

Package: dani (via r-universe)

October 26, 2024

Type Package

Title Design and Analysis of Non-Inferiority Trials

Version 1.0-0

Date 2024-10-04

Author Matteo Quartagno

Maintainer Matteo Quartagno <m.quartagno@ucl.ac.uk>

Description Provides tools to help with the design and analysis of non-inferiority trials. These include functions for doing sample size calculations and for analysing non-inferiority trials, using a variety of outcome types and population-level summary measures. It also features functions to make trials more resilient by using the concept of non-inferiority frontiers, as described in Quartagno et al. (2019) <[arXiv:1905.00241](https://arxiv.org/abs/1905.00241)>. Finally it includes function to design and analyse MAMS-ROCI (aka DURATIONS) trials.

License GPL-2

Imports boot, survival, graphics, DescTools, ratesci, exact2x2, margineffects, mfp, tibble, BSDA, coxphw, survRM2, flexsurv, numDeriv, mratios, DescrTab2, exact2x2, contingencytables, PropCIs, pracma

Suggests knitr, rmarkdown

VignetteBuilder knitr

Repository <https://matteo21q.r-universe.dev>

RemoteUrl <https://github.com/matteo21q/dani>

RemoteRef HEAD

RemoteSha 5f589f310d17095dd655790d0255b3f6a5e0c865

Contents

compare.NIfrontier.binary	2
compare.NIfrontier.continuous	3

compare.NIfrontier.binary	5
convertmargin.binary	6
convertmargin.continuous	7
convertmargin.binary	8
plot.ROCI	10
power.NI.binary	11
power.NI.continuous	12
power.NIfrontier.binary	14
power.ROCI.binary	15
samplesize.NI.binary	17
samplesize.NI.continuous	19
samplesize.NI.binary	21
samplesize.NIfrontier.binary	22
samplesize.ROCI.binary	23
summary.ROCI	26
test.NI.binary	27
test.NI.continuous	29
test.NI.binary	31
test.NIfrontier.binary	33
test.ROCI.binary	35
type1error.NIfrontier.binary	37

Index 39

compare.NIfrontier.binary

Comparing Non-Inferiority Frontiers for trials with binary outcomes

Description

A function to compare non-inferiority frontiers corresponding to designing a trial with binary outcomes using different summary measures, i.e. risk difference, risk ratio or odds ratio. The function plots the frontiers for the specified design parameters and ranks them in terms of power for a fixed sample size.

Usage

```
compare.NIfrontier.binary(p.control.expected, p.experim.target=NULL,
                          p.range, NI.margin, summary.measure="RD")
```

Arguments

- p.control.expected
The expected control event risk.
- p.experim.target
The target experimental arm event risk at which to power the trial.
- p.range
The range of risks around the expected control event risk to investigate. These should be plausible risks in case the assumed control event risk was uncertain.

NI.margin	The Non-Inferiority margin on the specified population-level summary measure. This is automatically converted for the specified control event risk to the other summary measures for comparison.
summary.measure	The summary measure on which the NI margin is specified. Can be either "RD" (Risk Difference), "OR" (Odds Ratio) or "RR" (Risk Ratio).

Details

This function compares various non-inferiority frontiers corresponding to each population-level summary measure. First, the frontiers for risk difference, risk ratio and odds ratio margins with the specified design parameters are plotted. Then, the frontiers are ranked in terms of power for a fixed sample size. This function can be used to choose a summary measure for a non-inferiority trial maximising power before performing proper sample size calculations.

Value

A data.frame with the coordinates of the NI frontiers used in the plot and the distances of each frontier from the trial expected point. A figure comparing the frontiers is plotted and the ranking of the different frontiers is printed on screen.

Examples

```
p0.expected<-0.05 # Expected control event risk
p1.alt<-p0.expected # Same as expected control event risk
NI.m<-0.05
range.of.p0<-c(0.005,0.10)
```

```
compare.NIfrontier.binary(p.control.expected=p0.expected, p.experim.target=p1.alt, p.range=range.of.p0, NI.marg
```

```
compare.NIfrontier.continuous
```

Comparing Non-Inferiority Frontiers for trials with continuous outcomes

Description

A function to compare non-inferiority frontiers corresponding to designing a trial with continuous outcomes using different summary measures, i.e. mean difference or mean ratio. The function plots the frontiers for the specified design parameters and ranks them in terms of power for a fixed sample size.

Usage

```
compare.NIfrontier.continuous(mean.control.expected, mean.experim.target=NULL,
                              mean.range, NI.margin, summary.measure="difference")
```

 compare.Nifrontier.survival

Comparing Non-Inferiority Frontiers for trials with survival outcomes

Description

A function to compare non-inferiority frontiers corresponding to designing a trial with survival outcomes using different summary measures, i.e. hazard ratio, difference in restricted mean survival time or difference in surviving proportion. The function plots the frontiers for the specified design parameters and ranks them in terms of power for a fixed sample size.

Usage

```
compare.Nifrontier.survival(rate.control.expected=NULL, rate.experim.target=NULL,
  t.expected=NULL, p.control.expected=NULL, p.experim.target=NULL,
  rates.range=NULL, NI.margin, summary.measure="HR", tau.RMST,
  t.DS=tau.RMST)
```

Arguments

- `rate.control.expected`
The expected control event rate assuming an exponential distribution of event times. Can be omitted if event risks at `t.expected` are passed as arguments.
- `rate.experim.target`
The target experimental arm event rate in the alternative hypothesis used for powering the trial. This assumes an exponential distribution for event times within arm and can be omitted if `p.experim.target` at `t.expected` are passed as arguments.
- `t.expected`
The time at which the expected event risk in the control arm is provided, assuming exponential event distribution. Can be omitted if event rates are passed as arguments. The default is `t.DS`, the time at which the difference in surviving proportion is to be evaluated as summary measure.
- `p.control.expected`
The expected control event risk at time `t.expected`. Can be omitted if event rates are passed as arguments.
- `p.experim.target`
The experimental arm event risk at time `t.expected` in the alternative hypothesis used for powering the trial. Can be omitted if event rates are passed as arguments.
- `rates.range`
The range of event rates to investigate around the expected control event rate. This assumes exponential distribution of event times. These should be plausible rates in case the assumed control event risk was uncertain. Defaults to 0.3 below and above `rate.control.expected`, or the rate corresponding to `p.control.expected` at `t.expected`.

NI.margin	The Non-Inferiority margin on the specified population-level summary measure. This is automatically converted for the specified control event rate to the other summary measures for comparison, assuming exponential distributions within arm.
summary.measure	The summary measure on which the NI margin is specified. Can be either "HR" (Hazard Ratio), "DRMST" (Difference in Restricted Mean Survival Time) or "DS" (Difference in Surviving proportions).
tau.RMST	Horizon time tau at which difference in RMST should be used as a summary measure.
t.DS	Horizon time t at which difference in surviving proportion should be used as a summary measure. Defaults to same as tau.RMST.

Details

This function compares various non-inferiority frontiers corresponding to each population-level summary measure. First, the frontiers for hazard ratio, difference in restricted mean survival time and difference in surviving proportion margins with the specified design parameters are plotted. Then, the frontiers are ranked in terms of power for a fixed sample size. This function can be used to choose a summary measure for a non-inferiority trial maximising power before performing proper sample size calculations. Note that, for the moment, this assumes exponential survival times within arm and proportional hazards.

Value

A data.frame with the coordinates of the NI frontiers used in the plot and the distances of each frontier from the trial expected point. A figure comparing the frontiers is plotted and the ranking of the different frontiers is printed on screen.

Examples

```
set.seed(1)
out1<-compare.NIfrontier.survival(rate.control.expected = 0.3, rate.experim.target=0.3, NI.margin=1.2, summary.measure="HR")
set.seed(1)
out2<-try(compare.NIfrontier.survival(p.control.expected = 0.8, t.expected=3, NI.margin=0.16, summary.measure="DS"))
```

convertmargin.binary *Function to convert non-inferiority margin for binary outcome between summary measures.*

Description

Function to convert non-inferiority margin between different summary measures with a binary outcome. Supported summary measures are risk difference (RD), risk ratio (RR), odds ratio (OR) and arcsine difference (AS).

Usage

```
convertmargin.binary(p.control.expected, NI.margin.original,
                    summary.measure.original, summary.measure.target)
```

Arguments

`p.control.expected`
The expected control event risk.

`NI.margin.original`
The non-inferiority margin on the original scale, i.e. using the original summary measure.

`summary.measure.original`
The original summary measure on which the non-inferiority margin is specified. One of "RD" (risk difference), "RR" (risk ratio), "OR" (odds ratio) or "AS" (arcsine difference).

`summary.measure.target`
The summary measure in which we want to convert the non-inferiority margin. One of "RD" (risk difference), "RR" (risk ratio), "OR" (odds ratio) or "AS" (arcsine difference).

Value

This function returns the non-inferiority margin in the target summary measure.

Examples

```
# Example: RD to AS
p0.expected<-0.05
NI.margin.RD<-0.05
NI.margin.AS<-convertmargin.binary(p0.expected, NI.margin.RD,"RD","AS")
# Example: RR to RD
NI.margin.RR<-2
NI.margin.RD<-convertmargin.binary(p0.expected, NI.margin.RR,"RR","RD")
# Example: RD to OR
NI.margin.OR<-convertmargin.binary(p0.expected, NI.margin.RD,"RD","OR")
```

convertmargin.continuous

Function to convert non-inferiority margin for continuous outcome between summary measures.

Description

Function to convert non-inferiority margin between different summary measures with a continuous outcome. Supported summary measures are mean difference ("difference") and ratio ("ratio").

Usage

```
convertmargin.continuous(mean.control.expected, NI.margin.original,
                        summary.measure.original, summary.measure.target)
```

Arguments

`mean.control.expected`
The assumed mean in the control arm.

`NI.margin.original`
The non-inferiority margin on the original scale, i.e. using the original summary measure.

`summary.measure.original`
The original summary measure on which the non-inferiority margin is specified. One of "difference" (mean difference) or "ratio" (mean ratio).

`summary.measure.target`
The summary measure in which we want to convert the non-inferiority margin. One of "difference" (mean difference) or "ratio" (mean ratio).

Value

This function returns the non-inferiority margin in the target summary measure.

Examples

```
# Example: difference to ratio
m0.expected<-2
NI.margin.diff<-1
NI.margin.rat<-convertmargin.continuous(m0.expected,
    NI.margin.diff, "difference", "ratio")
# Example: ratio to difference
NI.margin.diff<-convertmargin.continuous(m0.expected,
    NI.margin.rat, "ratio", "difference")
```

```
convertmargin.survival
```

Function to convert non-inferiority margin for survival outcome between summary measures.

Description

Function to convert non-inferiority margin between different summary measures with a survival outcome. Supported summary measures are hazard ratio (HR), difference in restricted mean survival time (DRMST) and difference in surviving proportion (DS).

Usage

```
convertmargin.survival(rate.control.expected=NULL, t.expected=NULL,
  p.control.expected=NULL, NI.margin.original,
  summary.measure.original, summary.measure.target,
  tau.RMST=NULL, t.DS=NULL, BH.est="exponential", S.control=NULL)
```

Arguments

rate.control.expected	The expected control event rate. This is the rate parameter of an exponential distribution for event times in the control arm. Note this can be avoided if passing t.expected and p.control.expected.
t.expected	The time at which the control event risk (p.control.expected) is provided. Note this can be avoided if passing rate.control.expected.
p.control.expected	The expected control event risk at t.expected. Note this can be avoided if passing rate.control.expected.
NI.margin.original	The non-inferiority margin on the original scale, i.e. using the original summary measure.
summary.measure.original	The original summary measure on which the non-inferiority margin is specified. One of "HR" (hazard ratio), "DRMST" (difference in restricted mean survival time) or "DS" (difference in surviving proportion).
summary.measure.target	The summary measure in which we want to convert the non-inferiority margin. One of "RD" (risk difference), "RR" (risk ratio), "OR" (odds ratio) or "AS" (arcsine difference).
tau.RMST	The horizon time tau for RMST.
t.DS	The horizon time t for difference in surviving proportion.
BH.est	How should the baseline hazard be estimated in the conversion of the margin? "exponential" assumes the event distribution is exponential, while "surv.func" uses a survival function fitted on some data (e.g. the control arm data from the trial).
S.control	The survival function in the control arm. This is necessary if using BH.est="surv.func".

Value

This function returns the non-inferiority margin in the target summary measure.

Examples

```
# Example: HR to DRMST
p0.expected<-0.1
t.ex<-3
NI.margin.HR<-2
NI.margin.DRMST<-convertmargin.survival(t.expected=t.ex,
```

```

    p.control.expected=p0.expected,
    NI.margin.original=NI.margin.HR, summary.measure.original="HR",
    summary.measure.target="DRMST", tau.RMST=3)
# Example: DS to HR
NI.margin.DS<-0.05
NI.margin.HR<-convertmargin.survival(t.expected=t.ex,
  p.control.expected=p0.expected,
  NI.margin.original=NI.margin.DS, summary.measure.original="DS",
  summary.measure.target="HR", t.DS=3)
# Example: DS to DRMST
NI.margin.DRMST<-convertmargin.survival(t.expected=t.ex,
  p.control.expected=p0.expected,
  NI.margin.original=NI.margin.DS, summary.measure.original="DS",
  summary.measure.target="DRMST",tau.RMST=3, t.DS=3)

```

plot.ROCI

Plot treatment-response curve estimated with a ROCI trial

Description

A function to plot the results of the analysis of a ROCI randomised trial analysed with test.ROCI.binary.

Usage

```

## S3 method for class 'ROCI'
plot(x,type="summary.measure", ylim=NULL, pch=15,
      xlab = "Treatment level", ylab=NULL,
      lwd=3, ...)

```

Arguments

x	A list obtained as an output from function test.ROCI.binary.
type	Type of plot. Can be either "tr.curve" or "summary.measure". "tr.curve" plots the treatment response
ylim	the y limits of the plot.
pch	Either an integer specifying a symbol or a single character to be used as the default in plotting points.
xlab	a label for the x axis, defaults to "Treatment level".
ylab	a label for the y axis, defaults to a description of y.
lwd	A vector of line widths. Defaults to 3.
...	Other graphical parameters

Details

This is a function to plot results of a call to `test.ROCI.binary`. Two different types of plot are possible: with `type="tr.curve"`, the average estimate treatment-response curve is provided, with a red line indicating the acceptability curve and a red point indicating the optimal treatment level. If `type="summary.level"`, the estimated population-level summary measures are plotted, with associated confidence intervals around them. The optimal treatment level is painted red.

Examples

```
duration.arms=c(8,10,12,14,16,18,20)
sam.sizes=c(700)
NI.margin.RD<-0.1

durations<-rep(duration.arms, each=100)
y<-rbinom(sam.sizes,1,0.05+(20-durations)*0.01)

data.ex<-data.frame(y,durations)
myformula<-as.formula(y~treat(durations))

res1<-test.ROCI.binary(formula=myformula, data=data.ex,
                       se.method="delta", treatment.levels=8:20, summary.measure="RD",
                       NI.margin=NI.margin.RD)

plot(res1, type="tr.curve")
plot(res1, type="summary.measure")
```

power.NI.binary	<i>Function to compute the power for a given sample size for a standard 2-arm non-inferiority trial.</i>
-----------------	--

Description

A function that can be used to do power calculations for a non-inferiority trial with binary outcome. The trial can aim to use one of several possible analysis methods and summary measures.

Usage

```
power.NI.binary(p.control.expected, p.experim.target, NI.margin, sig.level = 0.025,
               n.control, n.experim, summary.measure = "RD", print.out = TRUE, test.type=NULL,
               unfavourable=T, n.rep=1000, M.boot=2000, BB.adj=0.0001)
```

Arguments

`p.control.expected`
Expected event risk in the control arm.

`p.experim.target`
Target event risk in the experimental arm under which to power the trial.

NI.margin	Non-inferiority margin. Can be either risk difference, risk ratio, odds ratio or arc-sine difference.
sig.level	One-sided significance level for testing. Default is 0.025, i.e. 2.5%.
n.control	The sample size in the control arm for which to estimate power.
n.experim	The sample size in the experimental arm for which to estimate power.
summary.measure	The population-level summary measure to be estimated, i.e. the scale on which we define the non-inferiority margin. Can be one of "RD" (Risk Difference), "RR" (Risk Ratio), "OR" (Odds Ratio) or "AS" (Arc-Sine difference).
print.out	Logical. If FALSE, no output is printed.
test.type	A string that indicates the type of test to be assumed for the sample size calculation. Currently, three options are supported: "Wald", "score" and "local".
unfavourable	A logical variable. If TRUE, the outcome is considered unfavourable. This is used to check that the NI margin specified is meaningful.
n.rep	The number of repetitions of the simulations to estimate power.
M.boot	Number of bootstrap samples if using a bootstrap-based analysis method.
BB.adj	Adjustment factor for "Berger.Boos" method.

Details

This function estimates power of a standard two-arm non-inferiority trial for a given sample size, running a certain number n.rep of simulations under the alternative hypothesis and calculating estimated success rate of the trial with the desired design and analysis methods.

Value

The estimated power. On screen, the Monte-Carlo Confidence Interval is printed as well.

Examples

```
power<-power.NI.binary(p.control.expected=0.2, p.experim.target=0.2, NI.margin=0.1, n.control=200, n.experim=200,
n.rep=500)
```

power.NI.continuous *Power calculation tool for Non-Inferiority trials with a continuous outcome.*

Description

A function for estimating power of a non-inferiority trial whose primary outcome is continuous. Allows for different summary measures, test types and both favourable (e.g. cure) and unfavourable (e.g. death) events.

Usage

```
power.NI.continuous(mean.control, mean.experim, sd, NI.margin, sig.level = 0.025,
                    n.control, n.experim, summary.measure = "mean.difference", print.out = TRUE,
                    test.type=NULL, higher.better=T, M.boot=2000, n.rep=1000)
```

Arguments

mean.control	Assumed mean in the control arm.
mean.experim	target mean in the experimental arm for powering the trial.
sd	Standard deviation in both arms.
NI.margin	Non-inferiority margin. Can be either defined as a mean difference, or mean ratio.
sig.level	One-sided significance level for testing. Default is 0.025, i.e. 2.5%.
n.control	The sample size in the control arm for which to estimate power.
n.experim	The sample size in the experimental arm for which to estimate power.
summary.measure	The population-level summary measure to be estimated, i.e. the scale on which we define the non-inferiority margin. Can be one of "mean.difference" (Difference of Means) or "mean.ratio" (Ratio of Means).
print.out	Logical. If FALSE, no output is printed.
higher.better	Logical. If FALSE, the outcome is considered unfavourable, i.e. higher scores indicate worse outcomes. Default is TRUE, i.e. favourable outcome, higher scores indicate better outcomes.
test.type	A character string defining the method to be used for sample size calculations. For the mean difference, methods available include "Z.test" and "t.test". For the mean ratio, methods available include "Fiellers" test, "log.t.test" or "log.Z.test".
n.rep	The number of repetitions of the simulations to estimate power.
M.boot	Number of bootstrap samples if using a bootstrap-based analysis method.

Details

This is a function to estimate through simulations the power of a fixed sample size to test non-inferiority of an active treatment against the control within a specific NI margin. The margin can be specified on a number of different scales.

Value

The output is an estimate of power, and on screen the Monte-Carlo CI for this estimate may be printed as well.

Examples

```
power<-power.NI.continuous(mean.control=2, mean.experim=2, sd=1, NI.margin=-1, n.control=20, n.experim=20, n.rep=1000)
```

power.NIfrontier.binary

Function to compute the power for a given sample size for a 2-arm non-inferiority trial designed using non-inferiority frontiers.

Description

A function that can be used to do power calculations for a non-inferiority trial with binary outcome designed using non-inferiority frontiers. The trial can aim to use one of several possible analysis methods and summary measures.

Usage

```
power.NIfrontier.binary(p.control.expected, p.experim.target=NULL, NI.frontier, sig.level=0.025,
  summary.measure="RD", print.out=TRUE, unfavourable=TRUE,
  n.control, n.experim, n.rep=1000, M.boot=2000, BB.adj=0.0001,
  test.type="LRT")
```

Arguments

p.control.expected	Expected event risk in the control arm.
p.experim.target	Target event risk in the experimental arm under which to power the trial.
NI.frontier	Non-inferiority frontier, a function whos eonly input should be the control event risk and that returns the NI margin for that risk expressed as the specified summary measure.
sig.level	One-sided significance level for testing. Default is 0.025, i.e. 2.5%.
n.control	The sample size in the control arm for which to estimate power.
n.experim	The sample size in the experimental arm for which to estimate power.
summary.measure	The population-level summary measure to be estimated, i.e. the scale on which we define the non-inferiority margin. Can be one of "RD" (Risk Difference), "RR" (Risk Ratio), "OR" (Odds Ratio) or "AS" (Arc-Sine difference).
print.out	Logical. If FALSE, no output is printed.
test.type	A string that indicates the type of test to be assumed for the sample size calculation. Currently, three options are supported: "Wald", "score" and "local".
unfavourable	A logical variable. If TRUE, the outcome is considered unfavourable. This is used to check that the NI margin specified is meaningful.
n.rep	The number of repetitions of the simulations to estimate power.
M.boot	Number of bootstrap samples if using a bootstrap-based analysis method.
BB.adj	Adjustment factor for "Berger.Boos" method.

Details

This function estimates power of a two-arm non-inferiority trial designed using non-inferiority frontiers for a given sample size, running a certain number `n.rep` of simulations under the alternative hypothesis and calculating estimated success rate of the trial with the desired design and analysis methods.

Value

The estimated power. On screen, the Monte-Carlo Confidence Interval is printed as well.

Examples

```
NI.f<-function(p) {
  marg<-ifelse(p<0.15,p, 0.15)
  return(marg)
}
power<-power.NIfrontier.binary(p.control.expected=0.2, p.experim.target=0.2, NI.frontier=NI.f, n.control=200, n.
                             n.rep=500)
```

power.ROCI.binary	<i>Function to compute the power for a given sample size for a MAMS-ROCI (aka DURATIONS) randomised trial.</i>
-------------------	--

Description

A function that can be used to do power calculations for a MAMS-ROCI randomised trial with binary outcome. The trial can aim to use one of several possible analysis methods to find the minimum/maximum non-inferior arm.

Usage

```
power.ROCI.binary(p.expected.curve, NI.margin, reference=max(treatment.levels),
                  optimal=min(treatment.levels), range=optimal, unfavourable=T,
                  se.method="bootstrap", treatment.levels, summary.measure="RD",
                  tr.model="FP2.classic", M.boot=NULL, parallel="no", n.cpus=1, sig.level=0.025,
                  n.per.arm, print.out=T, simulations=FALSE, n.rep=1000, bootCI.type="bca")
```

Arguments

p.expected.curve	The expected event risks at the specified treatment levels. It should be a numeric vector of the same length as <code>treatment.levels</code> .
NI.margin	The non-inferiority margin(s). It can be either a scalar, if the margin is the same for all treatment levels, or a vector of length equal to that of <code>treatment.levels</code> minus one. For <code>summary.measure="target.risk"</code> , this represents the target probability that we want to achieve, and if given as a vector it has length equal to that of <code>treatment.levels</code> .

reference	The arm that should act as a reference. It must be one of the arms in treatment levels. Default is the largest value among treatment.levels.
optimal	The arm that should be considered as the optimal under the given data generating mechanism. It must be one of the arms in treatment levels. Default is the smallest value among treatment.levels.
range	The arm(s) that should be considered in the acceptable range for the range power method under the given data generating mechanism. It must be one or more of the arms in treatment levels. Default is the smallest value among treatment.levels (same as optimal).
unfavourable	If TRUE the outcome is taken to be an unfavourable event, e.g. death or relapse. If FALSE, the outcome is considered a positive one, like cure.
se.method	Method for calculating standard errors around the treatment-response curve. Can be either "bootstrap" or "delta".
treatment.levels	A vector with all the treatment levels to be investigated. These will generally correspond to the treatment arms.
summary.measure	The population-level summary measure that defines the estimand. It can be either "RD" (Risk Difference), "RR" (Risk Ratio), "target.risk" (target probability) or "OR" (Odds Ratio).
tr.model	Model to use for the treatment-response curve. It can be one of: "FP1.classic" or "FP2.classic" (Multivariable Fractional Polynomials as in package mfp, with a maximum of either 1 or 2 powers), "FP1.fixed" or "FP2.fixed" (Fractional polynomials fixing the number of powers to exactly 1 or 2).
M.boot	Number of bootstrap samples if using se.method="bootstrap".
parallel	If using bootstrap, this gives the chance to parallelise calculations. Use parallel="snow" with windows, or parallel="multicore" with Linux. See help page for the boot function.
n.cpus	If using bootstrap in parallel, this gives the number of cpus to parallelise computations on.
sig.level	One-sided significance level.
n.per.arm	The sample size (number of patients per arm) for which we want to compute power. It can either be a scalar, assuming same sample size in all treatment.levels, or a vector of length equal to treatment.levels.
print.out	If TRUE the estimated sample sizes are printed on screen.
simulations	A logical variable. If set to TRUE, power is estimated by running simulations. Otherwise, a modified version of the formula in samplesize.ROCI.binary is used. Default is FALSE.
n.rep	The number of repetitions to be used for the simulations if simulations=TRUE.
bootCI.type	Method for computing the confidence intervals if using se.method="bootstrap" and simulations=TRUE. It can be either "norm", "basic", "perc" or "bca". Default is "bca", which is the recommended option when possible.

Details

This function computes optimal, range and acceptable power for a ROCI trial with a given sample size. It requires as input the expected treatment response curve under which to power the trial, the NI margins and a specification of all the methods. It can estimate power either using a formula or simulations. If using formula, the function creates a data set of expected outcomes (e.g. if risk in one arm is 0.7, it creates 70 records with event=1 and 30 with event=0) and fits the model for the treatment-response curve on this data set. It then estimates the variance of the summary measure of interest and uses this to do a power calculation based on simple formula from normal theory for non-inferiority with a continuous outcome. If using simulations, the function runs n.rep simulations with the specified sample size and analysis methods. The function allows for the margin to be expressed as either a risk difference, risk ratio, odds ratio or a target probability.

Value

A list including computed optimal, range and acceptable power. Acceptable power is estimated as the maximum power with the given sample size for any of the treatment levels. The function also returns power for all specific treatment levels.

Examples

```
p.expected.curve=rep(0.05,7)
NI.margin=rep(0.05,6)
reference=20
optimal=8
range=c(8,10,12)
se.method="delta"
treatment.levels<-c(8,10,12,14,16,18,20)
summary.measure<-"RD"
tr.model="FP2.classic"
sig.level=0.05
unfavourable=T
```

```
ss<-power.ROCI.binary(p.expected.curve, NI.margin, reference=reference, optimal=optimal, range=range, unfavourable=unfavourable,
                      se.method=se.method, treatment.levels=treatment.levels, summary.measure=summary.measure,
                      tr.model=tr.model, sig.level = sig.level, n.per.arm = 150)
```

samplesize.NI.binary *Sample size calculation tool for Non-Inferiority trials with a binary outcome.*

Description

A function for calculating sample size of a non-inferiority trial whose primary outcome is binary. Allows for different summary measures, test types and both favourable (e.g. cure) and unfavourable (e.g. death) events.

Usage

```
samplesize.NI.binary(p.control.expected, p.experim.target, NI.margin,
                     sig.level = 0.025, power = 0.9, r = 1,
                     summary.measure = "RD", print.out = TRUE,
                     test.type="score", unfavourable=T, cont.corr=F,
                     round=T, ltfu=0)
```

Arguments

p.control.expected	Expected event risk in the control arm.
p.experim.target	Target event risk in the experimental arm under which to power the trial.
NI.margin	Non-inferiority margin. Can be either risk difference, risk ratio, odds ratio or arc-sine difference.
sig.level	One-sided significance level for testing. Default is 0.025, i.e. 2.5%.
power	Power of the trial, i.e. one minus type-II error of the study. Default is 0.9, i.e. 90%.
r	Allocation ratio, i.e. ratio between sample sizes in the active and control groups. Default is 1.
summary.measure	The population-level summary measure to be estimated, i.e. the scale on which we define the non-inferiority margin. Can be one of "RD" (Risk Difference), "RR" (Risk Ratio), "OR" (Odds Ratio) or "AS" (Arc-Sine difference).
print.out	Logical. If FALSE, no output is printed.
test.type	A string that indicates the type of test to be assumed for the sample size calculation. Currently, three options are supported: "Wald", "score" and "local".
unfavourable	A logical variable. If TRUE, the outcome is considered unfavourable. This is used to check that the NI margin specified is meaningful.
cont.corr	A logical variable. If TRUE, a continuity correction is applied. This is currently supported for the risk difference scale only.
round	A logical variable. If TRUE, sample sizes are rounded to the next integer, using the ceiling function. Otherwise, they are left unrounded.
ltfu	A numeric variable with the expected proportion of patients lost to follow-up. Default is 0.

Details

This is a function to calculate sample size needed to test non-inferiority of an active treatment against the control within a specific NI margin. The margin can be specified on a number of different scales.

Value

The output is a vector `ss`, containing the sample sizes for the control and active arms respectively.

Examples

```

p.control.expected<-0.05 # Expected control event rate
p.experim.target<-p.control.expected # Same as expected active event rate
NI.marg.RD<-0.05 # Non-inferiority margin on risk difference scale
NI.marg.RR<-NI.marg.OR<-2 # Non-inferiority margin on risk or odds ratio scale
# Non-inferiority margin on arc-sine difference scale:
  NI.marg.AS<-arcsine.margin(p.experim.target+NI.marg.RD, p.control.expected)

r<-1 # Allocation ratio
power<-0.9 # Power
alph<-0.025 # Significance level

samplesize.RD<-samplesize.NI.binary(sig.level=alph, power=power,
  p.control.expected=p.control.expected, p.experim.target=p.experim.target,
  NI.margin=NI.marg.RD, r=r) # Risk difference scale
samplesize.RR<-samplesize.NI.binary(sig.level=alph, power=power,
  p.control.expected=p.control.expected, p.experim.target=p.experim.target,
  NI.margin=NI.marg.RR, r=r, summary.measure="RR") # Log-risk ratio scale
samplesize.AS<-samplesize.NI.binary(sig.level=alph, power=power,
  p.control.expected=p.control.expected, p.experim.target=p.experim.target,
  NI.margin=NI.marg.AS, r=r, summary.measure="AS") # Arc-sine difference scale
samplesize.ORD<-samplesize.NI.binary(sig.level=alph, power=power,
  p.control.expected=p.control.expected, p.experim.target=p.experim.target,
  NI.margin=NI.marg.OR, r=r, summary.measure="OR") # Log-odds ratio scale

```

samplesize.NI.continuous

Sample size calculation tool for Non-Inferiority trials with a continuous outcome.

Description

A function for calculating sample size of a non-inferiority trial whose primary outcome is continuous. Allows for different summary measures, test types and both favourable (e.g. cure) and unfavourable (e.g. death) events.

Usage

```

samplesize.NI.continuous(mean.control, mean.experim, sd, NI.margin,
  sig.level = 0.025, power = 0.9, r = 1,
  summary.measure = "mean.difference", print.out = TRUE,
  test.type=NULL, higher.better=T,
  round=T, ltfu=0)

```

Arguments

mean.control Assumed mean in the control arm.

mean.experim	target mean in the experimental arm for powering the trial.
sd	Standard deviation in both arms.
NI.margin	Non-inferiority margin. Can be either defined as a mean difference, or mean ratio.
sig.level	One-sided significance level for testing. Default is 0.025, i.e. 2.5%.
power	Power of the trial, i.e. one minus type-II error of the study. Default is 0.9, i.e. 90%.
r	Allocation ratio, i.e. ratio between sample sizes in the active and control groups. Default is 1.
summary.measure	The population-level summary measure to be estimated, i.e. the scale on which we define the non-inferiority margin. Can be one of "mean.difference" (Difference of Means) or "mean.ratio" (Ratio of Means).
print.out	Logical. If FALSE, no output is printed.
higher.better	Logical. If FALSE, the outcome is considered unfavourable, i.e. higher scores indicate worse outcomes. Default is TRUE, i.e. favourable outcome, higher scores indicate better outcomes.
test.type	A character string defining the method to be used for sample size calculations. For the mean difference, methods available include "Z.test" and "t.test". For the mean ratio, methods available include "Fiellers" test, "log.t.test" or "log.Z.test".
round	A logical variable. If TRUE, sample sizes are rounded to the next integer, using the ceiling function. Otherwise, they are left unrounded.
ltfu	A numeric variable with the expected proportion of patients lost to follow-up. Default is 0.

Details

This is a function to calculate sample size needed to test non-inferiority of an active treatment against the control within a specific NI margin. The margin can be specified on a number of different scales.

Value

The output is a vector `ss`, containing the sample sizes for the control and active arms respectively.

Examples

```
out<-try(samplesize.NI.continuous(20, mean.experim=20, sd=40, NI.margin=-10,
  sig.level = 0.025, power = 0.9, r = 1,
  summary.measure = "mean.difference",
  print.out = TRUE, test.type="t.test",
  higher.better=T))
```

samplesize.NI.survival

Sample size calculation tool for Non-Inferiority trials with a survival outcome.

Description

A function for calculating sample size of a non-inferiority trial whose primary outcome is survival. Allows for different summary measures, test types and both favourable (e.g. cure) and unfavourable (e.g. death) events.

Usage

```
samplesize.NI.survival(HR.target=NULL, p.control.expected=NULL,
                       p.experim.target=NULL, NI.margin, sig.level = 0.025,
                       power = 0.9, r = 1, summary.measure = "HR",
                       print.out = TRUE, test.type="logrank.Schoenfeld",
                       unfavourable=T, round=T, ltfu=0)
```

Arguments

HR.target	The target hazard ratio between experimental and control arm at which to power the trial.
p.control.expected	Expected event risk in the control arm.
p.experim.target	Target event risk in the active arm at which to power the trial.
NI.margin	Non-inferiority margin on the selected summary measure.
sig.level	One-sided significance level for testing. Default is 0.025, i.e. 2.5%.
power	Power of the trial, i.e. one minus type-II error of the study. Default is 0.9, i.e. 90%.
r	Allocation ratio, i.e. ratio between sample sizes in the active and control groups. Default is 1.
summary.measure	The population-level summary measure to be estimated, i.e. the scale on which we define the non-inferiority margin. Can be one of "HR" (Hazard Ratio), "DRMST" (Difference in Restricted Mean Survival Time by time tau), or "DS" (Difference in Surviving proportion at time tau).
print.out	Logical. If FALSE, no output is printed.
test.type	A string that indicates the type of test to be assumed for the sample size calculation. Currently, three options are supported for "HR" only: "logrank.Freedman", "logrank.Schoenfeld", "KM".
unfavourable	A logical variable. If TRUE, the outcome is considered unfavourable. This is used to check that the NI margin specified is meaningful.

round	A logical variable. If TRUE, sample sizes are rounded to the next integer, using the ceiling function. Otherwise, they are left unrounded.
ltfu	A numeric variable with the expected proportion of patients lost to follow-up. Default is 0.

Details

This is a function to calculate sample size needed to test non-inferiority of an active treatment against the control within a specific NI margin. The margin can be specified on a number of different scales, though functions for summary measures other than HR are currently under development.

Value

The output is a vector, containing the sample sizes for the control and active arms respectively.

Examples

```
out5A<-samplesize.NI.survival(HR.target=1,p.control.expected=0.2,
                             p.experim.target=0.2, NI.margin=1.5)
```

samplesize.NIfrontier.binary

Sample size calculator for trials with binary outcomes designed with Non-Inferiority Frontiers

Description

A function to estimate sample size of a trial designed with a non-inferiority frontier and analysed on either the risk difference, risk ratio, odds ratio or arc-sine difference scale.

Usage

```
samplesize.NIfrontier.binary(p.control.expected, p.experim.target=NULL, NI.frontier, sig.level=0.025,
                             summary.measure="RD", print.out=TRUE, unfavourable=TRUE,
                             power=0.9, r=1, round=T, ltfu=0)
```

Arguments

p.control.expected	Expected event risk in the control arm.
p.experim.target	Target event risk in the experimental arm under which to power the trial.
NI.frontier	Non-inferiority frontier, a function whos eonly input should be the control event risk and that returns the NI margin for that risk expressed as the specified summary measure.
sig.level	One-sided significance level for testing. Default is 0.025, i.e. 2.5%.

power	Power of the trial, i.e. one minus type-II error of the study. Default is 0.9, i.e. 90%.
r	Allocation ratio, i.e. ratio between sample sizes in the active and control groups. Default is 1.
summary.measure	The population-level summary measure to be estimated, i.e. the scale on which we define the non-inferiority margin. Can be one of "RD" (Risk Difference), "RR" (Risk Ratio), "OR" (Odds Ratio) or "AS" (Arc-Sine difference).
print.out	Logical. If FALSE, no output is printed.
round	A logical variable. If TRUE, sample sizes are rounded to the next integer, using the ceiling function. Otherwise, they are left unrounded.
ltfu	A numeric variable with the expected proportion of patients lost to follow-up. Default is 0.

Details

This function estimates sample size to achieve certain power for a non-inferiority trial designed defining a non-inferiority frontier. The NI frontier can be defined using different summary measures. The method used for sample size calculation is "LRT".

Value

A data.frame with estimated powers at each control event risk.

Examples

```
NI.frontier.RD<-function(p) return(0.1)
out<-try(samplesize.NI.frontier.binary(p.control.expected=0.1, p.experim.target=0.1,
NI.frontier=NI.frontier.RD, sig.level = 0.025,
power = 0.9, r = 1,
summary.measure = "RD", print.out = TRUE,
unfavourable=T))
```

samplesize.ROCI.binary

Function to compute the necessary sample size for a MAMS-ROCI (aka DURATIONS) randomised trial.

Description

A function that can be used to do sample size calculations for a MAMS-ROCI randomised trial with binary outcome. The trial can aim to use one of several possible analysis methods to find the minimum/maximum non-inferior arm.

Usage

```
samplesize.ROCI.binary(p.expected.curve, NI.margin, reference=max(treatment.levels),
                        optimal=min(treatment.levels), range=optimal, unfavourable=T,
                        se.method="bootstrap", treatment.levels, summary.measure="RD",
                        tr.model="FP2.classic", M.boot=NULL, parallel="no", n.cpus=1, sig.level=0.025,
                        n.per.arm=100, power=0.8, print.out=T, round=T, ltfu=0)
```

Arguments

<code>p.expected.curve</code>	The expected event risks at the specified treatment levels. It should be a numeric vector of the same length as <code>treatment.levels</code> .
<code>NI.margin</code>	The non-inferiority margin(s). It can be either a scalar, if the margin is the same for all treatment levels, or a vector of length equal to that of <code>treatment.levels</code> minus one. For <code>summary.measure="target.risk"</code> , this represents the target probability that we want to achieve, and if given as a vector it has length equal to that of <code>treatment.levels</code> .
<code>reference</code>	The arm that should act as a reference. It must be one of the arms in <code>treatment.levels</code> . Default is the largest value among <code>treatment.levels</code> .
<code>optimal</code>	The arm that should be considered as the optimal under the given data generating mechanism. It must be one of the arms in <code>treatment.levels</code> . Default is the smallest value among <code>treatment.levels</code> .
<code>range</code>	The arm(s) that should be considered in the acceptable range for the range power method under the given data generating mechanism. It must be one or more of the arms in <code>treatment.levels</code> . Default is the smallest value among <code>treatment.levels</code> (same as <code>optimal</code>).
<code>unfavourable</code>	If TRUE the outcome is taken to be an unfavourable event, e.g. death or relapse. If FALSE, the outcome is considered a positive one, like cure.
<code>se.method</code>	Method for calculating standard errors around the treatment-response curve. Can be either "bootstrap" or "delta".
<code>treatment.levels</code>	A vector with all the treatment levels to be investigated. These will generally correspond to the treatment arms.
<code>summary.measure</code>	The population-level summary measure that defines the estimand. It can be either "RD" (Risk Difference), "RR" (Risk Ratio), "target.risk" (target probability) or "OR" (Odds Ratio).
<code>tr.model</code>	Model to use for the treatment-response curve. It can be one of: "FP1.classic" or "FP2.classic" (Multivariable Fractional Polynomials as in package <code>mfp</code> , with a maximum of either 1 or 2 powers), "FP1.fixed" or "FP2.fixed" (Fractional polynomials fixing the number of powers to exactly 1 or 2).
<code>M.boot</code>	Number of bootstrap samples if using <code>se.method="bootstrap"</code> .
<code>parallel</code>	If using bootstrap, this gives the chance to parallelise calculations. Use <code>parallel="snow"</code> with windows, or <code>parallel="multicore"</code> with Linux. See help page for the <code>boot</code> function.

n.cpus	If using bootstrap in parallel, this gives the number of cpus to parallelise computations on.
sig.level	One-sided significance level.
n.per.arm	The number of patients per arm used in the estimation of variance on a data set of expected outcomes. Default is 100.
power	The targeted power level.
print.out	If TRUE the estimated sample sizes are printed on screen.
round	A logical variable. If TRUE, sample sizes are rounded to the next integer, using the ceiling function. Otherwise, they are left unrounded.
ltfu	A numeric variable with the expected proportion of patients lost to follow-up. Default is 0.

Details

This function estimates the sample size to target both optimal and acceptable power for a ROCI trial. It requires as input the expected treatment response curve under which to power the trial, the NI margins and a specification of all the methods. The function creates a data set of expected outcomes (e.g. if risk in one arm is 0.7, it creates 70 records with event=1 and 30 with event=0) and fits the model for the treatment-response curve on this data set. It then estimates the variance of the summary measure of interest and uses this to do a sample size calculation based on simple formula from normal theory for non-inferiority with a continuous outcome. The function allows for the margin to be expressed as either a risk difference, risk ratio, odds ratio or a target probability.

Value

A list including both total and arm-specific sample sizes to reach the required level of both optimal and acceptable power. Acceptable power is estimated conservatively as the minimum sample size to achieve certain power for any of the treatment levels. The function also returns sample size to achieve same power level for all specific treatment levels.

Examples

```
p.expected.curve=rep(0.05,7)
NI.margin=rep(0.05,6)
reference=20
optimal=8
range=c(8,10,12)
se.method="delta"
treatment.levels<-c(8,10,12,14,16,18,20)
summary.measure<-"RD"
tr.model="FP2.classic"
sig.level=0.05
power=0.8
unfavourable=T
```

```
ss<-sampleSize.ROCI.binary(p.expected.curve, NI.margin, reference=reference, optimal=optimal, range=range, unfav
se.method=se.method,treatment.levels=treatment.levels,summary.measure=summary.measure,
tr.model=tr.model, sig.level = sig.level, power=power)
```

`summary.ROCI`*Summarise the results of the analysis of a MAMS-ROCI trial*

Description

A function for summarising results of the analysis of a MAMS-ROCI randomised trial that has been analysed with the `test.ROCI.binary` function.

Usage

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

Arguments

<code>object</code>	A list obtained as an output from function <code>test.ROCI.binary</code> .
<code>...</code>	Not used.

Details

This is a function to summarise results of a call to `test.ROCI.binary`.

Examples

```
duration.arms=c(8,10,12,14,16,18,20)  
sam.sizes=c(700)  
NI.margin.RD<-0.1  
  
durations<-rep(duration.arms, each=100)  
y<-rbinom(sam.sizes,1,0.05+(20-durations)*0.01)  
  
data.ex<-data.frame(y,durations)  
myformula<-as.formula(y~treat(durations))  
  
res1<-test.ROCI.binary(formula=myformula, data=data.ex,  
                        se.method="delta", treatment.levels=8:20, summary.measure="RD",  
                        NI.margin=NI.margin.RD)  
  
summary(res1)
```

test.NI.binary	<i>Non-inferiority testing with binary outcome.</i>
----------------	---

Description

Functions for testing non-inferiority when the outcome is binary. It allows for a number of tests on different summary measures: risk difference, risk ratio, odds ratio or arc-sine difference.

Usage

```
test.NI.binary(n.control=NULL, n.experim=NULL, e.control=NULL,
              e.experim=NULL, formula=NULL, data=NULL,
              control.level=0, NI.margin, sig.level=0.025,
              summary.measure="RD", print.out=TRUE,
              unfavourable=TRUE, test.type=NULL, M.boot=2000,
              bootCI.type="bca", BB.adj=0.0001,
              recursive.p.estim=FALSE)
```

Arguments

e.control	Number of events in the control arm.
e.experim	Number of events in the active arm.
n.control	Total sample size of the control arm.
n.experim	Total sample size of the active arm.
formula	The formula for the outcome model. The variable indicating treatment has to be put within brackets and preceded by treat, e.g. treat(treatment).
data	A data.frame with all data.
NI.margin	Non-inferiority margin, expressed as the specified summary measure.
control.level	Defines the control level in the treatment variable when using the formula and data interface. Defaults to 0.
sig.level	One-sided significance level for testing. Default is 0.025, i.e. 2.5%.
summary.measure	The population-level summary measure to be estimated, i.e. the scale on which we define the non-inferiority margin. Can be one of "RD" (Risk Difference), "RR" (Risk Ratio), "OR" (Odds Ratio) or "AS" (Arc-Sine difference).
print.out	Logical. If FALSE, no output is printed.
unfavourable	Logical. If FALSE, the outcome is considered favourable, e.g. cure. Default is TRUE, i.e. unfavourable outcome, e.g. death.
test.type	A character string defining the method to be used for calculation of the confidence interval. For the risk difference, methods available include "Wald", "Wald.cc" (with continuity correction), "Hauck.Anderson", "Gart.Nam", "Newcombe10", "Newcombe11", "Haldane", "Jeffrey.Perks", "Agresti.Caffo", "Miettinen.Nurminen", "Farrington.Manning", "logistic" (marginalisation after fitting

logistic regression model with glm), "bootstrap", "Agresti.Min", "Brown.Li.Jeffreys", "Chan.Zhang", "BLNM", "Mee", "uncond.midp", "Berger.Boos", "MUE.Lin", "MUE.parametric.bootstrap". For risk ratio, possible methods are "Wald.Katz", "adjusted.Wald.Katz", "inverse.hyperbolic.sine", "Koopman", "MOVER.R", "Miettinen.Nurminen", "MOVER", "Gart.Nam", "score.cc" (with continuity correction), "logregression" (binomial regression with log link), "bootstrap", "Bailey", "Noether", "Chan.Zhang", "Agresti.Min", "uncond.midp", "Berger.Boos". For odds ratio, methods available include "Wald.Woolf", "adjusted.Wald.Woolf", "inverse.hyperbolic.sine", "Cornfield.exact", "MOVER.R", "Miettinen.Nurminen", "MOVER", "Gart.Nam", "score.cc", "logistic" (logistic regression), "bootstrap", "Cornfield.midp", "Baptista.Pike.exact", "Baptista.Pike.midp", "Chan.Zhang", "Agresti.Min", "uncond.midp", "Berger.Boos". Finally, for arc-sine difference the only available method is "Wald".

M.boot	Number of bootstrap samples, e.g. for "bootstrap" and "MUE.parametric.bootstrap" methods.
bootCI.type	Method for computing the confidence intervals if using a bootstrap method. It can be either "norm", "basic", "perc" or "bca". Default is "bca", which is the recommended option when possible.
BB.adj	Adjustment factor for "Berger.Boos" method.
recursive.p.estim	If TRUE, the p value is estimated by recursively running the function with varied significance level until the NI margin is crossed. If FALSE (default), the p-value is either the one calculated with standard methods or an estimate based on normal approximation.

Details

This is a function to test non-inferiority of an experimental treatment against the active control within a specific NI margin. The margin can be specified on a number of different summary measures, including absolute risk difference, risk ratio, odds ratio and arc-sine difference. It is possible to test both with favourable (e.g. cure) or unfavourable (e.g. death) outcomes and using a multitude of methods taken from other packages. See the entry on the test.type argument for the specific methods available for each summary measure.

Value

The output is a list, containing the estimate, standard error, confidence interval (two-sided $2*\alpha$ level), Z statistic and p-value. Additionally, a non-inferiority indicator is included and an indicator of whether the p-value was precise or just estimated from the confidence interval using normal approximation.

Examples

```
n.control<-1000
n.experim<-1000
e.control<-0.05*n.control
e.experim<-0.05*n.experim
NIIRD=0.05
NIIRR=2
```

```

NI.mOR=1.7
NI.mAS=arcsine.margin(0.1, 0.05)
alpha=0.025

```

```

test1<-test.NI.binary(n.control=n.control, n.experim=n.experim, e.control=e.control, e.experim=e.experim, NI.mar
test1b<-test.NI.binary(n.control=n.control, n.experim=n.experim, e.control=e.control, e.experim=e.experim, NI.ma
test2<-test.NI.binary(n.control=n.control, n.experim=n.experim, e.control=e.control, e.experim=e.experim, NI.mar
test3<-test.NI.binary(n.control=n.control, n.experim=n.experim, e.control=e.control, e.experim=e.experim, NI.mar
test4<-test.NI.binary(n.control=n.control, n.experim=n.experim, e.control=e.control, e.experim=e.experim, NI.mar

```

test.NI.continuous *Non-inferiority testing with continuous outcome.*

Description

Functions for testing non-inferiority when the outcome is continuous. It allows for a number of tests on different summary measures: difference of means or ratio of means.

Usage

```

test.NI.continuous(y.control=NULL, y.experim=NULL, NI.margin, sig.level=0.025,
                  summary.measure="mean.difference",
                  formula=NULL, data=NULL, control.level=0,
                  print.out=TRUE, higher.better=TRUE, test.type=NULL,
                  M.boot=2000, bootCI.type="bca", sd.control=NULL,
                  sd.experim=NULL)

```

Arguments

y.control	Vector of measurements in the control arm.
y.experim	Vector of measurements in the experimental arm.
NI.margin	Non-inferiority margin, expressed as the specified summary measure.
sig.level	One-sided significance level for testing. Default is 0.025, i.e. 2.5%.
summary.measure	The population-level summary measure to be estimated, i.e. the scale on which we define the non-inferiority margin. Can be one of "mean.difference" (Difference of Means) or "mean.ratio" (Ratio of Means).
formula	The formula for the outcome model. The variable indicating treatment has to be put within brackets and preceded by treat, e.g. treat(treatment).
data	A data.frame with all data.
control.level	Defines the control level in the treatment variable when using the formula and data interface. Defaults to 0.
print.out	Logical. If FALSE, no output is printed.

higher.better	Logical. If FALSE, the outcome is considered unfavourable, i.e. higher scores indicate worse outcomes. Default is TRUE, i.e. favourable outcome, higher scores indicate better outcomes.
test.type	A character string defining the method to be used for calculation of the confidence interval. For the mean difference, methods available include "Z.test", "t.test" or bootstrap based on 3 different types of confidence intervals: "bootstrap.basic", "bootstrap.bca" or "bootstrap.percentile". For the mean ratio, methods available include "Fiellers" test, "lm" (marginalisation after using linear regression) or the three methods using bootstrap: "bootstrap.basic", "bootstrap.bca" or "bootstrap.percentile".
M.boot	Number of bootstrap samples, e.g. for "bootstrap" and "MUE.parametric.bootstrap" methods.
bootCI.type	Method for computing the confidence intervals if using a bootstrap method. It can be either "norm", "basic", "perc" or "bca". Default is "bca", which is the recommended option when possible.
sd.control	The assumed standard deviation of the control arm if using a Z test.
sd.experim	The assumed standard deviation of the experimental arm if using a Z test.

Details

This is a function to test non-inferiority of an experimental treatment against the active control within a specific NI margin. The margin can be specified on a number of different summary measures, including mean difference, or mean ratio. It is possible to test both with favourable (e.g. cognitive score) or unfavourable (e.g. pain score) outcomes and using a multitude of methods taken from other packages. See the entry on the test.type argument for the specific methods available for each summary measure.

Value

The output is a list, containing the estimate, standard error, confidence interval (two-sided $2 \times \alpha$ level), Z statistic and p-value. Additionally, a non-inferiority indicator is included and an indicator of whether the p-value was precise or just estimated from the confidence interval using normal approximation.

Examples

```

y0<-rnorm(10,2)
y1<-rnorm(10,2)
NI.m=-0.75
alpha=0.025

set.seed(1)
out5A<-test.NI.continuous(y0, y1, NI.m, alpha, test.type="Z.test", sd.control=1, sd.experim = 1)

```

test.NI.survival	<i>Non-inferiority testing with survival outcome.</i>
------------------	---

Description

Functions for testing non-inferiority when the outcome is survival (time-to-event). It allows for a number of tests on different summary measures: Hazard Ratio, Difference in Restricted Mean Survival Time and Difference in Survival.

Usage

```
test.NI.survival(time, event, treat, covariates=NULL, NI.margin, sig.level=0.025, summary.measure="H",
  print.out=TRUE, unfavourable=TRUE, test.type=NULL,
  M.boot=2000, bootCI.type="bca", tau=NULL, control.level=NULL,
  k=2, knots=NULL, bknots=NULL)
```

Arguments

time	A vector with the event (or censoring) times.
event	A vector with the event indicators.
treat	A vector with treatment indicators.
covariates	Optional: a data.frame including all baseline covariates to be adjusted for.
NI.margin	Non-inferiority margin, expressed as the specified summary measure.
sig.level	One-sided significance level for testing. Default is 0.025, i.e. 2.5%.
summary.measure	The population-level summary measure to be estimated, i.e. the scale on which we define the non-inferiority margin. Can be one of "HR" (Hazard Ratio), "DRMST" (Difference in Restricted Mean Survival Time by time tau), or "DS" (Difference in Surviving proportion at time tau).
print.out	Logical. If FALSE, no output is printed.
unfavourable	Logical. If FALSE, the outcome is considered favourable, e.g. cure. Default is TRUE, i.e. unfavourable outcome, e.g. death.
test.type	A character string defining the method to be used for calculation of the confidence interval. For the hazard ratio, methods available include "Cox.PH" (simple Cox proportional hazards model), "flexsurv.PH" (flexible parametric regression model) or "Cox.weighted" (Weighted Cox regression model, using Prentice weights). For "DRMST", possible methods are "KM" (Kaplan Meier), "Cox.PH.bootstrap" (bootstrap on estimates from Cox), "flexsurv.PH.delta" (delta method on estimates from flexible parametric survival model), "flexsurv.PH.bootstrap" (bootstrap on estimates from flexible parametric survival model). For "DS", methods available include "KM" (Kaplan Meier), "Cox.PH.bootstrap", "flex-surv.PH.delta", "flexsurv.PH.bootstrap".
M.boot	Number of bootstrap samples.

<code>bootCI.type</code>	Method for computing the confidence intervals if using a bootstrap method. It can be either "norm", "basic", "perc" or "bca". Default is "bca", which is the recommended option when possible.
<code>k</code>	If using flexsurv methods, this is the number of knots in the spline. The default <code>k=0</code> gives a Weibull model. <code>k</code> is equivalent to <code>df-1</code> in the notation of <code>stpm</code> for Stata. The knots are then chosen as equally-spaced quantiles of the log uncensored survival times, for example, at the median with one knot, or at the 33% and 67% quantiles of log time (or time, see "timescale") with two knots. To override this default knot placement, specify <code>knots</code> instead.
<code>knots</code>	If using flexsurv methods, Locations of knots on the axis of log time. If not specified, knot locations are chosen as described in <code>k</code> above. Either <code>k</code> or <code>knots</code> must be specified. If both are specified, <code>knots</code> overrides <code>k</code> .
<code>bknots</code>	If using flexsurv methods, this is the location of boundary knots, on the axis of log time. If not supplied, these are chosen as the minimum and maximum log event time.
<code>tau</code>	The horizon time <code>tau</code> if using either "DRMST" or "DS" as summary measures.
<code>control.level</code>	A character variable specifying the level of the factor variable for treatment that has to be considered as the control one. If missing, the automatic reference will be assumed to be the control and a NOTE is printed on screen specifying which one that is.

Details

This is a function to test non-inferiority of an experimental treatment against the active control within a specific NI margin. The margin can be specified on a number of different summary measures, including HR, DRMST and difference in survival. It is possible to test both with favourable (e.g. cure) or unfavourable (e.g. death) outcomes and using different analysis methods. See the entry on the `test.type` argument for the specific methods available for each summary measure.

Value

The output is a list, containing the estimate, standard error, confidence interval (two-sided $2*\alpha$ level) and p-value. Additionally, a non-inferiority indicator is included and an indicator of whether the p-value was precise or just estimated from the confidence interval using normal approximation.

Examples

```
time1<-rnorm(200,100,10)
event1<-rbinom(200,1,0.5)
treat1<-rep(0:1,each=100)
out5A<-test.NI.survival(time=time1, event=event1, treat=treat1, NI.margin=1.5)
```

test.NIfrontier.binary

Non-inferiority testing with binary outcome using a non-inferiority frontier.

Description

Functions for testing non-inferiority when the outcome is binary and the goal is to use a non-inferiority frontier. It allows for a number of tests on different summary measures: risk difference, risk ratio, odds ratio or arc-sine difference. It also allows for direct Likelihood Ratio Test of the frontier.

Usage

```
test.NIfrontier.binary(n.control, n.experim, e.control, e.experim,
                      NI.frontier, sig.level, summary.measure="RD",
                      print.out=TRUE, unfavourable=TRUE, test.type=NULL,
                      M.boot=2000, bootCI.type="bca", BB.adj=0.0001)
```

Arguments

e.control	Number of events in the control arm.
e.experim	Number of events in the active arm.
n.control	Total sample size of the control arm.
n.experim	Total sample size of the active arm.
NI.frontier	Non-inferiority frontier, a function whos eonly input should be the control event risk and that returns the NI margin for that risk expressed as the specified summary measure.
sig.level	One-sided significance level for testing. It can be both a scalar (same significance level for any control risk) or a function in oen variable (giving significance level to be used for each observed risk). Default is 0.025, i.e. 2.5%.
summary.measure	The population-level summary measure to be estimated, i.e. the scale on which we define the non-inferiority margins through the frontier. Can be one of "RD" (Risk Difference), "RR" (Risk Ratio), "OR" (Odds Ratio) or "AS" (Arc-Sine difference).
print.out	Logical. If FALSE, no output is printed.
unfavourable	Logical. If FALSE, the outcome is considered favourable, e.g. cure. Default is TRUE, i.e. unfavourable outcome, e.g. death.
test.type	A character string defining the method to be used for calculation of the confidence interval. For the risk difference, methods available include "Wald", "Wald.cc" (with continuity correction), "Hauck.Anderson", "Gart.Nam", "Newcombe10", "Newcombe11", "Haldane", "Jeffrey.Perks", "Agresti.Caffo", "Miettinen.Nurminen", "Farrington.Manning", "logistic" (marginalisation after fitting

logistic regression model with glm), "bootstrap", "Agresti.Min", "Brown.Li.Jeffreys", "Chan.Zhang", "BLNM", "Mee", "uncond.midp", "Berger.Boos", "MUE.Lin", "MUE.parametric.bootstrap" and "LRT". For risk ratio, possible methods are "Wald.Katz", "adjusted.Wald.Katz", "inverse.hyperbolic.sine", "Koopman", "MOVER.R", "Miettinen.Nurminen", "MOVER", "Gart.Nam", "score.cc" (with continuity correction), "logregression" (binomial regression with log link), "bootstrap", "Bailey", "Noether", "Chan.Zhang", "Agresti.Min", "uncond.midp", "Berger.Boos", "LRT". For odds ratio, methods available include "Wald.WoOLF", "adjusted.Wald.WoOLF", "inverse.hyperbolic.sine", "Cornfield.exact", "MOVER.R", "Miettinen.Nurminen", "MOVER", "Gart.Nam", "score.cc", "logistic" (logistic regression), "bootstrap", "Cornfield.midp", "Baptista.Pike.exact", "Baptista.Pike.midp", "Chan.Zhang", "Agresti.Min", "uncond.midp", "Berger.Boos", "LRT". Finally, for arc-sine difference the only available methods are "LRT" and "Wald".

M.boot	Number of bootstrap samples, e.g. for "bootstrap" and "MUE.parametric.bootstrap" methods.
bootCI.type	Method for computing the confidence intervals if using a bootstrap method. It can be either "norm", "basic", "perc" or "bca". Default is "bca", which is the recommended option when possible.
BB.adj	Adjustment factor for "Berger.Boos" method.

Details

This is a function to test non-inferiority of an experimental treatment against the active control within a specific NI margin defined using a NI frontier. The margin can be specified on a number of different summary measures, including absolute risk difference, risk ratio, odds ratio and arc-sine difference. It is possible to test both with favourable (e.g. cure) or unfavourable (e.g. death) outcomes and using a multitude of methods taken from other packages. See the entry on the test.type argument for the specific methods available for each summary measure. The function also allows for direct test of the frontier using LRT, reporting results on a chosen summary measure.

Value

The output is a list, containing the estimate, standard error, confidence interval (two-sided $2 \cdot \alpha$ level), Z statistic and p-value. Additionally, a non-inferiority indicator is included and an indicator of whether the p-value was precise or just estimated from the confidence interval using normal approximation.

Examples

```
n.control<-1000
n.experim<-1000
e.control<-0.05*n.control
e.experim<-0.05*n.experim
NImRD=function(p0) {
  return(0.05)
}
alpha=0.025
```

```
test1<-test.NIfrontier.binary(n.control=n.control, n.experim=n.experim, e.control=e.control, e.experim=e.experim)
```

test.ROCI.binary	<i>Function to analyse a MAMS-ROCI (aka DURATIONS) randomised trial.</i>
------------------	--

Description

A function that can be used to analyse a MAMS-ROCI randomised trial with binary outcome using one of several possible analysis methods to find the minimum/maximum non-inferior arm.

Usage

```
test.ROCI.binary(formula=NULL, data=NULL, NI.margin, reference=max(treatment.levels),
                 se.method="bootstrap", treatment.levels=unique(treatment), summary.measure="RD",
                 tr.model="FP2.fixed", M.boot=NULL, bootCI.type="bca", parallel="no", n.cpus=1, s
                 unfavourable=TRUE)
```

Arguments

formula	The formula for the outcome model. The variable indicating the treatment aspect to optimise has to be put within brackets and preceded by treat, e.g. treat(treatment).
data	A data.frame with all data.
NI.margin	The non-inferiority margin(s). It can be either a scalar, if the margin is the same for all treatment levels, or a vector of length equal to that of treatment.levels minus one. For summary.measure="target.risk", this represents the target probability that we want to achieve, and if given as a vector it has length equal to that of treatment.levels.
reference	The arm that should act as a reference. It must be one of the arms in treatment levels. Default is the largest value among treatment.levels.
se.method	Method for calculating standard errors around the treatment-response curve. Can be either "bootstrap" or "delta".
treatment.levels	A vector with all the treatment levels to be investigated. Default is the ones to which patients were randomised.
summary.measure	The population-level summary measure that defines the estimand. It can be either "RD" (Risk Difference), "RR" (Risk Ratio), "target.risk" (target probability) or "OR" (Odds Ratio).
tr.model	Model to use for the treatment-response curve. It can be one of: "FP1.classic" or "FP2.classic" (Multivariable Fractional Polynomials as in package mfp, with a maximum of either 1 or 2 powers), "FP1.fixed" or "FP2.fixed" (Fractional polynomials fixing the number of powers to exactly 1 or 2).

M.boot	Number of bootstrap samples if using se.method="bootstrap".
bootCI.type	Method for computing the confidence intervals if using se.method="bootstrap". It can be either "norm", "basic", "perc" or "bca". Default is "bca", which is the recommended option when possible.
parallel	If using bootstrap, this gives the chance to parallelise calculations. Use parallel="snow" with windows, or parallel="multicore" with Linux. See help page for the boot function.
n.cpus	If using bootstrap in parallel, this gives the number of cpus to parallelise computations on.
sig.level	One-sided significance level.
unfavourable	If TRUE the outcome is taken to be an unfavourable event, e.g. death or relapse. If FALSE, the outcome is considered a positive one, like cure.

Details

This function takes trial data on randomised treatment arm (e.g. duration) and associated binary outcome (e.g. 1-year death, or cure) and returns the estimated treatment-response curve. It estimates the chosen population-level summary measure (RD, RR or RR) either against the largest arm (when minimisation=T) or the shortest one (when minimisation=F). It returns what is the optimum level of treatment according to trial results as well. Results can then be explored and visualised with summary.durations and plot.durations.

Value

A list including the model fit, the optimal treatment level, mean estimates and upper and lower bounds of the CI for the summary measure on the primary outcome at all treatment levels against control. Additionally, the list includes inputs such as summary measure, analysis method, non-inferiority margin and treatment levels.

Examples

```
duration.arms=c(8,10,12,14,16,18,20)
sam.sizes=c(700)
NI.margin.RD<-0.1

durations<-rep(duration.arms, each=100)
y<-rbinom(sam.sizes,1,0.05+(20-durations)*0.01)

data.ex<-data.frame(y,durations)
myformula<-as.formula(y~treat(durations))

res1<-test.ROCI.binary(formula=myformula, data=data.ex,
                       se.method="delta", treatment.levels=8:20,
                       summary.measure="RD", NI.margin=NI.margin.RD)
```

```
type1error.NIfrontier.binary
```

Function to compute type 1 error rate for a 2-arm non-inferiority trial designed using non-inferiority frontiers.

Description

A function that can be used to compute type 1 error rate for a non-inferiority trial with binary outcome designed using non-inferiority frontiers. The trial can aim to use one of several possible analysis methods and summary measures.

Usage

```
type1error.NIfrontier.binary(p.control.expected, NI.frontier, sig.level=0.025,
                             summary.measure="RD", print.out=TRUE, unfavourable=TRUE,
                             n.control, n.experim, n.rep=1000, M.boot=2000, BB.adj=0.0001,
                             test.type="LRT")
```

Arguments

p.control.expected	Expected event risk in the control arm.
NI.frontier	Non-inferiority frontier, a function whos eonly input should be the control event risk and that returns the NI margin for that risk expressed as the specified summary measure.
sig.level	One-sided significance level for testing. Default is 0.025, i.e. 2.5%.
n.control	The sample size in the control arm for which to estimate type1error.
n.experim	The sample size in the experimental arm for which to estimate type1error.
summary.measure	The population-level summary measure to be estimated, i.e. the scale on which we define the non-inferiority margin. Can be one of "RD" (Risk Difference), "RR" (Risk Ratio), "OR" (Odds Ratio) or "AS" (Arc-Sine difference).
print.out	Logical. If FALSE, no output is printed.
test.type	A string that indicates the type of test to be assumed for the sample size calculation. Currently, three options are supported: "Wald", "score" and "local".
unfavourable	A logical variable. If TRUE, the outcome is considered unfavourable. This is used to check that the NI margin specified is meaningful.
n.rep	The number of repetitions of the simulations to estimate type1error.
M.boot	Number of bootstrap samples if using a bootstrap-based analysis method.
BB.adj	Adjustment factor for "Berger.Boos" method.

Details

This function estimates type 1 error rate of a two-arm non-inferiority trial designed using non-inferiority frontiers for a given sample size, running a certain number `n.rep` of simulations under the null hypothesis and calculating estimated success rate of the trial with the desired design and analysis methods.

Value

The estimated type1error. On screen, the Monte-Carlo Confidence Interval is printed as well.

Examples

```
NI.f<-function(p) {  
  marg<-ifelse(p<0.15,p, 0.15)  
  return(marg)  
}  
t1err<-type1error.NIfrontier.binary(p.control.expected=0.2, NI.frontier=NI.f, n.control=200, n.experim=200,  
  n.rep=500)
```

Index

compare.NIfrontier.binary, [2](#)
compare.NIfrontier.continuous, [3](#)
compare.NIfrontier.survival, [5](#)
convertmargin.binary, [6](#)
convertmargin.continuous, [7](#)
convertmargin.survival, [8](#)

plot.ROCI, [10](#)
power.NI.binary, [11](#)
power.NI.continuous, [12](#)
power.NIfrontier.binary, [14](#)
power.ROCI.binary, [15](#)

samplesize.NI.binary, [17](#)
samplesize.NI.continuous, [19](#)
samplesize.NI.survival, [21](#)
samplesize.NIfrontier.binary, [22](#)
samplesize.ROCI.binary, [23](#)
summary.ROCI, [26](#)

test.NI.binary, [27](#)
test.NI.continuous, [29](#)
test.NI.survival, [31](#)
test.NIfrontier.binary, [33](#)
test.ROCI.binary, [35](#)
type1error.NIfrontier.binary, [37](#)