma1 = -log(1-0.05)/12,
             gamma2 = -log(1-0.05)/12, accrualDuration = NA,
             followupTime = 18, fixedFollowup = TRUE)
```

`rmsamplesize1s` *Sample Size for One-Sample Restricted Mean Survival Time*

Description

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

Usage

```
rmsamplesize1s(
  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_,
  milestone = NA_real_,
  rmstH0 = NA_real_,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  piecewiseSurvivalTime = 0L,
  stratumFraction = 1L,
  lambda = NA_real_,
  gamma = 0L,
  accrualDuration = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  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. 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.
`milestone`	The milestone time at which to calculate the restricted survival time.
`rmstH0`	The restricted mean survival time under the null hypothesis.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda`	A vector of hazard rates for the event in each analysis time interval by stratum under the alternative hypothesis.
`gamma`	The hazard rate for exponential dropout or a vector of hazard rates for piecewise exponential dropout. Defaults to 0 for no dropout.
`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.
`interval`	The interval to search for the solution of accrualDuration, followupDuration, or the proportionality constant of accrualIntensity. Defaults to `c(0.001, 240)`.
`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. Defaults to 1 for sample size rounding.

Value

A list of two components:

- `resultsUnderH1`: An S3 class `rmpower1s` object under the alternative hypothesis.
- `resultsUnderH0`: An S3 class `rmpower1s` object under the null hypothesis.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

See Also

`rmpower1s`

Examples

```
# Example 1: Obtains follow-up duration given power, accrual intensity,
# and accrual duration for variable follow-up

rmsamplesize1s(beta = 0.2, kMax = 2,
               informationRates = c(0.8, 1),
               alpha = 0.025, typeAlphaSpending = "sfOF",
               milestone = 18, rmstH0 = 10,
               accrualTime = seq(0, 8),
               accrualIntensity = 26/9*seq(1, 9),
               piecewiseSurvivalTime = c(0, 6),
               stratumFraction = c(0.2, 0.8),
               lambda = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
               gamma = -log(1-0.05)/12, accrualDuration = 22,
               followupTime = NA, fixedFollowup = FALSE)

# Example 2: Obtains accrual intensity given power, accrual duration, and
# follow-up duration for variable follow-up

rmsamplesize1s(beta = 0.2, kMax = 2,
               informationRates = c(0.8, 1),
               alpha = 0.025, typeAlphaSpending = "sfOF",
               milestone = 18, rmstH0 = 10,
               accrualTime = seq(0, 8),
               accrualIntensity = 26/9*seq(1, 9),
               piecewiseSurvivalTime = c(0, 6),
```

```
               stratumFraction = c(0.2, 0.8),
               lambda = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
               gamma = -log(1-0.05)/12, accrualDuration = 22,
               followupTime = 18, fixedFollowup = FALSE)


# Example 3: Obtains accrual duration given power, accrual intensity, and
# follow-up duration for fixed follow-up

rmsamplesize1s(beta = 0.2, kMax = 2,
               informationRates = c(0.8, 1),
               alpha = 0.025, typeAlphaSpending = "sfOF",
               milestone = 18, rmstH0 = 10,
               accrualTime = seq(0, 8),
               accrualIntensity = 26/9*seq(1, 9),
               piecewiseSurvivalTime = c(0, 6),
               stratumFraction = c(0.2, 0.8),
               lambda = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
               gamma = -log(1-0.05)/12, accrualDuration = NA,
               followupTime = 18, fixedFollowup = TRUE)
```

`rmsamplesizeequiv` *Sample Size for Equivalence in Restricted Mean Survival Time Difference*

Description

Obtains the sample size for equivalence in restricted mean survival time difference.

Usage

```
rmsamplesizeequiv(
  beta = 0.2,
  kMax = 1L,
  informationRates = NA_real_,
  criticalValues = NA_real_,
  alpha = 0.05,
  typeAlphaSpending = "sfOF",
  parameterAlphaSpending = NA_real_,
  userAlphaSpending = NA_real_,
  milestone = NA_real_,
  rmstDiffLower = NA_real_,
  rmstDiffUpper = 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,
  interval = as.numeric(c(0.001, 240)),
  spendingTime = NA_real_,
  rounding = 1L
)
```

Arguments

`beta`	The type II error.
`kMax`	The maximum number of stages.
`informationRates`	The information rates. Defaults to `(1:kMax) / kMax` if left unspecified.
`criticalValues`	Upper boundaries on the z-test statistic scale for stopping for efficacy.
`alpha`	The significance level for each of the two one-sided tests. Defaults to 0.05.
`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.
`milestone`	The milestone time at which to calculate the restricted mean survival time.
`rmstDiffLower`	The lower equivalence limit of restricted mean survival time difference.
`rmstDiffUpper`	The upper equivalence limit of restricted mean survival time difference.
`allocationRatioPlanned`	Allocation ratio for the active treatment versus control. Defaults to 1 for equal randomization.
`accrualTime`	A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).
`accrualIntensity`	A vector of accrual intensities. One for each accrual time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.

`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`interval`	The interval to search for the solution of accrualDuration, followupDuration, or the proportionality constant of accrualIntensity. Defaults to `c(0.001, 240)`.
`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. Defaults to 1 for sample size rounding.

Value

An S3 class `rmpowerequiv` object

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

See Also

`rmpowerequiv`

Examples

```
rmsamplesizeequiv(beta = 0.1, kMax = 2, informationRates = c(0.5, 1),
                  alpha = 0.05, typeAlphaSpending = "sfOF",
                  milestone = 18,
                  rmstDiffLower = -2, rmstDiffUpper = 2,
                  allocationRatioPlanned = 1, accrualTime = seq(0, 8),
                  accrualIntensity = 26/9*seq(1, 9),
                  piecewiseSurvivalTime = c(0, 6),
                  stratumFraction = c(0.2, 0.8),
                  lambda1 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
                  lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
```

```
                    gamma1 = -log(1-0.05)/12,
                    gamma2 = -log(1-0.05)/12, accrualDuration = NA,
                    followupTime = 18, fixedFollowup = FALSE)
```

`rmst` *Restricted Mean Survival Time*

Description

Obtains the restricted mean survival time over an interval.

Usage

```
rmst(t1 = 0, t2 = NA_real_, piecewiseSurvivalTime = 0L, lambda = NA_real_)
```

Arguments

`t1`	Lower bound of the analysis time interval.
`t2`	Upper bound of the analysis time interval.
`piecewiseSurvivalTime`	A vector that specifies the starting time of piecewise exponential survival time intervals. Must start with 0, e.g., `c(0, 6)` breaks the time axis into 2 event intervals: [0, 6) and [6, Inf). Defaults to 0 for exponential distribution.
`lambda`	A vector of hazard rates for the event. One for each analysis time interval.

Value

The integral of the survival function from `t1` to `t2`

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
rmst(t1 = 0, t2 = 7, piecewiseSurvivalTime = c(0, 6),
     lambda = c(0.0533, 0.0309))
```

rmstat *Stratified Difference in Restricted Mean Survival Times*

Description

Obtains the stratified restricted mean survival times and difference in restricted mean survival times at given calendar times.

Usage

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

Arguments

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

`lambda1`	A vector of hazard rates for the event in each analysis time interval by stratum for the active treatment group.
`lambda2`	A vector of hazard rates for the event in each analysis time interval by stratum for the control group.
`gamma1`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the active treatment group.
`gamma2`	The hazard rate for exponential dropout, a vector of hazard rates for piecewise exponential dropout applicable for all strata, or a vector of hazard rates for dropout in each analysis time interval by stratum for the control group.
`accrualDuration`	Duration of the enrollment period.
`followupTime`	Follow-up time for the last enrolled subject.
`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.

Value

A data frame containing the following variables:

- `time`: The calendar 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.
- `milestone`: The milestone time relative to randomization.
- `nmilestone`: The total number of subjects reaching milestone.
- `nmilestone1`: The number of subjects reaching milestone in the active treatment group.
- `nmiletone2`: The number of subjects reaching milestone in the control group.
- `rmst1`: The restricted mean survival time for the treatment group.
- `rmst2`: The restricted mean survival time for the control group.
- `rmstDiff`: The difference in restricted mean survival times, i.e., `rmst1 - rmst2`.
- `vrmst1`: The variance for `rmst1`.
- `vrmst2`: The variance for `rmst2`.
- `vrmstDiff`: The variance for `rmstDiff`.
- `information`: The information for `rmstDiff`, equal to `1/vrmstDiff`.
- `rmstDiffZ`: The Z-statistic value, i.e., `rmstDiff/sqrt(vrmstDiff)`.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

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

rmstat(time = c(22, 40),
       milestone = 18,
       allocationRatioPlanned = 1,
       accrualTime = seq(0, 8),
       accrualIntensity = 26/9*seq(1, 9),
       piecewiseSurvivalTime = c(0, 6),
       stratumFraction = c(0.2, 0.8),
       lambda1 = c(0.0533, 0.0309, 1.5*0.0533, 1.5*0.0309),
       lambda2 = c(0.0533, 0.0533, 1.5*0.0533, 1.5*0.0533),
       gamma1 = -log(1-0.05)/12,
       gamma2 = -log(1-0.05)/12,
       accrualDuration = 22,
       followupTime = 18, fixedFollowup = FALSE)
```

`rmvnorm` *Random Multivariate Normal Generation*

Description

Generates random samples from a multivariate normal distribution with a specified mean vector and covariance matrix.

Usage

```
rmvnorm(n, mean, sigma)
```

Arguments

`n`	The number of samples to generate.
`mean`	A numeric vector representing the mean of the distribution.
`sigma`	A numeric matrix representing the covariance matrix.

Details

This function generates samples from a multivariate normal distribution using the Cholesky decomposition method. It first computes the Cholesky factorization of the covariance matrix, then generates standard normal random variables, and finally transforms them to the desired multivariate normal distribution.

Value

A numeric matrix where each row represents a sample from the multivariate normal distribution.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
# Generate 5 samples from a bivariate normal distribution with mean (0,0)
# and covariance matrix [[1, 0.5], [0.5, 1]]

set.seed(314159)
rmvnorm(5, c(0, 0), matrix(c(1, 0.5, 0.5, 1), nrow=2))
```

`rtpwexp` *Random Number Generation Function of Truncated Piecewise Exponential Distribution*

Description

Obtains random samples from a truncated piecewise exponential distribution.

Usage

```
rtpwexp(n, piecewiseSurvivalTime = 0, lambda = 0.0578, lowerBound = 0)
```

Arguments

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

Value

The random numbers from truncated piecewise exponential distribution.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
rtpwexp(n = 10, piecewiseSurvivalTime = c(0, 6, 9, 15),
        lambda = c(0.025, 0.04, 0.015, 0.007))
```

`runShinyApp_lrstat` *Run Shiny App*

Description

Runs the log-rank test power and sample size calculation Shiny app.

Usage

```
runShinyApp_lrstat()
```

Value

No return value, called for side effects.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

`sexagg` *Urinary tract infection data from the logistf package*

Description

This data set deals with urinary tract infection in sexually active college women, along with covariate information on age an contraceptive use. The variables are all binary and coded in 1 (condition is present) and 0 (condition is absent).

Usage

```
sexagg
```

Format

An object of class `data.frame` with 36 rows and 9 columns.

Details

`case` urinary tract infection, the study outcome variable

`age` >= 24 years

`dia` use of diaphragm

`oc` use of oral contraceptive

`vic` use of condom

`vicl` use of lubricated condom

`vis` use of spermicide

`shilong` *The randomized clinical trial SHIVA data in long format from the ipcwswitch package*

Description

The original SHIdat data set contains an anonymized excerpt of data from the SHIVA01 trial. This was the first randomized clinical trial that aimed at comparing molecularly targeted therapy based on tumor profiling (MTA) versus conventional therapy (CT) for advanced cancer. Patients were randomly assigned to receive the active or control treatment and may switch to the other arm or subsequent anti-cancer therapy upon disease progression. The restructured data is in the long format.

`id` The patient's identifier

`tstart` The start of the time interval

`tstop` The end of the time interval

`event` Whether the patient died at the end of the interval

`agerand` The patient's age (in years) at randomization

`sex.f` The patients' gender, either Male or Female

`tt_Lnum` The number of previous lines of treatment

`rmh_alea.c` The Royal Marsden Hospital score segregated into two categories

`pathway.f` The molecular pathway altered (the hormone receptors pathway, the PI3K/ AKT/mTOR pathway, and the RAF/MEK pathway)

`bras.f` The patient's randomized arm, either MTA or CT

`ps` The ECOG performance status

`ttc` The presence of concomitant treatments

`tran` The use of platelet transfusions

`dpd` The relative day of a potential progression

`dco` The relative day of treatment switching

`ady` The relative day of the latest news

`dcut` The relative day of administrative cutoff

`pd` Whether the patient had disease progression

`co` Whether the patient switched treatment

Usage

```
shilong
```

Format

An object of class `data.frame` with 602 rows and 19 columns.

`simon2stage` *Simon's Two-Stage Design*

Description

Obtains Simon's two-stage minimax, admissible, and optimal designs.

Usage

```
simon2stage(
  alpha = NA_real_,
  beta = NA_real_,
  piH0 = NA_real_,
  pi = NA_real_,
  n_max = 110L
)
```

Arguments

`alpha`	Type I error rate (one-sided).
`beta`	Type II error rate (1-power).
`piH0`	Response probability under the null hypothesis.
`pi`	Response probability under the alternative hypothesis.
`n_max`	Upper limit for sample size, defaults to 110.

Value

A data frame containing the following variables:

- `piH0`: Response probability under the null hypothesis.
- `pi`: Response probability under the alternative hypothesis.
- `alpha`: The specified one-sided significance level.
- `beta`: The specified type II error.
- `n`: Total sample size.
- `n1`: Stage 1 sample size.
- `r1`: Futility boundary for stage 1.
- `r`: Futility boundary for stage 2.

- `EN0`: Expected sample size under the null hypothesis.
- `attainedAlpha`: Attained type 1 error.
- `power`: Attained power.
- `PET0`: Probability of early stopping under the null hypothesis.
- `w_lower`: Lower bound of the interval for `w`.
- `w_upper`: Upper bound of the interval for `w`.
- `design`: Description of the design, e.g., minimax, admissible, or optimal.

Here `w` is the weight in the objective function: `w*n + (1-w)*EN0`.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
simon2stage(0.05, 0.2, 0.1, 0.3)
```

`simonBayesAnalysis` *Analysis of Simon's Bayesian Basket Trials*

Description

Obtains the prior and posterior probabilities for Simon's Bayesian basket discovery trials.

Usage

```
simonBayesAnalysis(
  nstrata = NA_integer_,
  r = NA_real_,
  n = NA_real_,
  lambda = NA_real_,
  gamma = NA_real_,
  phi = NA_real_,
  plo = NA_real_
)
```

Arguments

`nstrata`	The number of strata.
`r`	The vector of number of responders across strata.
`n`	The vector of number of subjects across strata.
`lambda`	The prior probability that the drug activity is homogeneous across strata.
`gamma`	The prior probability that the drug is active in a stratum.
`phi`	The response probability for an active drug.
`plo`	The response probability for an inactive drug.

Value

A list containing the following five components:

- `case`: The matrix with each row corresponding to a combination of drug activity over strata represented by the columns.
- `prior_case`: The vector of joint prior probabilities for the stratum-specific response rates.
- `prior_stratum`: The vector of marginal prior probabilities for the stratum-specific response rates.
- `post_case`: The vector of joint posterior probabilities for the stratum-specific response rates.
- `post_stratum`: The vector of marginal posterior probabilities for the stratum-specific response rates.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
a = simonBayesAnalysis(
  nstrata = 10,
  r = c(8,0,1,1,6,2,0,0,3,3),
  n = c(19,10,26,8,14,7,8,5,4,14),
  lambda = 0.5, gamma = 0.33,
  phi = 0.35, plo = 0.15)

a$post_stratum
```

`simonBayesSim` *Simulation of Simon's Bayesian Basket Trials*

Description

Obtains the simulated raw and summary data for Simon's Bayesian basket discovery trials.

Usage

```
simonBayesSim(
  p = NA_real_,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  stratumFraction = 1L,
  lambda = NA_real_,
  gamma = NA_real_,
  phi = NA_real_,
  plo = NA_real_,
  T = NA_real_,
```

```
  maxSubjects = NA_integer_,
  plannedSubjects = NA_integer_,
  maxNumberOfIterations = 1000L,
  maxNumberOfRawDatasets = 1L,
  seed = NA_integer_
)
```

Arguments

`p`	The vector of true response probabilities across strata.
`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.
`stratumFraction`	A vector of stratum fractions that sum to 1. Defaults to 1 for no stratification.
`lambda`	The prior probability that the drug activity is homogeneous across strata.
`gamma`	The prior probability that the drug is active in a stratum.
`phi`	The response probability for an active drug.
`plo`	The response probability for an inactive drug.
`T`	The threshold for a conclusive posterior probability to stop enrollment.
`maxSubjects`	The maximum total sample size.
`plannedSubjects`	The planned cumulative number of subjects at each stage.
`maxNumberOfIterations`	The number of simulation iterations. Defaults to 1000.
`maxNumberOfRawDatasets`	The number of raw datasets to extract.
`seed`	The seed to reproduce the simulation results. The seed from the environment will be used if left unspecified,

Value

A list containing the following four components:

- `rawdata`: A data frame for subject-level data, containing the following variables:
  - `iterationNumber`: The iteration number.
  - `stageNumber`: The stage number.
  - `subjectId`: The subject ID.
  - `arrivalTime`: The enrollment time for the subject.
  - `stratum`: The stratum for the subject.
  - `y`: Whether the subject was a responder (1) or nonresponder (0).
- `sumdata1`: A data frame for simulation and stratum-level summary data, containing the following variables:

  - `iterationNumber`: The iteration number.
  - `stageNumber`: The stage number.
  - `stratum`: The stratum number.
  - `active`: Whether the drug is active in the stratum.
  - `n`: The number of subjects in the stratum.
  - `r`: The number of responders in the stratum.
  - `posterior`: The posterior probability that the drug is active in the stratum.
  - `open`: Whether the stratum is still open for enrollment.
  - `positive`: Whether the stratum has been determined to be a positive stratum.
  - `negative`: Whether the stratum has been determined to be a negative stratum.
- `sumdata2`: A data frame for the simulation level summary data, containing the following variables:
  - `iterationNumber`: The iteration number.
  - `numberOfStrata`: The total number of strata.
  - `n_active_strata`: The number of active strata.
  - `true_positive`: The number of true positive strata.
  - `false_negative`: The number of false negative strata.
  - `false_positive`: The number of false positive strata.
  - `true_negative`: The number of true negative strata.
  - `n_indet_strata`: The number of indeterminate strata.
  - `numberOfSubjects`: The number of subjects.
- `overview`: A data frame for the summary across simulations, containing the following variables:
  - `numberOfStrata`: The total number of strata.
  - `n_active_strata`: The average number of active strata.
  - `true_positive`: The average number of true positive strata.
  - `false_negative`: The average number of false negative strata.
  - `false_positive`: The average number of false positive strata.
  - `true_negative`: The average number of true negative strata.
  - `n_indet_strata`: The average number of indeterminate strata.
  - `numberOfSubjects`: The average number of subjects.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
sim1 = simonBayesSim(
  p = c(0.25, 0.25, 0.05),
  accrualIntensity = 5,
  stratumFraction = c(1/3, 1/3, 1/3),
  lambda = 0.33, gamma = 0.5,
  phi = 0.25, plo = 0.05,
  T = 0.8, maxSubjects = 50,
```

```
  plannedSubjects = seq(5, 50, 5),
  maxNumberOfIterations = 1000,
  maxNumberOfRawDatasets = 1,
  seed = 314159)

sim1$overview
```

`six` *The repeated measures data from the "Six Cities" study of the health effects of air pollution (Ware et al. 1984).*

Description

The data analyzed are the 16 selected cases in Lipsitz et al. (1994). The binary response is the wheezing status of 16 children at ages 9, 10, 11, and 12 years. A value of 1 of wheezing status indicates the occurrence of wheezing. The explanatory variables city of residence, age, and maternal smoking status at the particular age.

Usage

```
six
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 64 rows and 6 columns.

Details

`case` case id

`city` city of residence

`age` age of the child

`smoke` maternal smoking status

`wheeze` wheezing status

`survfit_phregr` *Survival Curve for Proportional Hazards Regression Models*

Description

Obtains the predicted survivor function for a proportional hazards regression model.

Usage

```
survfit_phregr(
  object,
  newdata,
  sefit = TRUE,
  conftype = "log-log",
  conflev = 0.95
)
```

Arguments

`object`	The output from the `phregr` call.
`newdata`	A data frame with the same variable names as those that appear in the `phregr` call. For right-censored data, one curve is produced per row to represent a cohort whose covariates correspond to the values in `newdata`. For counting-process data, one curve is produced per `id` in `newdata` to present the survival curve along the path of time-dependent covariates at the observed event times in the data used to fit `phregr`.
`sefit`	Whether to compute the standard error of the survival estimates.
`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 `asin(sqrt(surv))`.
`conflev`	The level of the two-sided confidence interval for the survival probabilities. Defaults to 0.95.

Details

If `newdata` is not provided and there is no covariate, survival curves based on the `basehaz` data frame will be produced.

Value

A data frame with the following variables:

- `id`: The id of the subject for counting-process data with time-dependent covariates.
- `time`: The observed times in the data used to fit `phregr`.
- `nrisk`: The number of patients at risk at the time point in the data used to fit `phregr`.
- `nevent`: The number of patients having event at the time point in the data used to fit `phregr`.

- `cumhaz`: The cumulative hazard at the time point.
- `surv`: The estimated survival probability at the time point.
- `sesurv`: The standard error of the estimated survival probability.
- `lower`: The lower confidence limit for survival probability.
- `upper`: The upper confidence limit for survival probability.
- `conflev`: The level of the two-sided confidence interval.
- `conftype`: The type of the confidence interval.
- `covariates`: The values of covariates based on `newdata`.
- `stratum`: The stratum of the subject.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

Terry M. Therneau and Patricia M. Grambsch. Modeling Survival Data: Extending the Cox Model. Springer-Verlag, 2000.

Examples

```
library(dplyr)

# Example 1 with right-censored data
fit1 <- phregr(data = rawdata %>% filter(iterationNumber == 1) %>%
                 mutate(treat = 1*(treatmentGroup == 1)),
               stratum = "stratum",
               time = "timeUnderObservation", event = "event",
               covariates = "treat")

surv1 <- survfit_phregr(fit1,
                        newdata = data.frame(
                          stratum = as.integer(c(1,1,2,2)),
                          treat = c(1,0,1,0)))

# Example 2 with counting process data and robust variance estimate
fit2 <- phregr(data = heart %>% mutate(rx = as.numeric(transplant) - 1),
               time = "start", time2 = "stop", event = "event",
               covariates = c("rx", "age"), id = "id", robust = TRUE)

surv2 <- survfit_phregr(fit2,
                        newdata = data.frame(
                          id = c(4,4,11,11),
                          age = c(-7.737,-7.737,-0.019,-0.019),
                          start = c(0,36,0,26),
                          stop = c(36,39,26,153),
                          rx = c(0,1,0,1)))
```

`survQuantile` *Brookmeyer-Crowley Confidence Interval for Quantiles of Right-Censored Time-to-Event Data*

Description

Obtains the Brookmeyer-Crowley confidence interval for quantiles of right-censored time-to-event data.

Usage

```
survQuantile(
  time = NA_real_,
  event = NA_real_,
  cilevel = 0.95,
  transform = "loglog",
  probs = NA_real_
)
```

Arguments

`time`	The vector of possibly right-censored survival times.
`event`	The vector of event indicators.
`cilevel`	The confidence interval level. Defaults to 0.95.
`transform`	The transformation of the survival function to use to construct the confidence interval. Options include "linear" (alternatively "plain"), "log", "loglog" (alternatively "log-log" or "cloglog"), "asinsqrt" (alternatively "asin" or "arcsin"), and "logit". Defaults to "loglog".
`probs`	The vector of probabilities to calculate the quantiles. Defaults to c(0.25, 0.5, 0.75).

Value

A data frame containing the estimated quantile and confidence interval corresponding to each specified probability. It includes the following variables:

- `prob`: The probability to calculate the quantile.
- `quantile`: The estimated quantile.
- `lower`: The lower limit of the confidence interval.
- `upper`: The upper limit of the confidence interval.
- `cilevel`: The confidence interval level.
- `transform`: The transformation of the survival function to use to construct the confidence interval.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
survQuantile(
  time = c(33.7, 3.9, 10.5, 5.4, 19.5, 23.8, 7.9, 16.9, 16.6,
           33.7, 17.1, 7.9, 10.5, 38),
  event = c(0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1),
  probs = c(0.25, 0.5, 0.75))
```

`svdcpp` *Singular Value Decomposition of a Matrix*

Description

Computes the singular-value decomposition of a rectangular matrix.

Usage

```
svdcpp(X, outtransform = 1L, decreasing = 1L)
```

Arguments

`X`	A numeric matrix whose SVD decomposition is to be computed.
`outtransform`	Whether the orthogonal matrices composing of the left and right singular vectors are to be computed.
`decreasing`	Whether the singular values should be sorted in decreasing order and the corresponding singular vectors rearranged accordingly.

Details

Given $A \in R^{m \times n} (m \geq n)$, the following algorithm overwrites A with $U^T AV = D$, where $U \in R^{m \times m}$ is orthogonal, $V \in R^{n \times n}$ is orthogonal, and $D \in R^{m \times n}$ is diagonal.

Value

A list with the following components:

- `d`: A vector containing the singular values of X.
- `U`: A matrix whose columns contain the left singular vectors of X.
- `V`: A matrix whose columns contain the right singular vectors of X.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

Gene N. Golub and Charles F. Van Loan. Matrix Computations, second edition. Baltimore, Maryland: The John Hopkins University Press, 1989, p.434.

Examples

```
A <- matrix(c(1,0,0,0, 1,2,0,0, 0,1,3,0, 0,0,1,4), 4, 4)
svdcpp(A)
```

`tobin` *Tobin's tobit data from the survival package*

Description

Data from Tobin's original paper.

`durable`	Durable goods purchase
`age`	Age in years
`quant`	Liquidity ratio (x 1000)

Usage

```
tobin
```

Format

An object of class `data.frame` with 20 rows and 3 columns.

`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)
```

`zstatOddsRatio` *Miettinen-Nurminen Score Test Statistic for Two-Sample Odds Ratio*

Description

Obtains the Miettinen-Nurminen score test statistic for two-sample odds ratio possibly with stratification.

Usage

```
zstatOddsRatio(
  oddsRatioH0 = 1,
  n1 = NA_real_,
  y1 = NA_real_,
  n2 = NA_real_,
  y2 = NA_real_
)
```

Arguments

`oddsRatioH0`	The odds ratio under the null hypothesis. Defaults to 1.
`n1`	The sample size for the active treatment group.
`y1`	The number of responses for the active treatment group.
`n2`	The sample size for the control group.
`y2`	The number of responses for the control group.

Details

The Mantel-Haenszel sample size weights are used for stratified samples.

Value

The value of the score test statistic.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
zstatOddsRatio(oddsRatioH0 = 1, n1 = c(10,10), y1 = c(4,3),
               n2 = c(20,10), y2 = c(2,0))
```

`zstatRateDiff` *Miettinen-Nurminen Score Test Statistic for Two-Sample Rate Difference*

Description

Obtains the Miettinen-Nurminen score test statistic for two-sample rate difference possibly with stratification.

Usage

```
zstatRateDiff(
  rateDiffH0 = 0,
  t1 = NA_real_,
  y1 = NA_real_,
  t2 = NA_real_,
  y2 = NA_real_
)
```

Arguments

`rateDiffH0`	The rate difference under the null hypothesis. Defaults to 0.
`t1`	The exposure for the active treatment group.
`y1`	The number of events for the active treatment group.
`t2`	The exposure for the control group.
`y2`	The number of events for the control group.

Details

The Mantel-Haenszel weights are used for stratified samples.

Value

The value of the score test statistic.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

Examples

```
zstatRateDiff(rateDiffH0 = 0, t1 = c(10,10), y1 = c(4,3),
              t2 = c(20,10), y2 = c(2,0))
```

---

zstatRateRatio *Miettinen-Nurminen Score Test Statistic for Two-Sample Rate Ratio*

Description

Obtains the Miettinen-Nurminen score test statistic for two-sample rate ratio possibly with stratification.

Usage

```
zstatRateRatio(
  rateRatioH0 = 1,
  t1 = NA_real_,
  y1 = NA_real_,
  t2 = NA_real_,
  y2 = NA_real_
)
```

Arguments

`rateRatioH0`	The rate ratio under the null hypothesis. Defaults to 1.
`t1`	The exposure for the active treatment group.
`y1`	The number of events for the active treatment group.
`t2`	The exposure for the control group.
`y2`	The number of events for the control group.

Details

The Mantel-Haenszel weights are used for stratified samples.

Value

The value of the score test statistic.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

Examples

```
zstatRateRatio(rateRatioH0 = 1, t1 = c(10,10), y1 = c(4,3),
               t2 = c(20,10), y2 = c(2,0))
```

`zstatRiskDiff` *Miettinen-Nurminen Score Test Statistic for Two-Sample Risk difference*

Description

Obtains the Miettinen-Nurminen score test statistic for two-sample risk difference possibly with stratification.

Usage

```
zstatRiskDiff(
  riskDiffH0 = 0,
  n1 = NA_real_,
  y1 = NA_real_,
  n2 = NA_real_,
  y2 = NA_real_
)
```

Arguments

`riskDiffH0`	The risk difference under the null hypothesis. Defaults to 0.
`n1`	The sample size for the active treatment group.
`y1`	The number of responses for the active treatment group.
`n2`	The sample size for the control group.
`y2`	The number of responses for the control group.

Details

The Mantel-Haenszel sample size weights are used for stratified samples.

Value

The value of the score test statistic.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
zstatRiskDiff(riskDiffH0 = 0, n1 = c(10,10), y1 = c(4,3),
              n2 = c(20,10), y2 = c(2,0))
```

`zstatRiskRatio` *Miettinen-Nurminen Score Test Statistic for Two-Sample Risk Ratio*

Description

Obtains the Miettinen-Nurminen score test statistic for two-sample risk ratio possibly with stratification.

Usage

```
zstatRiskRatio(
  riskRatioH0 = 1,
  n1 = NA_real_,
  y1 = NA_real_,
  n2 = NA_real_,
  y2 = NA_real_
)
```

Arguments

`riskRatioH0`	The risk ratio under the null hypothesis. Defaults to 1.
`n1`	The sample size for the active treatment group.
`y1`	The number of responses for the active treatment group.
`n2`	The sample size for the control group.
`y2`	The number of responses for the control group.

Details

The Mantel-Haenszel sample size weights are used for stratified samples.

Value

The value of the score test statistic.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
zstatRiskRatio(riskRatioH0 = 1, n1 = c(10,10), y1 = c(4,3),
               n2 = c(20,10), y2 = c(2,0))
```

Index