Package 'NNTbiomarker'

October 12, 2022

Type Package

Title Calculate Design Parameters for Biomarker Validation Studies

Version 0.29.11

Date 2015-08-20

Author Roger Day

Maintainer Roger Day <day01@pitt.edu>

Description Helps a clinical trial team discuss
the clinical goals of a well-defined biomarker with a diagnostic,
staging, prognostic, or predictive purpose. From this discussion will
come a statistical plan for a (non-randomized) validation trial.
Both prospective and retrospective trials are supported. In a specific
focused discussion, investigators should determine the range of
``discomfort'' for the NNT, number needed to treat. The meaning of
the discomfort range, [NNTlower, NNTupper], is that within this range
most physicians would feel discomfort either in treating or withholding
treatment. A pair of NNT values bracketing that range, NNTpos and NNTneg,
become the targets of the study's design. If the trial can demonstrate
that a positive biomarker test yields an NNT less than NNTlower,
and that a negative biomarker test yields an NNT less than NNTlower,
then the biomarker may be useful for patients. A highlight of the package
is visualization of a ``contra-Bayes'' theorem, which produces criteria for
retrospective case-controls studies.

License GPL-3

Imports shiny, xtable, stringr, magrittr, mvbutils

Collate aaa.R ifVerboseCat.R sesp-pv-NNT.R achievable.se.sp.R
binom.confint.R NNTintervals.R run.R ROCplots.R argmin.R zzz.R

Suggests testthat (>= 0.8.1), knitr (>= 1.6), rmarkdown, ggplot2, plyr

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2015-08-21 07:38:39

R **topics documented:**

NNTbiomarker-package   *Plan a biomarker validation study by focusing on desired clinical actionability.*

Description

Clarifying what performance would suffice if the test is to improve medical care makes it possible to design meaningful validation studies.

Details

Package:	NNTbiomarker
Type:	Package
Version:	0.1
Date:	2015-03-21
License:	What license is it under?

This package bases the design of a biomarker study on the idea of "number needed to treat" (NNT). It postulates a "range of discomfort" for NNT, within which the clinical decision is uncomfortable for a treating physician. provides a shiny window for eliciting the boundaries of the range of number needed to

Author(s)

Roger Day

Maintainer: Roger Day <day01@pitt.edu>

References

See author for unpublished manuscript.

Examples

```
NNT.to.pv(NNTpos=5, NNTneg=28)
NNT.from.sesp(se=0.7, sp=0.9, prev=0.1)
pv.to.sesp(pv = cbind(ppv=seq(.5, .9, .1), npv=0.9), prev = 0.2)
```

---

achievable.se.sp	*achievable.se.sp(): target sensitivity and specificity for a retrospective study.*

Description

For a retrospective study design, given a prevalence value, produce a plot displaying the achievable contours of either predictive values or NNT values. The calculation uses the "contra-Bayes" theorem, sesp.from.pv.

Usage

```
achievable.se.sp(the.prev = 0.5, axes = c("pv", "NNT"),
  sesp.seq = seq(0.5, 1, 0.1), drawNNTaxes = TRUE, drawPVaxes = FALSE,
  drawArrows = TRUE, drawTable = TRUE, latexTable = TRUE,
  placePointLabels = TRUE, cexText = 0.5, cexSubtitle = 0.5,
  cexTitle = 0.7, y0arrow = 0.25, lwdArrow = 1, ltyArrow = 2,
  title = FALSE, mtext = FALSE, contours = TRUE, ...)
```

Arguments

the.prev	Prevalence (prior probability)
axes	Should the axes be predictive values ("pv") or NNT values? Default is "pv".
sesp.seq	Sequence of values at which the sensitivity and specificity will be explored.
drawNNTaxes	(default=TRUE) Option for tweaking the plot.
drawPVaxes	(default=FALSE) Option for tweaking the plot.
drawArrows	(default=TRUE) Arrow option; deprecated.
drawTable	(default=TRUE) Option for tweaking the plot.
latexTable	(default=TRUE) Option for tweaking the plot.

placePointLabels	
	(default=TRUE) Write A, B, C, ... in circles where sensitivity=specificity.
cexText	(default=0.5) Option for tweaking the plot.
cexSubtitle	(default=0.5) Option for tweaking the plot.
cexTitle	(default=0.7) Option for tweaking the plot.
y0arrow	(default=0.25) Arrow option; deprecated.
lwdArrow	(default=1) Arrow option; deprecated.
ltyArrow	(default=2) Arrow option; deprecated.
title	(default=FALSE) Option for tweaking the plot.
mtext	(default=FALSE) Option for tweaking the plot.
contours	(default=TRUE) Option for tweaking the plot.
...	Options to pass to plot.default()

Value

The predictive values when sensitivity equals specificity: sesp.to.pv(cbind(sesp.seq,sesp.seq), prev=the.prev))

---

argmin	*argmin Argmin function for a vector.*

Description

Return the index minimizing distance from v to target.

Usage

```
argmin(v, target = 0)
```

Arguments

v	The vector to compare to target.
target	The value sought in the vector; default=0.

Value

The index in v of the value which is closest to target.

---

binom.confint	*binom.confint*

Description

Exact confidence intervals for a binomial proportion parameter.

Usage

```
binom.confint(k, n, alpha = 0.05, side = c("two", "upper", "lower"))
```

Arguments

k	#"heads"
n	sample size
alpha	Confidence level
side	Sidedness of the hypothesis: c("two", "upper", "lower")

NNT.from.pv	*NNT.from.pv*

Description

Compute NNT values from predictive values.

Usage

```
NNT.from.pv(ppv, npv, pv)
```

Arguments

ppv	Positive predictive value
npv	Negative predictive value
pv	Alternative input of ppv and npv, in matrix or named vector.

Value

c(NNTpos=NNTpos, NNTneg=NNTneg)

`NNT.from.sesp`   *NNT.from.sesp*

Description

Compute NNT values from sensitivity aand specificity.

Usage

```
NNT.from.sesp(se, sp, sesp, prev)
```

Arguments

se	Positive predictive value
sp	Negative predictive value
sesp	Alternative input for se and sp, as matrix or named vector.
prev	Prevalence (prior probability)

Value

c(NNTpos=NNTpos, NNTneg=NNTneg)

`NNT.to.pv`   *NNT.to.pv*

Description

Convert between (NNTpos, NNTneg) and (PPV, NPV).

Usage

```
NNT.to.pv(NNTpos, NNTneg, NNT, prev, calculate.se.sp = F)
```

Arguments

NNTpos	NNT for a patient with positive test result
NNTneg	NNT for a patient with negative test result
NNT	A matrix or vector of (NNTpos, NNTneg) values.
prev	Prevalence of the "BestToTreat" group before testing.
calculate.se.sp	
	(default=FALSE) If TRUE, also calculate the sensitivity and specificity using the contra-Bayes theorem.

Value

For matrix input, cbind(ppv=ppv, npv=npv). For vector input, c(ppv=ppv, npv=npv).

`NNT.to.sesp`   *NNT.to.sesp*

Description

Compute sensitivity aand specificity from NNT values.

Usage

```
NNT.to.sesp(NNTpos, NNTneg, NNT, prev)
```

Arguments

NNTpos	NNT for a positive test result
NNTneg	NNT for a negative test result
NNT	Alternative way in input NNT values (matrix or vector)
prev	Prevalence (prior probability)

Value

c(se=se, sp=sp)

NNTintervalsProspective
                 NNTintervalsProspective

Description

Produce Bayesian and classical intervals for NNT from observations in a prospective study. Useful for "anticipated results" when designing a study, The setting: patients will be tested immediately, and followed to determine the BestToTreat/BestToWait classification. as well as analyzing study results. There were (or will be) Npositives patients with a positive test, Nnegatives with a negative test. The observed NNTs in each group were (or will be) NNTpos and NNTneg.

Usage

```
NNTintervalsProspective(Npositives, Nnegatives, NtruePositives, NtrueNegatives,
  prev = 0.15, alpha = 0.025, prior = c(1/2, 1/2))
```

Arguments

`Npositives`	Total number of observed positives.
`Nnegatives`	Total number of observed negatives.
`NtruePositives`	Observed or anticipated number of "BestToTreat" among the positives.
`NtrueNegatives`	Observed or anticipated number of "BestToWait" among the negatives.
`prev`	= 0.15 Prevalence of "BestToTreat" characteristic.
`alpha`	= 0.025 Significance level (one side).
`prior`	Beta parameters for prior. Default is the Jeffreys prior = c(1/2,1/2). Jaynes prior = c(0,0) won't work when #fp=1.

Value

The Bayesian predictive intervals for NNTpos and NNTneg. These are obtained from predictive intervals for PPV and NPV, based on Jeffreys' beta(1/2,1/2) prior.

NNTintervalsRetrospective

NNTintervalsRetrospective

Description

Bayes predictive intervals for sensitivity, specificity, NNTpos and NNTneg in a case-control retrospective study.

Usage

```
NNTintervalsRetrospective(Ncases = 10, Ncontrols = 30, NposCases = 6,
  NposControls = 2, prev = 0.15, alpha = 0.025, prior = c(1/2, 1/2))
```

Arguments

`Ncases`	Number of cases in the study
`Ncontrols`	Number of controls in the study
`NposCases`	Number of cases with positive test
`NposControls`	Number of controls with positive test
`prev`	Prevalence of the BestToTreat (versus BestToWait)
`alpha`	Significance level for interval.
`prior`	Beta parameters for prior. Default is the Jeffreys prior = c(1/2,1/2). Jaynes prior = c(0,0) won't work when #fp=1.

Value

A list with 3 components containing intervals (predictive or otherwise), with names intervalsForSN, intervalsForSP, intervalsForNNT. The intervals derive from assuming independent Jeffreys priors for SN and SP, sampling from joint independent posteriors for SN and SP incorporating the anticipated results, and applying NNT.from.sesp (Bayes theorem) to each sampled pair to obtain a sample of NNTpos and NNTneg.

---

pv.from.sesp   *pv.from.sesp*

Description

Computes predictive values from sensitivity and specificity.

Usage

```
pv.from.sesp(se = 0.8, sp = 0.8, sesp, prev = 0.001)
```

Arguments

se	Positive predictive value
sp	Negative predictive value
sesp	Alternative input for se and sp, as matrix or named vector.
prev	Prevalence (prior probability). Default = 0.001

Value

c(ppv=ppv, npv=npv)

ROCplots   *ROCplots*

Description

A variety of ROC-related plots for a binary target and a single continuous predictor.

Usage

```
ROCplots(data, whichPlots = c("density", "raw", "ROC", "pv", "nnt",
  "nntRange"), NNTlower = 3, NNTupper = 10, N = 1000, prev = 0.2,
  diffInSD = 2, ...)
```

Arguments

`data`	Data frame with columns "class" (binary target variable) and "X" (predictor).
`whichPlots`	Which plots to do. Options are c("density", "raw", "ROC", "pv", "nnt")
`NNTlower`	Subjective input. If NNT < NNTlower, the decision is clearly to Treat.
`NNTupper`	Subjective input. If NNT > NNTupper, the decision is clearly to Wait.
`N`	For simulated data: sample size
`prev`	For simulated data: Prevalence
`diffInSD`	For simulated data: Difference: E(X | group=1) - E(X | group=0),measured in units of S.D (common to the 2 groups).
`...`	Extra arguments for a plot. Do not supply unless length(whichPlots)==1.

Details

The plots display the values achievable by changing the cutoff, in comparison with the desired values as determined by NNTlower and NNTupper. The "whichPlots" options are as follows:

- "density"Marginal density of X, with rug.
- "raw"X versus class.
- "ROC"Standard ROC curve.
- "pv"Plot of ppv versus npv, with indication of the acceptable range for cutoff.
- "nnt"Plot of NNTpos versus NNTneg, with indication of the acceptable region
- "nntRange"Plot of NNTpos and NNTneg versus cutoff, with indication of the acceptable range.
  By default, all the plots are made.

---

run	*run*

Description

Run a shiny app for this package.

Usage

```
run(shinyDir)
```

Arguments

`shinyDir`	Current options are "shinyElicit" and "shinyCombinePlots". If not provided, a menu of the options is provided.

Details

The selected shiny app is run. See the vignette Using_the_NNTbiomarker_package for details, and the vignette The_Biomarker_Crisis for an overview.

---

runCombinePlots *runCombinePlots*

Description

Run a shiny app connecting a visual scale for NNT quantities and a "contra-Bayes" plot for mapping from predictive values to sensitivity/specificity (Bayes theorem in reverse).

Usage

```
runCombinePlots()
```

See Also

run

---

runElicit *runElicit*

Description

Run a shiny app outlining the process of specifying a design for a biomarker validation study.

Usage

```
runElicit()
```

See Also

run

---

sesp.from.pv *sesp.from.pv*

Description

Computes sensitivity and specificity from predictive values.

Usage

```
sesp.from.pv(ppv = 0.1, npv = 0.7, pv, prev = 0.2)
```

Arguments

ppv	Positive predictive value
npv	Negative predictive value
pv	Alternative input of ppv and npv, in matrix or named vector.
prev	Prevalence (prior probability)

Value

c(se=se,sp=sp)

---

sesp.from.pv.feasible   *sesp.from.pv.feasible*

Description

Computes sensitivity and specificity from predictive values.

Usage

```
sesp.from.pv.feasible(ppv, npv, prev, feasible = TRUE)
```

Arguments

ppv	Positive predictive value
npv	Negative predictive value
prev	Prevalence (prior probability)
feasible	Only return results in [0,1]. Default=TRUE

Details

NNT stands for Number Needed to Treat. We have a range, such that if NNT < NNTpos, all patients will be treated, if NNT > NNTneg then all patients will not be treated. Suppose N=NNTpos is the number of patients such that if N pts are positive,one will be a true positive. The "eq" means that we choose NNTpos so that treating all or not treating all would be equivalent. E(loss | treat) = (NNTpos-1) * L[A,H] = E(loss | wait) = 1 * L[W,D] Actually we choose N SMALLER so that TREATing is definitely, comfortably the right thing. E(loss | treat) = (NNTpos-1) * L[A,H] « E(loss | wait) = 1 * L[W,D] Suppose N=NNTneg is the number of patients such that if N pts are negative, one will be a false negative. The "eq" means that we choose NNTneg so that treating all or not treating any would be equivalent. E(loss | treat) = (NNTneg-1) * L[A,H] = E(loss | wait) = 1 * L[W,D] Actually we choose N LARGER so that WAITing is definitely, comfortably the right thing. E(loss | treat) = (NNTpos-1) * L[A,H] » E(loss | wait) = 1 * L[W,D]

Value

c(ppv=ppv, npv=npv, sp=sp, se=se)

setVerboseCatOption *setVerboseCatOption*

Description

Allows user to toggle on and off printing messages on a per-function basis. Should be usable in other packages, but not by importing.

Usage

```
setVerboseCatOption(fname, value)
```

Arguments

fname	Name of the function to control.
value	Boolean value: should this function print out messages?

Value

The new value of the namespace option for fname ifVerboseCat

%&% *%&% string concatenation*

Description

From mvbutils

Usage

```
a %&% b
```

Arguments

a	a string
b	another string

Value

paste0(a, b)

Index