Efficacy and Futility Boundaries for Bayesian and Frequentist SPRT

Description

This function provides the efficacy and futility boundaries for the sequential ratio probability test under the frequentist and Bayesian setup.

Usage

```
SPRTboundary(method, n.event, n.start, n.increment, ve.null, ve.alt, type1,
power, ratio.alloc, optimum.w, w, a.prior, b.prior)
```

Arguments

`method`	Provide "Freq" for Frequentist approach; "Bayes" for Bayesian approach
`n.event`	Total number of events
`n.start`	Starting value of the event
`n.increment`	Differences between two interims, by default 1
`ve.null`	Vaccine efficacy under null, by default 0
`ve.alt`	Vaccine efficacy under alternative
`type1`	Specified Type I error, by default 0.025
`power`	Specified power, by default 0.9
`ratio.alloc`	Treatment vs control randomization = M:1, provide M; by default 1
`optimum.w`	TRUE if use optimum weight; FALSE if used provide the shape parameter a.prior; by default TRUE
`w`	Weight for calculating shape parameter of Beta distibution if method = "Bayes" and optimum.w = TRUE; ranges between 0 and 1; by default 0.5
`a.prior`	Shape parameter of Beta prior if method = "Bayes" and optimum.w = FALSE
`b.prior`	Scale parameter of Beta prior; by default 1

Value

`boundary`	Number of events and corresponding efficacy and futility boundaries
`method`	Provide "Freq" for Frequentist approach; "Bayes" for Bayesian approach
`n.event`	Total number of events
`n.start`	Starting value of the event
`n.increment`	Differences between two interims
`ve.null`	Vaccine efficacy under null
`ve.alt`	Vaccine efficacy under alternative
`type1`	Specified Type I error
`power`	Specified power, by default 0.9
`ratio.alloc`	Treatment vs control randomization = M:1, value of M
`optimum.w`	Optimum weight
`w`	Weight for calculating shape parameter of Beta distibution
`a.prior`	Shape parameter of Beta prior
`b.prior`	Scale parameter of Beta prior

Author(s)

Erina Paul <erina.paul@merck.com>

Examples

```
n.event = 60
n.start = 30
n.increment = 5
ve.null = 0.3
ve.alt = 0.75

# Bayesian
SPRTboundary(method = "Bayes", n.event = n.event, n.start = n.start,
             n.increment = n.increment, ve.null = ve.null, ve.alt = ve.alt)

# Frequentist
SPRTboundary(method = "Freq", n.event = n.event, n.start = n.start,
             n.increment = n.increment, ve.null = ve.null, ve.alt = ve.alt)
```

SPRT on Vaccine Efficacy

Description

This function provides the results for sequential ratio probability test under frequentist and Bayesian setup.

Usage

```
SPRTonVE(n.sim, boundary.val, ve.null, ve.alt, ve, ratio.alloc, seed)
```

Arguments

`n.sim`	Simulation size, by default 100,000
`boundary.val`	The total number of events, efficacy, and futility boundaries based on the interim for SPRT
`ve.null`	Vaccine efficacy under null, by default 0
`ve.alt`	Vaccine efficacy under alternative
`ve`	Vaccine efficacy based on the model, if under null, ve = ve.null; if under alternative, ve = ve.alt
`seed`	Seed value to simulate data from binomial, by default 134

Value

`boundary`	No of events and corresponding efficacy and futility boundaries
`cumulative.prob`	Cumulative probabilities (CP) under null and alternative - number of events, efficacy and futility boundaries, total CP, CP for reject null, CP for accept null under null (if ve = ve.nul) or alternative (if ve = ve.alt)
`type1.power`	Probability of rejecting null under null (if ve = ve.null) or alternative (if ve = ve.alt)
`prob.VE`	Probabilities that x <= efficacy and x >= futility boundaries-number of events, efficacy and futility boundaries under null (if ve = ve.null) or alternative (if ve = ve.alt)
`percent.stop`	Percentage of average sample sizes at which decision can be made under null and alternative
`summary.stat`	Summary statistics of events: min, Q1, Q2, mean, Q3, max, sd under null and alternative
`time`	Time to run the code under null and alternative

Author(s)

Erina Paul <erina.paul@merck.com>

Examples

```
method = "Bayes"
n.sim = 10000
n.event = 60
n.start = 30
n.increment = 5
ve.null =.3
ve.alt = 0.75
type1 = 0.025
power = 0.9
ratio.alloc = 1
optimum.w = TRUE
b.prior = 1
seed = 134

boundary.result = SPRTboundary(method = method, n.event = n.event,
                               n.start = n.start, n.increment = n.increment,
                               ve.null = ve.null, ve.alt = ve.alt,
                               type1 = type1, power = power,
                               ratio.alloc = ratio.alloc,
                               optimum.w = optimum.w, w = 0.5,
                               b.prior = b.prior)

# Under null hypothesis
SPRTonVE(n.sim = n.sim, boundary.val = boundary.result$boundary,
         ve.null = boundary.result$ve.null, ve.alt = boundary.result$ve.alt,
         ve = boundary.result$ve.null,
         ratio.alloc = boundary.result$ratio.alloc,
         seed = seed)
```

Optimum weight for Beta prior in Bayesian SPRT

Description

This function provides the optimum weight for calculating the shape parameter of the Beta prior in Bayesian SPRT.

Usage

```
SPRToptimumW(n.sim, n.event, n.start, n.increment, ve.null, ve.alt, type1,
power, ratio.alloc, w, b.prior, seed)
```

Arguments

`n.sim`	Simulation size, by default 100,000
`n.event`	Total number of events
`n.start`	Starting value of the event
`n.increment`	Differences between two interims, by default 1
`ve.null`	Vaccine efficacy under null, by default 0
`ve.alt`	Vaccine efficacy under alternative
`type1`	Specified Type I error, by default 0.025
`power`	Specified power, by default 0.9
`ratio.alloc`	Treatment vs control randomization = M:1, provide M; by default 1
`w`	Vector of weights for calculating the shape parameter of Beta prior; ranges between 0 and 1
`b.prior`	Scale parameter of Beta prior; by default 1
`seed`	Seed value to simulate data from binomial, by default 134

Value

`weight` Optimum weight to calculate the shape parameter of the Beta prior

Author(s)

Erina Paul <erina.paul@merck.com>

Examples

```
n.sim = 10000
n.event = 60
n.start = 30
n.increment = 5
ve.null =.3
ve.alt = 0.75
type1 = 0.025
power = 0.9
ratio.alloc = 1
w =  seq(0.1, 0.9, by = 0.1)
b.prior = 1
seed = 134
```

```
SPRToptimumW(n.sim = n.sim, n.event = n.event, n.start = n.start,
             n.increment = n.increment, ve.null = ve.null, ve.alt = ve.alt,
             type1 = type1, power = power, ratio.alloc = ratio.alloc, w = w,
             b.prior = b.prior, seed = seed)
```

Bayesian and Frequentist SPRT on Vaccine Efficacy

Description

This function provides the combined results for sequential ratio probability test under null and alternative hypothesis for both the frequentist and Bayesian setup.

Usage

```
outputSPRT(method, n.sim, n.event, n.start, n.increment, ve.null, ve.alt,
type1, power, ratio.alloc,  optimum.w, w, a.prior, b.prior, seed)
```

Arguments

`method`	Provide "Freq" for Frequentist approach; "Bayes" for Bayesian approach
`n.sim`	Simulation size, by default 100,000
`n.event`	Total number of events
`n.start`	Starting value of the event
`n.increment`	Differences between two interims, by default 1
`ve.null`	Vaccine efficacy under null, by default 0
`ve.alt`	Vaccine efficacy under alternative
`type1`	Specified Type I error, by default 0.025
`power`	Specified power, by default 0.9
`ratio.alloc`	Treatment vs control randomization = M:1, provide M; by default 1
`optimum.w`	TRUE if use optimum weight; FALSE if used provide the shape parameter a.prior
`w`	Weight for calculating shape parameter of Beta distibution if method = "Bayes" and optimum.w = TRUE; ranges between 0 and 1; by default 0.5
`a.prior`	Shape parameter of Beta prior if method = "Bayes" and optimum.w = FALSE
`b.prior`	Scale parameter of Beta prior; by default 1

seed	Seed value to simulate data from binomial, by default 134

Value

boundary	No of events and corresponding efficacy and futility boundaries
cumulative.prob	Cumulative probabilities (CP) under null and alternative - number of events, efficacy and futility boundaries, total CP, CP for reject null, CP for accept null under null (if ve = ve.nul) or alternative (if ve = ve.alt)
type1.power	Probability of rejecting null under null (if ve = ve.null) or alternative (if ve = ve.alt)
prob.VE	Probabilities that x <= efficacy and x >= futility boundaries-number of events, efficacy and futility boundaries under null (if ve = ve.null) or alternative (if ve = ve.alt)
percent.stop	Percentage of average sample sizes at which decision can be made under null and alternative
summary.stat	Summary statistics of events: min, Q1, Q2, mean, Q3, max, sd under null and alternative
time	Time to run the code under null and alternative
method	Provide "Freq" for Frequentist approach; "Bayes" for Bayesian approach
n.sim	Simulation size
n.event	Total number of events
n.start	Starting value of the event
n.increment	Differences between two interims
ve.null	Vaccine efficacy under null
ve.alt	Vaccine efficacy under alternative
type1	Specified Type I error
power	Specified power, by default 0.9
ratio.alloc	Treatment vs control randomization = M:1, value of M
optimum.w	Optimum weight
w	Weight for calculating shape parameter of Beta distibution
a.prior	Shape parameter of Beta prior
b.prior	Scale parameter of Beta prior
seed	Seed value to simulate data from the binomial distribution

Author(s)

Erina Paul <erina.paul@merck.com>

Examples

```
method = "Bayes"
n.sim = 10000
n.event = 60
n.start = 30
n.increment = 5
ve.null = 0.3
ve.alt = 0.75
type1 = 0.025
power = 0.9
ratio.alloc = 1
optimum.w = TRUE
b.prior = 1
seed = 134

opt.w = SPRToptimumW(n.sim = n.sim, n.event = n.event, n.start = n.start,
                     n.increment = 1, ve.null = ve.null, ve.alt = ve.alt,
                     type1 = type1, power = power, ratio.alloc = ratio.alloc,
                     w = seq(0.1, 0.9, by = 0.1), b.prior = b.prior,
                     seed = seed)

results = outputSPRT(method = method, n.sim = n.sim, n.event = n.event,
                     n.start = n.start, n.increment = n.increment,
                     ve.null = ve.null, ve.alt = ve.alt,
                     type1 = type1, power = power, ratio.alloc = ratio.alloc,
                     optimum.w = optimum.w, w = opt.w$weight,
                     b.prior = b.prior, seed = seed)
results
```

[Package *bayesiansprt* version 0.1.0]