Package: harmBounds (via r-universe)

May 22, 2026

Type Package

Title Harm Boundaries for Clinical Trials

Version 0.4.0

Author Yunda, Lukas

Maintainer Lukas Bütikofer <`lukas.buetikofer@unibe.ch`>

Description The harmBounds package calculates stopping probabilities,
defines stopping boundaries and generates plots for safety
monitoring using an event based approach.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

URL `https://github.com/dcr-unibe-ch/harmBounds`

BugReports `https://github.com/dcr-unibe-ch/harmBounds/issues`

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports ggplot2, dplyr

Depends R (>= 4.1)

Repository https://ctu-bern.r-universe.dev

Date/Publication 2026-03-23 17:02:53 UTC

RemoteUrl https://github.com/CTU-Bern/harmBounds

RemoteRef HEAD

RemoteSha 9a751ecb6e470dec1ee81bebac5b047c0cd16d69

Contents

`absstopPlot` *Plot absolute stopping probs*

Description

Plot absolute stopping probs

Usage

```
absstopPlot(harmbound)
```

Arguments

`harmbound`	harmbounds objects as generated using the getHarmBound function

Value

barplot with stopping probabilities under H0 and optionally H1

Examples

```
harmbound<-getHarmBound(nevents=seq(10,100,by=10),alpha_test=0.025,pH0=0.5,pH1=c(0.6,0.7,0.8))
absstopPlot(harmbound)
```

`convertRisks` *Convert the proportion of events in the intervention groups to risk differences and ratios and vice versa.*

Description

Convert the proportion of events in the intervention groups to risk differences and ratios and vice versa.

Usage

```
convertRisks(
  eprop = NULL,
  etotal = NULL,
  rd = NULL,
  rr = NULL,
  or = NULL,
  r0 = NULL,
  n0,
  n1 = n0
)
```

Arguments

`eprop`	proportion of events in intervention group
`etotal`	total number of events
`rd`	risk difference
`rr`	risk ratio
`or`	odds ratio
`r0`	risk in the control group
`n0`	number of patients in the control group
`n1`	number of patients in the intervention group

Value

vector with risks in control and intervention group (r0, r1), the risk difference (rd), risk ratio (rr) and odds ratio (or)

Examples

```
convertRisks(eprop=0.5,etotal=100,n0=200)
convertRisks(eprop=0.6,etotal=100,n0=200)
convertRisks(rr=1.5,n0=200,r0=0.2)
```

`cumstopPlot` *Plot cumulative stopping probs*

Description

Plot cumulative stopping probs

Usage

```
cumstopPlot(harmbound)
```

Arguments

`harmbound`	harmbounds objects as generated using the getHarmBound function

Value

barplot with cumulative stopping probabilities under H0 and optionally H1

Examples

```
harmbound<-getHarmBound(nevents=seq(10,100,by=10),alpha_test=0.025,pH0=0.5,pH1=c(0.6,0.7,0.8))
cumstopPlot(harmbound)
```

`findbound` *Find stopping boundary via binomial exact tests*

Description

Find stopping boundary via binomial exact tests

Usage

```
findbound(n, alpha_test = 0.025, pH0 = 0.5, alternative = "greater")
```

Arguments

`n`	total number of events
`alpha_test`	nominal alpha for the binomial test
`pH0`	proportion of events in the experimental arm under the null hypothesis, typically based on randomization ratio (e.g. 0.5 for a 1:1 randomization)
`alternative`	direction of alternative, "less" or "greater"

Value

number of events in the experimental group that would lead to a stopping

Examples

```
findbound(n=20, alpha_test=0.025, pH0 = 0.5, alternative="greater")
findbound(n=20, alpha_test=0.025, pH0 = 0.5, alternative="less")
```

`getAlphaPerTest` *getAlphaPerTest*

Description

Test-wise alpha necessary to control the overall type I error at a specified level (0.05 by default)

Usage

```
getAlphaPerTest(
  nevents,
  totalAlpha = 0.05,
  pH0 = 0.5,
  alpha.interval = c(10^(-10), 0.05)
)
```

Arguments

`nevents`	vector with number of events at which an interim analysis is done
`totalAlpha`	Overall type I error, 0.05 by default
`pH0`	proportion of events in the intervention arm under the null hypothesis, typically based on randomization ratio (e.g. 0.5 for a 1:1 randomization)
`alpha.interval`	Range for test-wise alpha, c(10^(-10),0.05) by default

Value

Test-wide alpha

Examples

```
apt<-getAlphaPerTest(nevents = c(10,50,100), totalAlpha = 0.05, pH0 = 0.5)
apt
getHarmBound(nevents = c(10,50,100),alpha_test = apt, pH0 = 0.5)
```

`getHarmBound` *Harm boundaries for safety testing*

Description

Calculates the boundaries at each interim analysis, i.e. the number of events in the intervention group that would lead to a stopping of the trial based on a binomial exact test, assuming that the events should be equally distributed amont both groups. The indicated scenario (and all more extreme) would lead to a rejection of H0 (equal distribution) and a stopping for safety.

Usage

```
getHarmBound(
  nevents,
  alpha_test,
  pH0,
  maxevents = NULL,
  pH1 = NULL,
  rrH1 = NULL,
  orH1 = NULL,
  rdH1 = NULL,
  r0 = NULL,
  n = NULL
)
```

Arguments

`nevents`	vector with number of events (over both arms) at which an interim analysis is done
`alpha_test`	the nominal alpha level to use for each test
`pH0`	proportion of events in the intervention arm under the null hypothesis, typically based on randomization ratio (e.g. 0.5 for a 1:1 randomization)
`maxevents`	optional maximum number of events expected for the trial (over both arms), used to calculate the expected number of events
`pH1`	optional alternative, numeric vector, proportion of events in the intervention arm
`rrH1`	alternative specification of alternative as risk ratio (intervention / control)
`orH1`	alternative specification of alternative as risk ratio (intervention / control). Requires the control proportion (r0).
`rdH1`	alternative specification of alternative as risk difference (intervention - control). Requires the control proportion (r0) and the number of participants (n).
`r0`	risk in the control group. Required if the alternative is given as risk difference or odds ratio.
`n`	total number of participants. Required if the alternative is given as risk difference.

Value

a list with 3 data.frames: bounds, stopprob and opchar. bounds has a row for each interim analysis and columns for number of events (events), number of events in control and intervention group that would lead to a stop (events_intervention, events_control), and the nominal alpha for each test (alpha_test). stopprob has a row for each interim analysis and columns for number of events (events), the hypothesis (pH), the stopping probability (stop_prob), and the cumulative stopping probability (cum_stop_prob) opchar has a row for each hypothesis (null plus each alternative) and columns for the assumed proportion of events in the intervention group (p), the cumulative stopping probabilities (cum_stop_prob) and the expected total number of events (expected_events) for the null and each alternative.

Examples

```
getHarmBound(nevents=c(10,50,100), alpha_test=0.025, pH0=0.5)
#adding an alternative
getHarmBound(nevents=c(10,50,100), alpha_test=0.025, pH0=0.5, pH1=0.6)
#assume that a total of 150 events might occur
getHarmBound(nevents=c(10,50,100), alpha_test=0.025, pH0=0.5, pH1=0.6, maxevents=150)
#or several alternatives
getHarmBound(nevents=c(10,50,100), alpha_test=0.025, pH0=0.5,
pH1 = seq(0.6,0.8,by=0.05), maxevents=150)
#or as risk ratio
getHarmBound(nevents=c(10,50,100), alpha_test=0.025, pH0=0.5, rrH1=1.5, maxevents=150)
```

`harmboundPlot` *Plot harmbounds and the observed events*

Description

Plot harmbounds and the observed events

Usage

```
harmboundPlot(
  harmbound,
  observed = NULL,
  colourbound = c("blue", "red"),
  fill_alpha = 0.5,
  colourobserved = "black",
  H0line = TRUE
)
```

Arguments

`harmbound`	harmbounds objects as generated using the getHarmBound function
`observed`	optional observed number of events, as a vector with the sequential groups in which an event occured (0 for control and 1 for intervention)

`colourbound`	vector with two colours for the bounds, in and out, default is blue and red
`fill_alpha`	opacity of the colours for the bounds
`colourobserved`	colour for the line with the observed events
`H0line`	logical, whether a line to indicate the expectations should be added.

Value

plot with the bounds and optionally the observed number of events

Examples

```
harmbound<-getHarmBound(nevents=seq(10,100,by=10),alpha_test=0.025,pH0=0.5)
harmboundPlot(harmbound)
set.seed(123)
eventgroups<-rbinom(n=100,size=1,prob=0.5)
harmboundPlot(harmbound,observed=eventgroups)
```

`opcharNPlot` *Plot expected number of events*

Description

Plot expected number of events

Usage

```
opcharNPlot(harmbound)
```

Arguments

`harmbound`	harmbounds objects as generated using the getHarmBound function

Value

line plot with the expected number of events for all alternatives

Examples

```
harmbound<-getHarmBound(nevents=seq(10,100,by=10),alpha_test=0.025,
pH0=0.5,pH1=seq(0.5,0.8,by=0.05),maxevents=150)
opcharNPlot(harmbound)
```

`opcharStopPlot` *Plot cumulative stopping probability by hypothesis*

Description

Plot cumulative stopping probability by hypothesis

Usage

```
opcharStopPlot(harmbound)
```

Arguments

`harmbound`	harmbounds objects as generated using the getHarmBound function

Value

line plot with the cumulative stopping for all alternatives

Examples

```
harmbound<-getHarmBound(nevents=seq(10,100,by=10),alpha_test=0.025,
pH0=0.5,pH1=seq(0.5,0.8,by=0.05),maxevents=150)
opcharStopPlot(harmbound)
```

`plot.harmbound` *Plot method for harmbound objects produced by* `getHarmBound`

Description

Plot method for harmbound objects produced by `getHarmBound`

Usage

```
## S3 method for class 'harmbound'
plot(x, which = "bounds", ...)
```

Arguments

`x`	harmbounds objects as generated using the getHarmBound function
`which`	one of `"bounds"`, `"abs_stopping"`, `"cum_stopping"`, `"exp_n"`.
`...`	options passed to plot

Examples

```
harmbound<-getHarmBound(nevents = seq(10, 100, by=10),alpha_test = 0.025,
pH0 = 0.5, pH1 = seq(0.55,0.7,by=0.05), maxevents = 150)
plot(harmbound, which = "bounds")
plot(harmbound, which = "abs_stopping")
plot(harmbound, which = "cum_stopping")
plot(harmbound, which = "opchar_stop")
plot(harmbound, which = "opchar_n")
```

`simSafetyStop` *Simulate safety stopping values. The effect can be given as proportion of events in the experimental group (pH1), the risk difference (rdH1), risk ratio (rrH1) or odds ratio (orH1).*

Description

Simulate safety stopping values. The effect can be given as proportion of events in the experimental group (pH1), the risk difference (rdH1), risk ratio (rrH1) or odds ratio (orH1).

Usage

```
simSafetyStop(
  nevents,
  pH0 = 0.5,
  alpha_test = 0.025,
  pH1 = NULL,
  rrH1 = NULL,
  orH1 = NULL,
  rdH1 = NULL,
  r0 = NULL,
  n = NULL
)
```

Arguments

`nevents`	vector with number of events at which an interim analysis is done
`pH0`	proportion of events in the experimental arm under the null hypothesis, typically based on randomization ratio (e.g. 0.5 for a 1:1 randomization)
`alpha_test`	nominal alpha level for binomial exact test
`pH1`	proportion of events in the experimental arm under the alternative hypothesis
`rrH1`	risk ratio (experimental / control).
`orH1`	risk ratio (experimental / control). Requires the control proportion (r0).
`rdH1`	risk difference (experimental - control). Requires the control proportion (r0) and the number of participants (n).

`r0`	risk in the control group. Required if the effect is given as risk difference or odds ratio.
`n`	total number of participants. Required if the effect is given as risk difference.

Value

list with a dataframe with number of events in each group plus upper limit for stopping and indicator for whether stopped, plus indicators number of stops and time points at first stop

Examples

```
set.seed(1)
simSafetyStop(nevents=seq(10,100,by=10),pH0 = 0.5, pH1 = 0.6,alpha_test=0.025)

set.seed(1)
simSafetyStop(nevents=seq(10,100,by=10),pH0 = 0.5, rrH1 = 0.6/(1-0.6), alpha_test=0.025)
```

Index