Package ‘brada’

January 24, 2023

Type Package

Title Bayesian Response-Adaptive Design Analysis

Version 1.0

Date 2023-01-18

Description Provides access to a range of functions for analyzing, applying and visualizing Bayesian response-adaptive trial designs for a binary endpoint. Includes the predictive probability approach and the predictive evidence value designs for binary endpoints.

Imports methods, fbst, extraDistr, doParallel, foreach, parallel, doSNOW, progress, cli

Suggests knitr, rmarkdown, DT

License GPL-3

VignetteBuilder knitr

NeedsCompilation no

Author Riko Kelter [aut, cre] (<`https://orcid.org/0000-0001-9068-5696`>)

Maintainer Riko Kelter <`riko.kelter@uni-siegen.de`>

Repository CRAN

Date/Publication 2023-01-24 10:40:11 UTC

R topics documented:

---

brada-package *Bayesian Response-Adaptive Design Analysis*

Description

Provides access to a range of functions for analyzing, applying and visualizing Bayesian response-adaptive trial designs for a binary endpoint. Includes the predictive probability approach and the predictive evidence value designs for binary endpoints.

Details

Provides access to a range of functions for analyzing, applying and visualizing
Bayesian response-adaptive trial designs for a binary endpoint. Includes the
predictive probability approach and the predictive evidence value designs for
binary endpoints.

Package:	brada
Type:	Package
Title:	Bayesian Response-Adaptive Design Analysis
Version:	1.0
Date:	2023-01-18
Authors@R:	c(person(given = "Riko", family = "Kelter", role = c("aut", "cre"), email = "riko.kelter@uni-siegen.de", co
Description:	Provides access to a range of functions for analyzing, applying and visualizing Bayesian response-adaptive
Imports:	methods, fbst, extraDistr, doParallel, foreach, parallel, doSNOW, progress, cli
Suggests:	knitr, rmarkdown, DT
License:	GPL-3
VignetteBuilder:	knitr
Author:	Riko Kelter [aut, cre] (<https://orcid.org/0000-0001-9068-5696>)
Maintainer:	Riko Kelter <riko.kelter@uni-siegen.de>

Index of help topics:

```
$,brada-method          Returns an object from an object of class
                        'brada'.
brada                   brada
brada-class             Class '"brada-class"'
brada-package           Bayesian Response-Adaptive Design Analysis
calibrate               calibrate
generateData            generateData
monitor                 monitor
names.brada             names.brada
plot.brada              plot.brada
power                   power
show.brada              show.brada
summary.brada           summary.brada
```

Author(s)

NA

Maintainer: NA

`access-method` *Returns an object from an object of class* `brada`.

Description

Returns an object from an object of class `brada`

Details

-

Value

No return value.

Author(s)

Riko Kelter

`brada` *brada*

Description

Performs a Bayesian response-adaptive design analysis for trials with a binary endpoint.

Usage

```
brada(a0=1,b0=1,Nmax=40,batchsize=5,nInit,p_true,p0,p1,
theta_T=0.90,theta_L=0.1,theta_U=1,nsim=100,
seed=42,method="PP",refFunc="flat",nu=0,
shape1=1,shape2=1,truncation=1,cores=2)
```

Arguments

`a0`	shape1 parameter of the beta prior.
`b0`	shape2 parameter of the beta prior.
`Nmax`	Maximum trial size.
`batchsize`	sample size after which an interim analysis is performed.
`nInit`	Initial sample size at which the first interim analysis is performed.
`p_true`	True binary response probability used for simulation.
`p0`	Right boundary of the null hypothesis to be tested.
`p1`	Left boundary of the alternative hypothesis to be tested.
`theta_T`	Threshold used in the designs for including trajectories as evidential.
`theta_L`	Stopping threshold for futility.
`theta_U`	Stopping threshold for efficacy.
`nsim`	Number of Monte Carlo iterations.
`seed`	Random number generator seed.
`cores`	Number of CPU cores to be used for computation. Defaults to 2, but 4 or larger is recommended.
`method`	Can be either "PP" or "PPe", depending on whether the predictive probability approach or the predictive evidence value design is desired. Note that the former is a special case of the latter.
`refFunc`	A string, either "flat", "beta", "binaryStep", "relu", "palu" or "lolu". See vignettes for explanation.
`nu`	A numeric value larger or equal to zero, indicating which evidence threshold if used in the predictive evidence value design.
`shape1`	shape1 parameter of the beta reference function, if used.
`shape2`	shape2 parameter of the beta reference function, if used.
`truncation`	Truncation point in case an artificial neural network reference function is used.

Value

Returns an object of class brada.

Author(s)

Riko Kelter

Examples

```
pp_design = brada(Nmax = 30, batchsize = 5, nInit = 10,
              p_true = 0.2 , p0 = 0.2, p1 = 0.2,
              nsim = 10,
              a0 = 1, b0 = 1,
              theta_T = 0.90, theta_L = 0.1, theta_U = 1,
              method = "PP",
              cores = 2)
summary(pp_design)
```

`brada-class` *Class* `"brada-class"`

Description

Class for modelling the results of a Bayesian response-adaptive design analysis

Objects from the Class

Store the results of a Bayesian response-adaptive design analysis

Slots

`data`**:** Object of class `"list"` holding the results of the Bayesian response-adaptive design analysis. `a0` and `b0` store the beta prior shape parameters, `Nmax` and `batchsize` store the maximum trial size and the batchsize used for interim analyses. `nInit` is the minimum sample size at which the first interim analysis is conducted. `p_true` is the true response probability used for simulation, `p0` is the right boundary of the null hypothesis and `p1` the left boundary of the alternative hypothesis. ...

`calibrate` *calibrate*

Description

Calibrates a brada object to achieve specified false-positive and false-negative rates.

Usage

```
calibrate(brada_object, nsim = 100, cores = 2, seq,
alpha=NULL, beta=NULL, calibration = "nu")
```

Arguments

`brada_object`	An object of class `brada`
`nsim`	Number of Monte Carlo iterations
`cores`	Number of cores used for computation
`seq`	Sequence of values for the evidence threshold `nu`, or sequence of the futility threshold `theta_L`, depending on which value is passed to the function in the `calibration` argument. For example, `seq=seq(0,1,0.1)` in combination with `calibration="nu"` and `alpha=0.1` implies that the function tries to calibrate the false-positive rate to decrease below `alpha=0.1` by iteratively increasing `nu` from zero to one in steps of size 0.1.

`alpha`	Upper bound for false-positive rate. Note that it is only possible to specify either `alpha` or `beta`. When `alpha` is passed as an argument, `calibration` should take the value `"nu"`, while when `beta` is passed as an argument, `calibration` should take the value `"theta_L"`.
`beta`	Upper bound for false-negative rate
`calibration`	String which specifies which parameter to calibrate. Can take the values `"nu"` or `"theta_L"`.

Value

Prints the output to the console and returns the false-positive rate or false-negative rate of the calibrated design, depending on which value the `calibration` argument takes.

Author(s)

Riko Kelter

`generateData` *generateData*

Description

Generates a matrix of trial data.

Usage

```
generateData(p,Nmax,nsim,seed=420)
```

Arguments

`p`	true response probability
`Nmax`	Maximum trial size.
`nsim`	Number of Monte Carlo iterations.
`seed`	Random number generator seed.

Value

Returns a matrix with simulated trial data.

Author(s)

Riko Kelter

Examples

```
generateData(p=0.2,Nmax=40,nsim=100,seed=420)
```

`monitor` *monitor*

Description

Monitors a running trial with a binary endpoint and calculates the predictive probability or predictive evidence that the trial will result in a success. Reports whether to stop early for futility or efficacy based on a vector of binary observations.

Usage

```
monitor(brada_object, obs)
```

Arguments

`brada_object`	An object of class `brada`.
`obs`	A vector of binary observations, where 1 is a success (response) and 0 a failure (no response).

Value

No return value, prints the result of the monitoring to the console.

Author(s)

Riko Kelter

Examples

```
design = brada(Nmax = 40, batchsize = 5, nInit = 10,
                   p_true = 0.2 , p0 = 0.2, p1 = 0.2,
                   nsim = 100,
                   a0 = 1, b0 = 1,
                   theta_T = 0.95, theta_L = 0.05, theta_U = 0.975,
                   method = "PP",
                   cores = 2)
monitor(design, obs = c(0,1,1,0,0,1,0,1,1,1))
```

`names.brada` *names.brada*

Description

Plots the names of the objects stored in the `brada` object of a Bayesian response-adaptive design analysis.

Usage

```
## S3 method for class 'brada'
names(x)
```

Arguments

`x`	An Object of class `"brada"`.

Details

Plots the names of the objects stored in the `trials` object of a Bayesian response-adaptive design analysis.

Value

Returns a list of names.

Author(s)

Riko Kelter

`plot.brada` *plot.brada*

Description

Plots the results of a Bayesian response-adaptive design analysis.

Usage

```
## S3 method for class 'brada'
plot(x, trajectories = 100, ...)
```

Arguments

`x`	An Object of class `"brada"`.
`trajectories`	Number of trajectories to be plotted. Defaults to 100.
`...`	Additional parameters, see `"plot(x, ...)"`.

Value

Returns a plot.

Author(s)

Riko Kelter

Examples

```
design = brada(Nmax = 40, batchsize = 5, nInit = 10,
               p_true = 0.2 , p0 = 0.2, p1 = 0.2,
               nsim = 100,
               a0 = 1, b0 = 1,
               theta_T = 0.90, theta_L = 0.1, theta_U = 1,
               method = "PP",
               cores = 2)
plot(design)
```

`power` *power*

Description

Performs a power analysis for a `brada` object.

Usage

```
power(brada_object, p_true, nsim=100, cores=2)
```

Arguments

`brada_object`	An object of class `brada`
`p_true`	the true response probability used for the power analysis
`nsim`	the number of Monte Carlo simulation, defaults to 100.
`cores`	CPU cores used for computation. Defaults to 2.

Value

Returns an object of class `brada`.

Author(s)

Riko Kelter

Examples

```
design = brada(Nmax = 30, batchsize = 5, nInit = 10,
               p_true = 0.2 , p0 = 0.2, p1 = 0.2,
               nsim = 1000,
               a0 = 1, b0 = 1,
               theta_T = 0.90, theta_L = 0.1, theta_U = 1,
               method = "PP",
               cores = 1)
    design_power = power(design, p_true = 0.4, nsim = 1000)
    plot(design_power)
```

`show.brada` *show.brada*

Description

Prints the main results of a Bayesian response-adaptive design analysis to the console.

Usage

```
## S3 method for class 'brada'
show(object)
```

Arguments

`object`	An Object of class "`brada`".

Details

Shows the main results of a Bayesian response-adaptive design analysis stored in an object of class `brada`.

Value

Prints the results onto the console.

Author(s)

Riko Kelter

`summary.brada` *summary.brada*

Description

Prints the results of a Bayesian response-adaptive design analysis.

Usage

```
## S3 method for class 'brada'
summary(object, ...)
```

Arguments

`object`	An Object of class "`brada`".
`...`	Additional parameters, see "`summary(object, ...)`".

Details

Summarises the results of a Bayesian response-adaptive design analysis.

Value

Prints the results onto the console.

Author(s)

Riko Kelter

Examples

```
pp_design = brada(Nmax = 40, batchsize = 5, nInit = 10,
              p_true = 0.2 , p0 = 0.2, p1 = 0.2,
              nsim = 100,
              a0 = 1, b0 = 1,
              theta_T = 0.90, theta_L = 0.1, theta_U = 1,
              method = "PP",
              cores = 2)
summary(pp_design)
```

Index