Package 'stoppingrule'

April 19, 2022

Type Package

Title Create and Evaluate Stopping Rules

Version 0.1.0

Author Michael J. Martens <mmartens@mcw.edu>

Maintainer Michael J. Martens <mmartens@mcw.edu>

Description Provides functions for creating, displaying, and evaluating stopping rules for safety monitoring in clinical studies. Implements stopping rule methods described in Goldman (1987) <doi:10.1016/0197-2456(87)90153-X>, Geller et al. (2003, ISBN:9781135524388), Ivanova, Qaqish, and Schell (2005) <doi:10.1111/j.1541-0420.2005.00311.x>, and Kulldorff et al. (2011) <doi:10.1080/07474946.2011.539924>.

Imports stats, utils

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2022-04-19 08:00:02 UTC

R topics documented:

---

calc.bnd *Stopping Boundary Calculation*

Description

Internal workhorse function to calculate stopping boundary for a given method

Usage

```
calc.bnd(n, p0, cval, type, param)
```

Arguments

n	Maximum sample size for safety monitoring
p0	The toxicity rate under the null hypothesis
cval	Critical value for stopping rule method
type	The method used for constructing the stopping rule
param	Extra parameter(s) needed for certain stopping rule methods. For binomial Wang-Tsiatis tests, this is the Delta parameter. For Bayesian beta-binomial model, this is the pair of hyperparameters for the beta prior on the toxicity rate. For modified SPRT, this is the targeted alternative toxicity rate p1.

Value

A vector of stopping boundaries at the sample sizes 1, 2, ..., n

---

calc.rule *Stopping Rule Calculation*

Description

Calculate a stopping rule for safety monitoring

Usage

```
calc.rule(ns, p0, type, param = NULL, alpha, iter = 50)
```

Arguments

ns	A vector of sample sizes at which sequential testing is performed
p0	The toxicity rate under the null hypothesis
type	The method used for constructing the stopping rule. Choices include a Pocock test ("Pocock"), a Wang-Tsiatis test ("WT"), a Bayesian beta-binomial model ("BB"), a truncated SPRT ("SPRT"), and a maximized SPRT ("MaxSPRT").
param	Extra parameter(s) needed for certain stopping rule methods. For binomial Wang-Tsiatis tests, this is the Delta parameter. For Bayesian beta-binomial model, this is the pair of hyperparameters for the beta prior on the toxicity rate. For modified SPRT, this is the targeted alternative toxicity rate p1.
alpha	The desired type I error / false positive rate for the stopping rule
iter	The number of iterations used to search for the boundary

Value

A matrix with two columns: the sample sizes ns and their corresponding rejection boundaries

References

Goldman, A.I. (1987). Issues in designing sequential stopping rules for monitoring side effects in clinical trials. *Controlled clinical trials* **8(4)**, 327-37.

Geller, N.L., Follman, D., Leifer, E.S. and Carter, S.L. (2003). Design of early trials in stem cell transplantation: a hybrid frequentist-Bayesian approach. *Advances in Clinical Trial Biostatistics*.

Ivanova, A., Qaqish, B.F. and Schell, M.J. (2005). Continuous toxicity monitoring in phase II trials in oncology. *Biometrics* **61(2)**, 540-545.

Kulldorff, M., Davis, R.L., Kolczak, M., Lewis, E., Lieu, T. and Platt, R. (2011). A maximized sequential probability ratio test for drug and vaccine safety surveillance. *Sequential analysis* **30(1)**, 58-78.

Pocock, S.J. (1977). Group sequential methods in the design and analysis of clinical trials. *Biometrika* **64(2)**, 191-199.

Wang, S.K. and Tsiatis, A.A. (1987). Approximately optimal one-parameter boundaries for group sequential trials. *Biometrics* **193-199**.

Examples

```
# Binomial Pocock test in 50 patient cohort at 10% level, expected toxicity rate of 20%
calc.rule(ns=1:50,p0=0.20,type="Pocock",alpha=0.10)
```

---

findconst   *Search for Calibration Value*

Description

Internal workhorse function to calculate the calibration constant value that attains level alpha for given method

Usage

```
findconst(ns, p0, type, alpha, l, u, iter = 50, param)
```

Arguments

ns	A vector of sample sizes at which sequential testing is performed
p0	The toxicity rate under the null hypothesis
type	The method used for constructing the stopping rule
alpha	The desired type I error / false positive rate for the stopping rule
l	Lower starting value of bracket for calibration constant
u	Upper starting value of bracket for calibration constant
iter	The number of iterations used to search for the boundary
param	Extra parameter(s) needed for certain stopping rule methods. For binomial Wang-Tsiatis tests, this is the Delta parameter. For Bayesian beta-binomial model, this is the pair of hyperparameters for the beta prior on the toxicity rate. For modified SPRT, this is the targeted alternative toxicity rate p1.

Value

The calibration constant used for subsequent stopping boundary calculation

---

lines.rule   *Add Stopping Rule Curve to Current Plot*

Description

Add a stopping rule graphically as a curve on current plot

Usage

```
## S3 method for class 'rule'
lines(x, ...)
```

Arguments

x   A rule object, being a matrix with two columns: the sample sizes at which sequential testing is performed, and their corresponding rejection boundaries

...   Other options to be passed to generic `lines` function

Value

No return value; function solely modifies current plot

Examples

```
# Binomial Pocock test in 50 patient cohort at 10% level, expected toxicity rate of 20%
poc_rule = calc.rule(ns=1:50,p0=0.20,type="Pocock",alpha=0.10)
# Bayesian beta-binomial monitoring in 50 patient cohort at 10% level, expected toxicity rate of 20%
bb_rule = calc.rule(ns=1:50,p0=0.20,type="BB",alpha=0.10,param=c(1,7))

# Plot stopping boundaries for stopping rules
plot(poc_rule,col="blue")
lines(bb_rule,col="red")
```

OC.rule   *Operating Characteristics Function*

Description

Compute operating characteristics for a stopping rule at a set of toxicity rates. Characteristics calculated include the overall rejection probability, the expected number of patients evaluated, and the expected number of events.

Usage

```
OC.rule(rule, ps)
```

Arguments

rule   A matrix with two columns: the sample sizes at which sequential testing is performed, and their corresponding rejection boundaries

ps   A vector of toxicity rates at which the operating characteristics will be computed

Value

A matrix with four columns: the toxicity rates ps, the corresponding rejection probabilities, the corresponding expected numbers of evaluated patients, and the corresponding expected numbers of events

Examples

```
# Binomial Pocock test in 50 patient cohort at 10% level, expected toxicity rate of 20%
poc = calc.rule(ns=1:50,p0=0.20,type="Pocock",alpha=0.10)

# Compute operating characteristics at toxicity rates of 20%, 25%, 30%, 35%, and 40%
OC.rule(rule=poc,ps=seq(0.2,0.4,0.05))
```

---

opchars   *Operating Characteristics Function*

Description

Internal workhorse function to calculate operating characteristics for a given stopping rule and event rate

Usage

```
opchars(rule, p)
```

Arguments

rule   A matrix with two columns: the sample sizes at which sequential testing is performed, and their corresponding rejection boundaries

p   The event probability

Value

A list with four objects: the toxicity rate p, the corresponding rejection probability, the corresponding expected number of evaluated patients, and the corresponding expected number of events

---

plot.rule   *Plot Stopping Rule*

Description

Display a stopping rule graphically as a curve

Usage

```
## S3 method for class 'rule'
plot(x, ...)
```

Arguments

x   A rule object, being a matrix with two columns: the sample sizes at which sequential testing is performed, and their corresponding rejection boundaries

...   Other options to be passed to generic plot function

Value

No return value; function solely generates a plot

Examples

```
# Binomial Pocock test in 50 patient cohort at 10% level, expected toxicity rate of 20%
poc_rule = calc.rule(ns=1:50,p0=0.20,type="Pocock",alpha=0.10)

# Plot stopping boundary with smoothing
plot(poc_rule,col="blue")
```

---

print.rule   *Print Stopping Rule*

Description

Display a stopping rule in tabular form

Usage

```
## S3 method for class 'rule'
print(x, ...)
```

Arguments

x   A rule object, containing a matrix describing a stopping rule

...   Other options to be passed to generic `plot` function

Value

A matrix with two columns: the sample sizes at which sequential testing is performed, and their corresponding rejection boundaries

Examples

```
# Binomial Pocock test in 50 patient cohort at 10% level, expected toxicity rate of 20%
poc_rule = calc.rule(ns=1:50,p0=0.20,type="Pocock",alpha=0.10)

# Print stopping rule in table
print(poc_rule)
```

---

smooth.bnd   *Smooth Stopping Rule Boundary*

Description

Function to compute smoothed stopping rule boundary. Takes as input a stopping rule computed by `calc.rule()`, where the rejection boundary is a step function, and returns a piecewise linear stopping boundary with identical rejection criteria.

Usage

```
smooth.bnd(rule)
```

Arguments

rule   A matrix with two columns: the sample sizes at which the stopping rule is evaluated and their corresponding rejection boundaries

Value

A matrix with two columns: the sample sizes at which the stopping rule is evaluated and their corresponding smoothed rejection boundaries

---

stoppingrule   *Create and Evaluate Stopping Rules*

Description

Provides functions for creating, displaying, and evaluating stopping rules for safety monitoring in clinical studies.

Author(s)

Michael J. Martens <mmartens@mcw.edu>

---

table.rule   *Tabulate Stopping Rule*

Description

Summarize a stopping rule in a condensed tabular format

Usage

```
table.rule(rule)
```

Arguments

rule   A matrix with two columns: the sample sizes at which sequential testing is
                performed, and their corresponding rejection boundaries

Value

A matrix with two columns: the ranges of evaluable patients, and corresponding rejection bound-
aries for these ranges

Examples

```
# Binomial Pocock test in 50 patient cohort at 10% level, expected toxicity rate of 20%
poc_rule = calc.rule(ns=1:50,p0=0.20,type="Pocock",alpha=0.10)

# Tabulate stopping boundary
table.rule(poc_rule)
```

Index