Package ‘InferenceBEAGSD’

September 12, 2018

Type Package

Title Inference for Binary Endpoint Adaptive Group-Sequential Designs

Version 0.1.0

Author Arsenio Nhacolo

Maintainer Arsenio Nhacolo <anhacolo@uni-bremen.de>

Description This package implents the inference methods proposed in the doctoral thesis intitled 'Bias and precision in early phase adaptive oncology studies and its consequences for confirmatory trials' authored by Arsenio Nhacolo. It includes functions for comparing the performance of various estimators for classical two-stage group-sequential designs with binary endpoint popular in oncology Phase II clinical trials, new inference methods (p-values, and point and interval estimates) proposed for adaptive versions of these designs, and new methods for estimating adjustment factors in order to get an adequately powered Phase III trials when planned based on Phase II trial data.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

R topics documented:

`adjustMet1` *Phase II efficacy estimates/Phase III sample size adjustment factors (Method 1).*

Description

`adjustMet1` calculates the multiplicative ajustment factor f to be applied to Phase II efficacy estimate, and the factor ρ to be applied to Phase III sample size estimate using Method 1 proposed by Nhacolo and Brannath (in press).

Usage

```
adjustMet1(p2d, p2r, p2e, p2p0 = NULL, p2p1 = NULL, p2a = NULL,
  p2b = NULL, p3p0 = NULL, p3p1 = NULL, p3a = NULL, p3b = NULL,
  nsimul = 5000, seed = NULL)
```

Arguments

`p2d` Dataframe with Phase II design, with similar as in `EKOptAdaptDesigns`.

`p2r` Dataframe containing results of Phase II trials following the design `p2d`. It is the output of the function `AnalyzeEKOAD`.

`p2e` Phase II estimate to consider among the estimates used by codeAnalyzeEKOAD. It can be `"pip"` (naive MLE) or one of the four estimates from methods proposed by Nhacolo and Brannath (2018): `"pim1"`, `"pim2"`, `"pim2v2"` or `"pim3"`.

`p2p0` Phase II response rate under H_0. If `NULL` (default), the value is taken `p2d`.

`p2p1` Phase II response rate under H_1. If `NULL` (default), the value is taken `p2d`.

`p2a` Phase II type I error rate. If `NULL` (default), the value is taken `p2d`.

`p2b` Phase II type II error rate. If `NULL` (default), the value is taken `p2d`.

`p3p0` Phase III response rate of the control group. If `NULL` (default), the value is set to `p2p0`.

`p3p1` Phase III response rate of the treatment group. If `NULL` (default), the value is set to `p2p1`.

`p3a` Phase III type I error rate. If `NULL` (default), the value is set `p2a`.

`p3b` hase III type II error rate. If `NULL` (default), the value is set `p2b`.

`nsimul` Number of (parametric) bootstrap samples (default 5000).

`seed` Seed for random number generator. If `NULL` (default), no seed is set.

Details

The aim of the adjustment is to get an adequately powered Phase III trial based on Phase II data. See the documentation of the function `AnIItoIIIRe` for more details about the designs.

Value

A list containing two dataframes `final` and `intermed`. `final` contains the final measures for the adjustment factors (f and ρ) and power. `intermed` holds the intermediate results (of each bootstrap sample).

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Using Estimates from Adaptive Phase II Oncology Trials to Plan Phase III Trials. *In press*.

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

Ahn, C., Heo, M. and Zhang, S. *Sample Size Calculations for Clustered and Longitudinal Outcomes in Clinical Research*. CRC Press, 2014.

See Also

`adjustMet1`, `SimulateEKOAD`, `AnalyzeEKOAD`.

Examples

```
## Not run:
vdid <- c(6,10) # design ids
vp2est <- c("pip","pim1","pim2","pim2v2","pim3")
nse <- 1000#number of simulations for each phase
cur <- 1; tot <- length(vdid)*length(vp2est)
for (did in vdid){
 for (p2est in vp2est){
   cat('Processing ',cur,' of ',tot,' (',100*round(cur/tot,1),'%)\n',sep = '')
    load(paste0("p2r",did,".rdata")) # output of the function AnalyzeEKOAD
   out <- adjustMet1(p2d = EKOADwn[EKOADwn$id==did,], p2r = rslt[1:nse,], p2e = p2est, nsimul = nse, seed = 3343
    write.csv(out$final,file = paste0("final",did,p2est,".csv"),row.names = FALSE)
   write.csv(out$intermed,file = paste0("intermed",did,p2est,".csv"),row.names = FALSE)
   cur <- cur+1
 }
}


vdid <- c(6,10)
vp2est <- c("pip","pim1","pim2","pim2v2","pim3")
fa <- data.frame()
for (did in vdid)
{
 for (p2est in vp2est){
    f <- read.csv(paste0("final",did,p2est,".csv"))
    fn <- names(f)
    f$dsgn <- did
    f <- f[,c('dsgn',fn)]
    fa <- rbind(fa,f)
  }
}
write.csv(fa,file = "final_all.csv",row.names = FALSE)

## End(Not run)
```

adjustMet2 *Phase III sample size adjustment factor (Method 2).*

Description

adjustMet2 calculates the multiplicative ajustment factor ρ to be applied to Phase III sample size estimate using Method 2 proposed by Nhacolo and Brannath (in press).

Usage

```
adjustMet2(p2d, p2r, p2e, p2p0 = NULL, p2p1 = NULL, p2a = NULL,
  p2b = NULL, p3p0 = NULL, p3p1 = NULL, p3a = NULL, p3b = NULL,
  nsimul = 5000, seed = NULL, rhorange = c(0.5, 5), p3mpt = 0.001,
  rhot = 1e-04)
```

Arguments

p2d Dataframe with Phase II design, with similar as in EKOptAdaptDesigns.

`p2r` Dataframe containing results of Phase II trials following the design `p2d`. It is the output of the function `AnalyzeEKOAD`.

`p2e` Phase II estimate to consider among the estimates used by codeAnalyzeEKOAD. It can be `"pip"` (naive MLE) or one of the four estimates from methods proposed by Nhacolo and Brannath (2018): `"pim1"`, `"pim2"`, `"pim2v2"` or `"pim3"`.

`p2p0` Phase II response rate under H_0. If `NULL` (default), the value is taken `p2d`.

`p2p1` Phase II response rate under H_1. If `NULL` (default), the value is taken `p2d`.

`p2a` Phase II type I error rate. If `NULL` (default), the value is taken `p2d`.

`p2b` Phase II type II error rate. If `NULL` (default), the value is taken `p2d`.

`p3p0` Phase III response rate of the control group. If `NULL` (default), the value is set to `p2p0`.

`p3p1` Phase III response rate of the treatment group. If `NULL` (default), the value is set to `p2p1`.

`p3a` Phase III type I error rate. If `NULL` (default), the value is set `p2a`.

`p3b` hase III type II error rate. If `NULL` (default), the value is set `p2b`.

`nsimul` Number of (parametric) bootstrap samples (default 5000).

`seed` Seed for random number generator. If `NULL` (default), no seed is set.

`rhorange` A vector specifying a range to serach for ρ. The default is `c(0.5,5)`.

`p3mpt` Tolerated error margin for the power, i.e., maximum allowed absolute diference between the estimated expected power and the target. The default is 0.001.

`rhot` Search for ρ is interrupted and deem unsuccessful if the absolute difference between current and the previous is less than or equal to `rhot`.

Details

The aim of the adjustment is to get an adequately powered Phase III trial based on Phase II data. ρ is found using numerical rearch. See the documentation of the function `AnIItoIIIRe` for more details about the designs.

Value

A list containing two dataframes `final` and `intermed`. `final` contains the final measures for the adjustment factor (ρ), and for the unadjusted and adjusted power. `intermed` holds the intermediate results (of each bootstrap sample).

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Using Estimates from Adaptive Phase II Oncology Trials to Plan Phase III Trials. *In press*.

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

Ahn, C., Heo, M. and Zhang, S. *Sample Size Calculations for Clustered and Longitudinal Outcomes in Clinical Research*. CRC Press, 2014.

See Also

adjustMet2, SimulateEKOAD, AnalyzeEKOAD.

Examples

```
## Not run:
vdid <- c(6,10) # design ids
vp2est <- c("pip","pim1","pim2","pim2v2","pim3")
nse <- 1000#number of simulations for each phase
cur <- 1; tot <- length(vdid)*length(vp2est)
for (did in vdid){
 for (p2est in vp2est){
   cat('Processing ',cur,' of ',tot,' (',100*round(cur/tot,1),'%)\n',sep = '')
    load(paste0("p2r",did,".rdata")) # output of the function AnalyzeEKOAD
   out <- adjustMet2(p2d = EKOADwn[EKOADwn$id==did,], p2r = rslt[1:nse,], p2e = p2est, nsimul = nse, seed = 3343
    write.csv(out$final,file = paste0("final",did,p2est,".csv"),row.names = FALSE)
   write.csv(out$intermed,file = paste0("intermed",did,p2est,".csv"),row.names = FALSE)
   cur <- cur+1
 }
}


vdid <- c(6,10)
vp2est <- c("pip","pim1","pim2","pim2v2","pim3")
fa <- data.frame()
for (did in vdid)
{
 for (p2est in vp2est){
    f <- read.csv(paste0("final",did,p2est,".csv"))
    fn <- names(f)
    f$dsgn <- did
    f <- f[,c('dsgn',fn)]
    fa <- rbind(fa,f)
  }
}
write.csv(fa,file = "final_all.csv",row.names = FALSE)

## End(Not run)
```

AnalyzeEKOAD *Analyze simulated adaptive trials.*

Description

AnalyzeEKOAD performs inference on trials simulated by the function SimulateEKOAD using the methods proposed by Nhacolo and Brannath (2018) and naive maximum likelihood.

Usage

```
AnalyzeEKOAD(replicates = NULL, basedir = NULL)
```

Arguments

`replicates`	Number of simulated trials to be analysed. If `NULL` (default), all trials found in `./basedir/SimulatedTrials` are analysed.
`basedir`	The base directory containing the sub-directory `SimulatedTrials` with the simulated trials. If `NULL` (default), the current working directory is uded.

Details

Overall p-values, point estimates and confidence intervals are calculated.

Value

A dataframe with the results. A copy is saved in the file `Results.csv` in the `basedir`.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

`SimulateEKOAD`, `mue1`, `mue2`, `mue2v2`, `mue3`.

`AnalyzePerformanceSimon` *Performance of estimation methods*

Description

It takes the results produced by `AnalyzeSimonDsgn` and `AnalyzeSimonDsgnAdaptN` and produces a dataframe containing bias, mean square error and variance of the estimators. It also calculates the power and the expected sample size (EN) where applicable.

Usage

```
AnalyzePerformanceSimon(designs = "all", basedir = NA)
```

Arguments

`designs`	Taking values `"fixed"`, `"adaptive"` or `"all"`, indicating whether only classical, adaptive or all designs should be included. The default is `"all"`.
`basedir`	The root directory in which simulations were performed. The current working directory is assumed by default. It must contain all the files and folders created by `SimulateSimonDsgn` and/or `SimulateSimonDsgnAdaptN`.

Details

Computations are done for different combinations of values of stop, (0,1), and success, (0,1). See AnalyzeSimonDsgn or AnalyzeSimonDsgnAdaptN. For instance, computations done on all simulated trials are marked with "both" in the columns stop and success, while the ones done only on trials that continued to the final stage have stop = "no" and success = "both".

Value

Dataframe containing bias, mean square error and variance of the estimators, power, expected sample size, and design information.

Author(s)

Arsenio Nhacolo

See Also

AnalyzeSimonDsgn, AnalyzeSimonDsgnAdaptN, pdata and AnalyzePerformanceSimon2.

Examples

```
## Not run:
AnalyzePerformanceSimon()

## End(Not run)
```

AnalyzePerformanceSimon2 *Performance of estimation methods*

Description

It takes the results produced by AnalyzeSimonDsgn and AnalyzeSimonDsgnAdaptN and produces a dataframe containing bias, mean square error and variance of the estimators. It also calculates the power and the expected sample size (EN) where applicable.

Usage

```
AnalyzePerformanceSimon2(designs = "all", basedir = NA)
```

Arguments

designs	Taking values "fixed", "adaptive" or "all", indicating whether only classical, adaptive or all designs should be included. The default is "all".
basedir	The root directory in which simulations were performed. The current working directory is assumed by default. It must contain all the files and folders created by SimulateSimonDsgn and/or SimulateSimonDsgnAdaptN.

Details

It is the same as AnalyzePerformanceSimon, but here the estimation is done only for two sets: all trials (unconditional), and only trials that continued to final stage (conditional).

Value

Dataframe containing bias, mean square error and variance of the estimators, power, expected sample size, and design information.

Author(s)

Arsenio Nhacolo

See Also

AnalyzeSimonDsgn, AnalyzeSimonDsgnAdaptN, pdata and AnalyzePerformanceSimon.

Examples

```
## Not run:
AnalyzePerformanceSimon2()

# Simulation example
seed = 1986
p0 <- 0.1
alpha <- 0.05
beta <- 0.1
repl <- 100 # number of replicated trials for each p
if (file.exists("PerforAll.csv")) unlink("PerforAll.csv")
coln <- TRUE
while (p0 < 0.5){
  pv <- seq(p0+0.2,p0+0.4,0.1) # p to simulate data
  p1v <- seq(p0+0.2,p0+0.3,0.1) # p to get design
  for (p1 in p1v){
    designParam <- CalculateSimonDsgn(p0, p1, alpha, beta)
    pstart <- p0+0.1
    SimulateSimonDsgn(repl, designParam, pstart, seed = seed)
    SimulateSimonDsgnAdaptN(repl, designParam, pstart, seed = seed)
    AnalyzeSimonDsgn()
    AnalyzeSimonDsgnAdaptN()
    perf <- AnalyzePerformanceSimon2()
    for (p in pv){
      SimulateSimonDsgn(repl, designParam, p, seed = seed)
      SimulateSimonDsgnAdaptN(repl, designParam, p, seed = seed)
      AnalyzeSimonDsgn()
      AnalyzeSimonDsgnAdaptN()
      perf <- rbind(perf, AnalyzePerformanceSimon2())
    }
    write.csv(perf, file = paste("PerforAll_a",alpha,"b",beta,"p0",p0,"p1",
                                 p1,".csv", sep = ""), row.names = F)
   write.table(perf, file ="PerforAll.csv", append = T, sep = ",", row.names = F, col.names = coln)
    coln <- FALSE
  }
  p0 <- p0+0.1
}

## End(Not run)
```

AnalyzeSimonDsgn *Analysis of simulated Simon's design trials*

Description

Analyses the trials simulated by SimulateSimonDsgn.

Usage

```
AnalyzeSimonDsgn(replicates = NA, basedir = NA)
```

Arguments

replicates	Number of trials to be analysed. By default all simulated trials are analysed.
basedir	The root directory in which simulations were performed. The current working directory is assumed by default. It must contain all the files and folders created by SimulateSimonDsgn.

Details

In addition to hypothesis testing, the response rate is estimated using different estimators: pm, pg, pu, pp and pk.

Value

Creates two data files in basedir containing results for optimal (*ResultsOptimalDesign.csv*) and minimax (*ResultsMinimaxDesign.csv*). The files contain a trial ID, stage 1, stage 2 and overall number of successful responses, s1, s2 and s, sample sizes (equal to those pre-specified by design), n1, n2 and n, and critical values, r1 and r. p0 the response rate assumed under H_0 and dsgnp1 under H_1. p1 is the true response rate (used for generating trial data). pm1 and pm2 are, respectively, pm based only of stage 1 and stage 2 data. stop indicates whether the trial stopped at first stage (stop = 1), and success indicates whether H_0 was rejected (success = 1).

Author(s)

Arsenio Nhacolo

See Also

CalculateSimonDsgn, SimulateSimonDsgn, AnalyzePerformanceSimon and AnalyzeSimonDsgnAdaptN.

Examples

```
AnalyzeSimonDsgn()
```

AnalyzeSimonDsgnAdaptN *Analysis of simulated adaptive Simon's design trials*

Description

Analyses the trials simulated by SimulateSimonDsgnAdaptN.

Usage

```
AnalyzeSimonDsgnAdaptN(replicates = NA, basedir = NA)
```

Arguments

replicates	Number of trials to be analysed. By default all simulated trials are analysed.
basedir	The root directory in which simulations were performed. The current working directory is assumed by default. It must contain all the files and folders created by SimulateSimonDsgnAdaptN.

Details

In addition to hypothesis testing, the response rate is estimated using different estimators: pm, pg, pu, pp and pk. The overall critical value, r, is recalculated using conditional type I error (*Englert and Kieser, 2012*).

Value

Creates two data files in basedir containing results for optimal (*ResultsOptimalDesignAdapt.csv*) and minimax (*ResultsMinimaxDesignAdapt.csv*). The files contain a trial ID, stage 1, stage 2 and overall number of successful responses, s1, s2 and s, sample sizes (equal to those pre-specified by design), n1, n2 and n, and critical values, r1 and r. p0 the response rate assumed under H_0 and dsgnp1 under H_1. p1 is the true response rate (used for generating trial data). pm1 and pm2 are, respectively, pm based only of stage 1 and stage 2 data. stop indicates whether the trial stopped at first stage (stop = 1), and success indicates whether H_0 was rejected (success = 1).

Author(s)

Arsenio Nhacolo

See Also

CalculateSimonDsgn, SimulateSimonDsgnAdaptN, AnalyzePerformanceSimon and AnalyzeSimonDsgn.

Examples

```
AnalyzeSimonDsgnAdaptN()
```

AnIItoIIIRe *Use of Phase II estimates to plan Phase III sample size.*

Description

AnIItoIIIRe calculates the power in a Phase III equal-size group two-arm randomized clinical trial with a binary response planned using estimates from Phase II adaptive two-stage trial.

Usage

```
AnIItoIIIRe(rslt, f = c(0.95, 0.96, 0.97, 0.98, 0.99))
```

Arguments

rslt	Dataframe containing the output from the function AnalyzeEKOAD, but with only successful trials (rslt$suco==1), i.e., trials in which H_0 was rejected.
f	Vector of length 5 containing multiplicative ajustment factors to be applied to Phase II estimates. The default is f = c(.95,.96,.97,.98,.99).

Details

The sample size (N) of the Phase III trial is based on the estimates naive MLE and estimators proposed by Nhacolo and Brannath (2018). Different values of retention factor f proposed by Kirby et al. (2012) are applied.The control group response rate is considered to be equal to that under the null hypothesis of the Phase II design, and the hypothesized treatment group response rate considered to be equal to that estimated from the Phase II trial. The target type I error and power are the same as of the Phase II design.Two-sided hypothesis test is assumed.N is a sample size per group, and equal size groups are assume. Hence, N total is 2*N. When calculating the power, the true response rate (in treatment group) is considered to be the one under which the Phase II trial was simulated (spi1).

Value

The input dataframe with corresponding Phase III sample size and power.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Using Estimates from Adaptive Phase II Oncology Trials to Plan Phase III Trials. *In press*.

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

Ahn, C., Heo, M. and Zhang, S. *Sample Size Calculations for Clustered and Longitudinal Outcomes in Clinical Research*. CRC Press, 2014.

See Also

AnalyzeEKOAD, SimulateEKOAD, PerforIItoIIIRe.

aop1 *Overall p-value (Method 1 of Nhacolo and Brannath, 2018).*

Description

aop1 calculates the overall p-value for adaptive two-stage designs with binary endpoint using the Method 1 (see Nhacolo and Brannath, 2018).

Usage

```
aop1(dsgn, x1o, xo, verbose = TRUE)
```

Arguments

dsgn	Dataframe containing one of the designs in EKOADwn.
x1o	The observed stage 1 number of responses.
xo	The total observed number of responses.
verbose	If TRUE (default) messages will be printed.

Details

This is one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

p-value.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

aop2, aop2v2, aop3e.

aop1e *Overall p-value for CI (Method 1 of Nhacolo and Brannath, 2018).*

Description

aop1e is a modified version of aop1 used for getting the confidence interval.

Usage

```
aop1e(dsgn, x1o, xo, newpi0)
```

Arguments

dsgn	Dataframe containing one of the designs in EKOADwn.
x1o	The observed stage 1 number of responses.
xo	The total observed number of responses.
newpi0	New response probability that replaces the one under the null hypothesis.

Details

This is one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

p-value.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

aop2e, aop2ev2, aop3e, aop1.

aop2 — *Overall p-value (Method 2 of Nhacolo and Brannath, 2018).*

Description

aop2 calculates the overall p-value for adaptive two-stage designs with binary endpoint using the Method 2 (see Nhacolo and Brannath, 2018).

Usage

```
aop2(dsgn, x1o, xo, verbose = TRUE)
```

Arguments

dsgn	Dataframe containing one of the designs in EKOADwn.
x1o	The observed stage 1 number of responses.
xo	The total observed number of responses.
verbose	If TRUE (default) messages will be printed.

Details

This is one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

p-value.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

aop1, aop2v2, aop3e

aop2e *Overall p-value for CI (Method 2 of Nhacolo and Brannath, 2018).*

Description

aop2e is a modified version of aop2 used for getting the confidence interval.

Usage

```
aop2e(dsgn, x1o, xo, newpi0)
```

Arguments

dsgn	Dataframe containing one of the designs in EKOADwn.
x1o	The observed stage 1 number of responses.
xo	The total observed number of responses.
newpi0	New response probability that replaces the one under the null hypothesis.

Details

This is one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

p-value.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

aop1e, aop2ev2, aop3e, aop2.

aop2ev2 *Overall p-value for CI (Method 2v2 of Nhacolo and Brannath, 2018).*

Description

`aop2ev2` is a modified version of `aop2v2` used for getting the confidence interval.

Usage

```
aop2ev2(dsgn, x1o, xo, newpi0)
```

Arguments

`dsgn`	Dataframe containing one of the designs in `EKOADwn`.
`x1o`	The observed stage 1 number of responses.
`xo`	The total observed number of responses.
`newpi0`	New response probability that replaces the one under the null hypothesis.

Details

This is one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

p-value.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

`aop1e`, `aop2e`, `aop3e`, `aop2v2`.

aop2v2 *Overall p-value (Method 2v2 of Nhacolo and Brannath, 2018).*

Description

aop2v2 calculates the overall p-value for adaptive two-stage designs with binary endpoint using the Method 2v2 (see Nhacolo and Brannath, 2018).

Usage

```
aop2v2(dsgn, x1o, xo, verbose = TRUE)
```

Arguments

dsgn	Dataframe containing one of the designs in EKOADwn.
x1o	The observed stage 1 number of responses.
xo	The total observed number of responses.
verbose	If TRUE (default) messages will be printed.

Details

This is one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

p-value.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

aop1, aop2

aop3e *Overall p-value (Method 3 of Nhacolo and Brannath, 2018).*

Description

aop3e calculates the overall p-value for adaptive two-stage designs with binary endpoint using the Method 3 (see Nhacolo and Brannath, 2018).

Usage

```
aop3e(dsgn, x1o, xo, newpi0 = NULL)
```

Arguments

dsgn	Dataframe containing one of the designs in EKOADwn.
x1o	The observed stage 1 number of responses.
xo	The total observed number of responses.
newpi0	New response probability that replaces the one under the null hypothesis. Omit it if the intention is only to calculate the overall p-value.

Details

This is one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

p-value.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

aop1, aop1e, aop2, aop2e, aop2v2, aop2ev2

CalculateSimonDsgn *Simon's designs*

Description

CalculateSimonDsgn finds Simon's optimal and minimax designs.

Usage

```
CalculateSimonDsgn(p0, p1, alpha, beta, verbose = TRUE)
```

Arguments

p0	The response rate under the null hypothesis.
p1	The response rate under the alternative hypothesis.
alpha	Type I error rate.
beta	Type II error rate.
verbose	If TRUE (default) the designs are printed (gives messy printout when the function is run without assigment).

Details

Simon's designs are two-stage single-arm for phase II clinical trials. They consist in first stage and overall sample sizes and critical values, n1 and n, and r1 and r, respectively.

Value

A two-row dataframe containing the optimal and the minimax designs.

Author(s)

Arsenio Nhacolo

References

Simon, R. Optimal two-stage designs for phase II clinical trials. *Control Clin Trials*, 1989, 10, 1-10.

See Also

SimulateSimonDsgn and SimulateSimonDsgnAdaptN.

Examples

```
d <- CalculateSimonDsgn(0.2, 0.4, 0.05, 0.1)
```

checkMonoDCF *Check the monotonicity of the sample space ordering.*

Description

checkMonoDCF checks the monotonicity of the sample space ordering defined based on inverse normal combination function (see Nhacolo and Brannath, 2018).

Usage

```
checkMonoDCF(d, verbose = TRUE)
```

Arguments

d	Dataframe containing one of the designs in EKOADwn.
verbose	If TRUE (default) messages about monotonicity will be printed.

Details

The monotonicity is with respect to the stage 2 number of successes.

Value

A list containing a dataframe (mono) with detailed info, and a logical variable notmono indicating wether non-monotonicity was concluded.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

Examples

```
## Not run:
#Check for all Englert and Kieser designs
notmov <- c()
for (i in 1:max(EKOADwn$id)){
  notmov <- c(notmov,checkMonoDCF(EKOADwn[EKOADwn$id==1,],verbose=FALSE)[[2]])
}
isMonotone <- !any(notmov);isMonotone

## End(Not run)
```

ci1 *Confidence interval (using Method 1 of Nhacolo and Brannath, 2018).*

Description

ci1 computes confidence interval.

Usage

```
ci1(dsgn, x1o, xo, alpha = 0.05, twosided = FALSE)
```

Arguments

dsgn	Dataframe containing one of the designs in EKOADwn.
x1o	The observed stage 1 number of responses.
xo	The total observed number of responses.
alpha	The significance level.
twosided	If FALSE (default) a one-sided CI is produced.

Details

This CI is obtained using the Method 1, one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

CI is a list with lower and upper bounds.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

ci2, ci2v2, ci3, aop1, aop1e, pipv1, , mue1.

ci2 *Confidence interval (using Method 2 of Nhacolo and Brannath, 2018).*

Description

`ci2` computes confidence interval.

Usage

```
ci2(dsgn, x1o, xo, alpha = 0.05, twosided = FALSE)
```

Arguments

`dsgn`	Dataframe containing one of the designs in `EKOADwn`.
`x1o`	The observed stage 1 number of responses.
`xo`	The total observed number of responses.
`alpha`	The significance level.
`twosided`	If `FALSE` (default) a one-sided CI is produced.

Details

This CI is obtained using the Method 2, one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

CI is a list with lower and upper bounds.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

`ci1`, `ci2v2`, `ci3`, `aop2`, `aop2e`, `pipv2`, `mue2`.

ci2v2 *Confidence interval (using Method 2v2 of Nhacolo and Brannath, 2018).*

Description

ci2v2 computes confidence interval.

Usage

```
ci2v2(dsgn, x1o, xo, alpha = 0.05, twosided = FALSE)
```

Arguments

dsgn	Dataframe containing one of the designs in EKOADwn.
x1o	The observed stage 1 number of responses.
xo	The total observed number of responses.
alpha	The significance level.
twosided	If FALSE (default) a one-sided CI is produced.

Details

This CI is obtained using the Method 2v2, one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

CI is a list with lower and upper bounds.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

ci1, ci2, ci3, aop2v2, aop2ev2, pipv2v2, , mue2v2.

ci3 *Confidence interval (using Method 3 of Nhacolo and Brannath, 2018).*

Description

`ci3` computes confidence interval.

Usage

```
ci3(dsgn, x1o, xo, alpha = 0.05, twosided = FALSE)
```

Arguments

`dsgn`	Dataframe containing one of the designs in `EKOADwn`.
`x1o`	The observed stage 1 number of responses.
`xo`	The total observed number of responses.
`alpha`	The significance level.
`twosided`	If `FALSE` (default) a one-sided CI is produced.

Details

This CI is obtained using the Method 3, one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

CI is a list with lower and upper bounds.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

`ci1`, `ci2`, `ci2v2`, `aop3e`, `pipv3`, `mue3`.

dsgnPrep *Pre-process the Englert and Kieser (2013) optimal adaptive designs.*

Description

dsgnPrep takes Englert and Kieser's optimal adaptive design and adds information that is needed by other functions.

Usage

```
dsgnPrep(dsgn = NULL, w1 = "n", w2 = NULL)
```

Arguments

dsgn	Dataframe containing one of the designs in EKOptAdaptDesigns.
w1, w2	Stage 1 and 2 weights. If w1="n" (default), weights a calculated based on stage-wise sample sizes as described in Nhacolo and Brannath (2018). If w1="sr2", then w1=w2=1/sqrt(2).

Details

The function adds, to each x1 leading to 2nd stage, the corresponding p-value (p1) and its backwards image (p1B), the stage-wise weights w1 and w2 and other information used in inference methods proposed by Nhacolo and Brannath (2018).

Value

Dataframe containing the input dataframe with added information.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

Examples

```
## Not run:
#Designs with w1a and w2 calculated based on sample sizes
EKOADwn <- data.frame()
for (j in 1:max(EKOptAdaptDesigns$id)){
 EKOADwn <- rbind(EKOADwn, dsgnPrep(dsgn = EKOptAdaptDesigns[EKOptAdaptDesigns$id==j,],w1 = "n"))
}
save(EKOADwn,file = "EKOADwn.RData")

## End(Not run)
```

EKOADwn *Pre-processed Englert and Kieser (2013)'s optimal adaptive designs.*

Description

A dataframe containing all the designd in EKOptAdaptDesigns pre-processed by the function dsgnPrep, the the argument w1 set to "n".

Usage

```
EKOADwn
```

Format

A dataframe with 709 rows and 20 variables.

EKOptAdaptDesigns *Englert and Kieser (2013)'s optimal adaptive designs.*

Description

A dataframe containing all optimal adaptive two-stage designs for phase II cancer clinical trials present in Englert and Kieser (2013).

Usage

```
EKOptAdaptDesigns
```

Format

A dataframe with 709 rows and 11 variables:

id Identifier of the designs

x1 Number of successes (responses) at stage 1

n2 Stage 2 sample size

D Discrete conditional error function

l Stage 2 decision boundary

pi0 Response probability under the null hypothesis

pi1 Response probability under the alternative hypothesis

alpha Type I error rate

beta Type II error rate

n1 Stage 1 sample size

n2max Maximum stage 2 sample size

Source

https://onlinelibrary.wiley.com/doi/abs/10.1002/bimj.201200220 getN2v2 *Number of patients to be enrolled in the second stage*

Description

Calculates the number of patients which should be enrolled in the second stage if the conditional power should be altert to "cp". It's a version of getN2.

Usage

```
getN2v2(cp, p1, design, k, mode = 0, alpha = 0.05)
```

Arguments

cp	conditional power to which the number of patients for the second stage should be adjusted.
p1	response probability under the alternative hypothesis.
design	a dataframe containing all critical values for a Simon's two-stage design defined by the columns r1, n1, r, n and p0. • r1 = critical value for the first stage (more than r1 responses needed to proceed to the second stage). • n1 = number of patients enrolled in the first stage. • r = critical value for the whole trial (more than r responses needed at the end of the study to reject the null hypothesis). • n = number of patients enrolled in the whole trial. • p0 = response probability under the null hypothesis.
k	number of responses observed at the interim analysis.
mode	a value out of 0,1,2,3 dedicating the methode spending the "rest alpha" (difference between nominal alpha level and actual alpha level for the given design). • 0 = "rest alpha" is not used. • 1 = "rest alpha" is spent proportionally. • 2 = "rest alpha" is spent equally. • 3 = "rest alpha" is spent only to the worst case scenario (minimal number of responses at the interim analysis so that the study can proceed to the second stage).
alpha	overall significance level the trial was planned for.

Details

This functon is the same as getN2 (OneArmPhaseTwoStudy package), with some changes in arguments' validation. It's is a helper to SimulateSimonDsgnAdaptN.

References

Englert S., Kieser M. Adaptive designs for single-arm phase II trials in oncology. *Pharm Stat*, 2012, 11, 241-249.

See Also

getN2, SimulateSimonDsgnAdaptN.

Examples

```
designParam <- CalculateSimonDsgn(0.2, 0.4, 0.05, 0.1)
dsgn <- designParam[designParam$Type == "Optimal",]
getN2v2(0.9, dsgn$p1, dsgn, 7)
```

mue1 *Median estimate (using Method 1 of Nhacolo and Brannath, 2018).*

Description

mue1 calculates the median estimate of the response rate.

Usage

```
mue1(dsgn, x1o, xo)
```

Arguments

dsgn	Dataframe containing one of the designs in EKOADwn.
x1o	The observed stage 1 number of responses.
xo	The total observed number of responses.

Details

This estimate is obtained using the Method 1, one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

Median estimate of response probability.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

mue2, mue2v2, mue3, aop1, aop1e, pipv1.

mue2 *Median estimate (using Method 2 of Nhacolo and Brannath, 2018).*

Description

mue2 calculates the median estimate of the response rate.

Usage

```
mue2(dsgn, x1o, xo)
```

Arguments

dsgn	Dataframe containing one of the designs in EKOADwn.
x1o	The observed stage 1 number of responses.
xo	The total observed number of responses.

Details

This estimate is obtained using the Method 2, one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

Median estimate of response probability.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

mue1, mue2v2, mue3, aop2, aop2e, pipv2.

mue2v2 — *Median estimate (using Method 2v2 of Nhacolo and Brannath, 2018).*

Description

`mue2v2` calculates the median estimate of the response rate.

Usage

```
mue2v2(dsgn, x1o, xo)
```

Arguments

`dsgn`	Dataframe containing one of the designs in `EKOADwn`.
`x1o`	The observed stage 1 number of responses.
`xo`	The total observed number of responses.

Details

This estimate is obtained using the Method 2v2, one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

Median estimate of response probability.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

`mue1`, `mue2`, `mue3`, `aop2v2`, `aop2ev2`, `pipv2v2`.

mue3 *Median estimate (using Method 3 of Nhacolo and Brannath, 2018).*

Description

mue3 calculates the median estimate of the response rate.

Usage

```
mue3(dsgn, x1o, xo)
```

Arguments

dsgn	Dataframe containing one of the designs in EKOADwn.
x1o	The observed stage 1 number of responses.
xo	The total observed number of responses.

Details

This estimate is obtained using the Method 3, one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

Median estimate of response probability.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

mue1, mue2, mue2v2, aop3e, pipv3.

Nct *Sample size per group for single-stage parallel-group RCT.*

Description

Nct calculates sample size for one group in an equal-size group two-arm randomized clinical trial with a binary response.

Usage

```
Nct(pc, pt, alp = 0.05, pow = 0.8)
```

Arguments

pc	Response probability in control group.
pt	Response probability in treatment group.
alp	Significance level (default: 0.05).
pow	Power (default: 0.8)

Details

The sample size is for one group (arm), double the number to get the total.

Value

Sample size for one group.

Author(s)

Arsenio Nhacolo

References

Ahn, C., Heo, M. and Zhang, S. *Sample Size Calculations for Clustered and Longitudinal Outcomes in Clinical Research.* CRC Press, 2014.

See Also

Pwr.

Examples

```
Nct(0.2,0.3,0.05,0.9)
```

pdata *Helper function for analyzing the performance of estimators*

Description

It takes the results produced by AnalyzeSimonDsgn or AnalyzeSimonDsgnAdaptN and produces a dataframe containing bias, mean square error and variance of the estimators.

Usage

```
pdata(t, design, stop, success, replicates)
```

Arguments

t	Dataframe containing results produced by AnalyzeSimonDsgn or AnalyzeSimonDsgnAdaptN.
stop	Taking value "yes", "no" or "both", indicating that only trials that stopped, continued or both were analyzed.
success	Taking value "yes", "no" or "both", indicating that only trials that were successful, unsuccessful or both were analyzed.
replicates	Number of trials analysed. It is equal to the number of rows in t.

Details

It is a helper function for AnalyzePerformanceSimon. It also calculates the power and the expected sample size (EN) where applicable.

Value

Dataframe containing bias, mean square error and variance of the estimators.

Author(s)

Arsenio Nhacolo

See Also

AnalyzeSimonDsgn, AnalyzeSimonDsgnAdaptN and AnalyzePerformanceSimon.

Examples

```
## Not run:
rslt <- read.csv("ResultsOptimalDesign.csv")
nrep <- nrow(rslt)
t <- rslt
presult <- pdata(t, "Optimal", "both", "both", nrep)
t <- rslt[rslt$stop == 0,]
presult <- rbind(presult, pdata(t, "Optimal", "no", "both", nrep))

## End(Not run)
```

pdata2 *Helper function for analyzing the performance of estimators*

Description

pdata2 is a helper function used by function PerformanceEKOAD.

Usage

```
pdata2(t, stop, success, replicates)
```

Details

Not to be used directly.

Value

Dataframe

Author(s)

Arsenio Nhacolo

PerforIItoIIIRe *Performance, with respect to Phase III power, of phase II estimates.*

Description

PerforIItoIIIRe calculates the mean and median power in a Phase III trials from the output of AnIItoIIIRe.

Usage

```
PerforIItoIIIRe(t)
```

Arguments

t	Dataframe containing the output from the function AnIItoIIIRe.

Value

Dataframe.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Using Estimates from Adaptive Phase II Oncology Trials to Plan Phase III Trials. *In press*.

See Also

AnIItoIIIRe.

Examples

```
## Not run:
rslt <- read.csv("ResultsAll.csv")
rsltfull <- rslt
rslt <- rslt[rslt$suco==1,]
rslt <- rslt[,c("pi0", "pi1", "spi1", "alpha", "beta", "suco", "pip",
                "pim1", "pim2", "pim2v2", "pim3")]
rslt <- rslt[rslt$spi1>=rslt$pi0+0.1 & rslt$spi1<=rslt$pi1+0.3,]
rslt$spi1f <- factor(rslt$spi1)
cats <- levels(rslt$spi1f)
ncats <- length(cats)
setwd(paste0("C:/Users/arsenio/Documents/PhD/Simulations/Paper2/Reuse/pi01by0.01/50000/",did))
save(ncats,file = "ncats.rdata")
for (i in 1:ncats){
  sr <- rslt[rslt$spi1f==cats[i],]#Single result (result of a specific spi1)
  save(sr,file = paste0("sr",i,".rdata"))
}
load("ncats.rdata")
PerfAll <- data.frame()
for (k in 1:ncats){
 load(paste0("sr",k,".rdata"))
 sre <- AnIItoIIIRe(rslt = sr,f = c(.95,.96,.97,.98,.99))
  PerfAll <- rbind(PerfAll,PerforIItoIIIRe(sre))
  rm(sr)
}
write.csv(PerfAll, file = "PerfAllIItoIII.csv", row.names = F)

## End(Not run)
```

PerformanceEKOAD *Performance of estimation methods*

Description

PerformanceEKOAD calculates performance measures (bias, mean square error, coverage probability) of the estimation methods based on the results produced by AnalyzeEKOAD.

Usage

```
PerformanceEKOAD(basedir = NULL)
```

Arguments

basedir	The base directory containing the file with the results (Results.csv). If NULL (default), the current working directory is uded.

Value

A dataframe with the performance results. A copy is saved in the file the PerformanceResults.csv in the basedir.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

SimulateEKOAD, AnalyzeEKOAD.

Examples

```
## Not run:
#SIMULATIONS
for (did in c(6,10)){#Design ID
cat("============================ Design ",did," ============================\n")
repl <- 50000 # number of replicated trials for each p
dir.create(as.character(did))
setwd(as.character(did))
design <- EKOADwn[EKOADwn$id==did,]
seed = 3343
if (file.exists("PerforAll.csv")) unlink("PerforAll.csv")
piv <- seq(0,1,0.025) # p to simulate data
resul <- data.frame()
perf <- data.frame()
k <- 0
pl <- length(piv)
for (pi in piv){
  k <- k+1
 cat("_________________________ pi = ",pi, " (",k," of ",pl,") _________________________\n",sep = "")
  SimulateEKOAD(replicates = repl, dsgn = design, newpi1 = pi, seed = seed)
  resul <- rbind(resul, AnalyzeEKOAD())
  perf <- rbind(perf, PerformanceEKOAD())
}
write.table(resul, file ="ResultsAll.csv", sep = ",", row.names = F, col.names = TRUE)
write.table(perf, file ="PerforAll.csv", sep = ",", row.names = F, col.names = TRUE)
cat("Design ID: ", design$id[1], "\nReplicates: ", repl, "\nSeed: ", seed,
    "\nDate last run: ", date(),file = "info.txt", sep = "", append = FALSE)
}

## End(Not run)
```

---

pg *Bias-reduced estimator*

Description

Calculates the bias-reduced estimator of the true response rate as proposed by *Guo and Liu (2005)*.

Usage

```
pg(s, n1, r1, n)
```

Arguments

`s` Total number of successes.

`n1` Stage 1 sample size.

`r1` Stage 1 critical value (trial is stopped at stage 1 if the number of successes is at most `r1`).

`n` Total sample size.

Details

It uses bias subtraction, with bias calculated by `sbias` and response rate estimated by `pm`.

Value

Estimate of the response rate.

Author(s)

Arsenio Nhacolo

References

Guo, H. Y. and Liu, A. A simple and efficient bias-reduced estimator of response probability following a group sequential phase II trial. *J Biopharm Stat*, 2005, 15, 773-781.

See Also

`sbias`, `pm`, `pu`, `pp` and `pk`.

Examples

```
pg(21, 19, 4, 54)
```

`pipv1` *Response rate to attain a specified p-value (using Method 1 of Nhacolo and Brannath, 2018).*

Description

`pipv1` finds the response probability under the null hypotheisis that, given the observed data, would yield a desired overall p-value.

Usage

```
pipv1(dsgn, x1o, xo, pv)
```

Arguments

`dsgn` Dataframe containing one of the designs in `EKOADwn`.

`x1o` The observed stage 1 number of responses.

`xo` The total observed number of responses.

`pv` The desired p-value.

Details

The p-value is obtained using the Method 1, one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

Response probability.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

`pipv2`, `pipv2v2`, `pipv3`, `aop1`, `aop1e`.

`pipv2` *Response rate to attain a specified p-value (using Method 2 of Nhacolo and Brannath, 2018).*

Description

`pipv2` finds the response probability under the null hypotheisis that, given the observed data, would yield a desired overall p-value.

Usage

```
pipv2(dsgn, x1o, xo, pv)
```

Arguments

`dsgn`	Dataframe containing one of the designs in `EKOADwn`.
`x1o`	The observed stage 1 number of responses.
`xo`	The total observed number of responses.
`pv`	The desired p-value.

Details

The p-value is obtained using the Method 2, one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

Response probability.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

`pipv1`, `pipv2v2`, `pipv3`, `aop2`, `aop2e`.

`pipv2v2` *Response rate to attain a specified p-value (using Method 2v2 of Nhacolo and Brannath, 2018).*

Description

`pipv2v2` finds the response probability under the null hypotheisis that, given the observed data, would yield a desired overall p-value.

Usage

```
pipv2v2(dsgn, x1o, xo, pv)
```

Arguments

`dsgn`	Dataframe containing one of the designs in `EKOADwn`.
`x1o`	The observed stage 1 number of responses.
`xo`	The total observed number of responses.
`pv`	The desired p-value.

Details

The p-value is obtained using the Method 2v2, one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

Response probability.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

pipv1, pipv2, pipv3, aop2v2, aop2ev2.

---

pipv3 *Response rate to attain a specified p-value (using Method 3 of Nhacolo and Brannath, 2018).*

Description

pipv3 finds the response probability under the null hypotheisis that, given the observed data, would yield a desired overall p-value.

Usage

```
pipv3(dsgn, x1o, xo, pv)
```

Arguments

dsgn	Dataframe containing one of the designs in EKOADwn.
x1o	The observed stage 1 number of responses.
xo	The total observed number of responses.
pv	The desired p-value.

Details

The p-value is obtained using the Method 3, one of the four methods proposed by Nhacolo and Brannath (2018) primarily for single-arm adaptive two-stage group sequential designs with a binary endpoint.

Value

Response probability.

Author(s)

Arsenio Nhacolo

References

Nhacolo, A. and Brannath, W. Interval and point estimation in adaptive Phase II trials with binary endpoint. *Stat Methods Med Res*, 2018.

See Also

pipv1, pipv2, pipv2v2, aop3e.

pk *Median unbiased estimator*

Description

Calculates the median unbiased estimator of true response rate for for Simon-like designs.

Usage

```
pk(s, n1, r1, n, p0)
```

Arguments

s	Total number of successes.
n1	Stage 1 sample size.
r1	Stage 1 critical value (trial is stopped at stage 1 if the number of successes is at most r1).
n	Total sample size.
p0	Response rate under the null hypothesis.

Details

Median unbiased estimator is the value of response rate such that the p-value is 0.5 (*Koyama and Chen, 2008*). The solution is found using numerical search, with a precision of 0.000001.

Value

Estimate of the response rate.

Author(s)

Arsenio Nhacolo

References

Koyama, T. and Chen, H. Proper inference from Simon's two-stage designs. *Stat Med*, 2008, 27, 3145-3154.

See Also

pvaluek, pquantile, pm, pg, pu and pp.

Examples

```
pk(21, 19, 4, 54, 0.2)
```

`pm` *Sample proportion*

Description

Calculates the sample proportion.

Usage

```
pm(s, n)
```

Arguments

`s`	Total number of successes.
`n`	Total sample size.

Details

For fixed designs the sample propotion is an unbiased (maximum likelihood) estimator of the response rate, but in group sequential designs (e.g., Simon's) it is biased.

Value

Estimate of the response rate.

Author(s)

Arsenio Nhacolo

See Also

`pg`, `pu`, `pp` and `pk`.

Examples

```
pm(21, 54)
```

`pp` *UMVCUE*

Description

Calculates the uniformly minimum variance conditionally unbiased estimator (UMVCUE) of the true response probability.

Usage

```
pp(s, n1, r1, n)
```

Arguments

`s`	Total number of successes.
`n1`	Stage 1 sample size.
`r1`	Stage 1 critical value (trial is stopped at stage 1 if the number of successes is at most `r1`).
`n`	Total sample size.

Details

The UMVCUE (*Pepe et al., 2009*) is conditional on on proceeding to the second stage. he sample proportion is used when the trial stopped at first stage.

Value

Estimate of the response rate.

Author(s)

Arsenio Nhacolo

References

Pepe, M. S.; Feng, Z.; Longton, G. and Koopmeiners, J. Conditional estimation of sensitivity and specificity from a phase 2 biomarker study allowing early termination for futility. *Stat Med*, 2009, 28, 762-779.

See Also

`pm`, `pg`, `pu` and `pk`.

Examples

```
pp(21, 19, 4, 54)
```

`pquantile` *Value of response rate to attain a given p-value*

Description

Finds, for Simon-like designs, the value of response probability that would yield a given p-value.

Usage

```
pquantile(s, n1, r1, n, p0, pvalue)
```

Arguments

`s` Total number of successes.

`n1` Stage 1 sample size.

`r1` Stage 1 critical value (trial is stopped at stage 1 if the number of successes is at most `r1`).

`n` Total sample size.

`p0` Response rate under the null hypothesis.

`pvalue` The desired p-value.

Details

The solution is found using numerical search, with a precision of 0.000001. The p-value is as defined by *Koyama and Chen (2008)*.

Value

Response probability.

Author(s)

Arsenio Nhacolo

References

Koyama, T. and Chen, H. Proper inference from Simon's two-stage designs. *Stat Med*, 2008, 27, 3145-3154.

See Also

`pvaluek` and `pk`.

Examples

```
pquantile(21, 19, 4, 54, 0.2, 0.5)
```

`pu` *UMVUE*

Description

Calculates the uniformly minimum variance unbiased estimator (UMVUE) of the true response probability.

Usage

```
pu(s, n1, r1, n)
```

Arguments

`s`	Total number of successes.
`n1`	Stage 1 sample size.
`r1`	Stage 1 critical value (trial is stopped at stage 1 if the number of successes is at most `r1`).
`n`	Total sample size.

Details

The UMVUE is based on approach by *Grishick et al. (1946)*. It was first considered by *Chang et al. (1989)* and further studied by *Jung et al. (2004)*.

Value

Estimate of the response rate.

Author(s)

Arsenio Nhacolo

References

Jung, S.-H. and Kim, K. M. On the estimation of the binomial probability in multistage clinical trials. *Stat Med*, 2004, 23, 881-896.

See Also

`pm`, `pg`, `pp` and `pk`.

Examples

```
pu(21, 19, 4, 54)
```

`pvaluek` *P-value*

Description

Calculates p-value for Simon-like designs.

Usage

```
pvaluek(s, n1, r1, n, p0)
```

Arguments

`s`	Total number of successes.
`n1`	Stage 1 sample size.
`r1`	Stage 1 critical value (trial is stopped at stage 1 if the number of successes is at most `r1`).
`n`	Total sample size.
`p0`	Response rate under the null hypothesis.

Details

It is based on the definition of p-value by *Koyama and Chen (2008)*.

Value

P-value.

Author(s)

Arsenio Nhacolo

References

Koyama, T. and Chen, H. Proper inference from Simon's two-stage designs. *Stat Med*, 2008, 27, 3145-3154.

See Also

pquantile and pk.

Examples

```
pvaluek(21, 19, 4, 54, 0.2)
```

Pwr *Power for single-stage parallel-group RCT.*

Description

Pwr calculates the power in an equal-size group two-arm randomized clinical trial with a binary response.

Usage

```
Pwr(pc, pt, Nc, alp = 0.05)
```

Arguments

pc	Response probability in control group.
pt	Response probability in treatment group.
Nc	Sample size per group.
alp	Significance level (default: 0.05).

Value

Sample size for one group.

Author(s)

Arsenio Nhacolo

References

Ahn, C., Heo, M. and Zhang, S. *Sample Size Calculations for Clustered and Longitudinal Outcomes in Clinical Research.* CRC Press, 2014.

See Also

`Nct`.

Examples

```
Pwr(0.2,0.3,389,0.05)
```

`sbias` *Bias of the sample proportion*

Description

Calculates bias due to using sample proportion as estimator of the true response rate.

Usage

```
sbias(n1, r1, n, p)
```

Arguments

`n1`	Stage 1 sample size.
`r1`	Stage 1 critical value (trial is stopped at stage 1 if the number of successes is at most `r1`).
`n`	Total sample size.
`p`	True success probability.

Details

For fixed designs the sample propotion is an unbiased (maximum likelihood) estimator of the response rate, but in group sequential designs (e.g., Simon's) it is biased.

Value

Bias.

Author(s)

Arsenio Nhacolo

References

Porcher, R. and Desseaux, K. What inference for two-stage phase II trials? *BMC Med Res Methodol*, 2012, 12, 117.

See Also

`sfms` and `pg`.

Examples

```
sbias(19, 4, 54, 0.4)
```

`sfms` *Probability mass function of* `(M, S)`

Description

Probability mass function of `M` (stage) and `S` (number of successes).

Usage

```
sfms(s, n1, r1, n, p, m = NA)
```

Arguments

`s`	Total number of successes.
`n1`	Stage 1 sample size.
`r1`	Stage 1 critical value (trial is stopped at stage 1 if the number of successes is at most `r1`).
`n`	Total sample size.
`p`	True success probability.
`m`	Stage number (1 or 2). It is automatically determined based on `s` and `r1`, therefore it shouldn't be provided, unless there are reasons to do so.

Details

Probability mass function of the statistic `(M, S)` for Simon-like designs (allowing early stopping for futility only).

Value

Density.

Author(s)

Arsenio Nhacolo

References

Jung, S.-H. and Kim, K. M. On the estimation of the binomial probability in multistage clinical trials. *Stat Med*, 2004, 23, 881-896.

See Also

`sbias` and `pg`.

Examples

```
sfms(21, 19, 4, 54, 0.4)
```

SimulateEKOAD *Simulate single-arm binary endpoint two-stage adaptive designs.*

Description

SimulateEKOAD Simulate trials following designs similar to that of Englert and Kieser(2013)'s.

Usage

```
SimulateEKOAD(replicates, dsgn, newpi1 = NULL, seed = NULL,
  deleteOld = TRUE)
```

Arguments

replicates	Number of trials to be simulated.
dsgn	Dataframe containing one of the designs in EKOADwn.
newpi1	New response rate under the alternative hypothesis used to simulate trials. If NULL (default), the one from the design is used.
seed	The seed for random number generator. If NULL (default), no seed is set and , hence, results are not reproducible.
deleteOld	If TRUE (default), the simulation sub-directory is cleared before simulations start.

Details

The original designs (like the ones in EKOptAdaptDesigns) must be pre-processed using the function dsgnPrep to get extra information like the designs in EKOADwn.

Value

Simulated trials are saved in the sub-directory ./SimulatedTrials.

Author(s)

Arsenio Nhacolo

References

Englert, S. and Kieser, M. Optimal adaptive two-stage designs for phase II cancer clinical trials. *Biometrical Journal*, 2013.

See Also

EKOptAdaptDesigns, EKOADwn.

SimulateSimonDsgn *Simon's designs data simulation*

Description

SimulateSimonDsgn simulates data from Simon's optimal and minimax designs.

Usage

```
SimulateSimonDsgn(replicates, designParam, newp1 = NA, seed = NA,
  deleteOld = TRUE)
```

Arguments

replicates	Number of trials to be generated.
designParam	A dataframe containing Simon's optimal and minimax designs, as returned by the function CalculateSimonDsgn.
newp1	If NA (default) data are generated assuming the same response probability under alternative hypothesis, p1, used to get the designs (see CalculateSimonDsgn). One may provide different values of newp1 if there is interest in studying the effect of departure from the design's assumed p1.
seed	Initial value (any integer) of random-number seed. It is useful for creating simulations that can be reproduced. The default is NA, meaning no reproducibility.
deleteOld	If TRUE (default) the sub-directories /Optimal/SimulatedTrials and /Minimax/SimulatedTrial are deleted, if they exist, before simulation starts. The old data files are still replaced by the new ones even if deleteOld is set to FALSE, but some old files remain in cases where the previous replicates was greater that the current one.

Details

The simulated trials are stored in the sub-directories /Optimal/SimulatedTrials and /Minimax/SimulatedTrials for optimal and minimax designs, repectively, under the current working directory. The subdirectories are automatically created. Individual trial data are stored in a CSV file named trial#, where # is the replicate number.

Value

The function is not intended to return an R object, instead it creates files (in CSV format) containing simulated trials data. See *Details*. It also saves in the current working directory the designParam argument (*DesignParameters.csv*).

Author(s)

Arsenio Nhacolo

See Also

CalculateSimonDsgn, SimulateSimonDsgnAdaptN and AnalyzeSimonDsgn.

Examples

```
d <- CalculateSimonDsgn(0.2, 0.4, 0.05, 0.1)
SimulateSimonDsgn(100, d, seed = 1986)
```

`SimulateSimonDsgnAdaptN` *Simon's adaptive designs data simulation*

Description

Simulates data from adaptive versions of Simon's optimal and minimax designs, propposed by *Englert and Kieser (2012).* Adaptation consists in recalculating the second stage sample size n2 in order to achieve a desired conditional power given the number of successes at first stage.

Usage

```
SimulateSimonDsgnAdaptN(replicates, designParam, newp1 = NA,
  condPwr = NA, restAlphaMet = 0, seed = NA, deleteOld = TRUE)
```

Arguments

`replicates` Number of trials to be generated.

`designParam` A dataframe containing Simon's optimal and minimax designs, as returned by the function `CalculateSimonDsgn`.

`newp1` If `NA` (default) data are generated assuming the same response probability under alternative hypothesis, `p1`, used to get the designs (see `CalculateSimonDsgn`). One may provide different values of `newp1` if there is interest in studying the effect of departure from the design's assumed `p1`.

`condPwr` The desired conditional power. The default is `1-beta`.

`restAlphaMet` The method for spending the "rest alpha" (difference between nominal alpha level and actual alpha level for the given design).

- 0: "rest alpha" is not used (default);
- 1: "rest alpha" is spent proportionally;
- 2: "rest alpha" is spent equally;
- 3: "rest alpha" is spent only to the worst case scenario (minimal number of responses at the interim analysis so that the study can proceed to the second stage).

`seed` Initial value (any integer) of random-number seed. It is useful for creating simulations that can be reproduced. The default is `NA`, meaning no reproducibility.

`deleteOld` If TRUE (default) the sub-directories `/OptimalAdapt/SimulatedTrials` and `/MinimaxAdapt/SimulatedTrials` are deleted, if they exist, before simulation starts. The old data files are still replaced by the new ones even if `deleteOld` is set to `FALSE`, but some old files remain in cases where the previous `replicates` was greater that the current one.

Details

The simulated trials are stored in the sub-directories /OptimalAdapt/SimulatedTrials and /MinimaxAdapt/Simulate for optimal and minimax designs, repectively, under the current working directory. The sub-directories are automatically created. Individual trial data are stored in a CSV file named trial#, where # is the replicate number.

Value

The function is not intended to return an R object, instead it creates files (in CSV format) containing simulated trials data. See *Details*. It also saves in the current working directory the designParam argument (*DesignParametersAdapt.csv*).

Author(s)

Arsenio Nhacolo

References

Englert S., Kieser M. Adaptive designs for single-arm phase II trials in oncology. *Pharm Stat*, 2012, 11, 241-249.

See Also

CalculateSimonDsgn, getN2, SimulateSimonDsgn and AnalyzeSimonDsgnAdaptN.

Examples

```
d <- CalculateSimonDsgn(0.2, 0.4, 0.05, 0.1)
SimulateSimonDsgnAdaptN(100, d, seed = 1986)
```

Index