Package ‘pssmooth’

January 22, 2019

Type Package

Title Flexible and Efficient Evaluation of Principal
Surrogates/Treatment Effect Modifiers

Version 1.0.2

BugReports https://github.com/mjuraska/pssmooth/issues

Description

Implements estimation and testing procedures for evaluating an intermediate biomarker response as a principal surrogate of a clinical response to treatment (i.e., principal stratification effect modification analysis), as described in Juraska M, Huang Y, and Gilbert PB (2018), Inference on treatment effect modification by biomarker response in a three-phase sampling design, Biostatistics, kxy074 <doi:10.1093/biostatistics/kxy074>. The methods avoid the restrictive 'placebo structural risk' modeling assumption common to past methods and further improve robustness by the use of nonparametric kernel smoothing for biomarker density estimation. A randomized controlled two-group clinical efficacy trial is assumed with an ordered categorical or continuous univariate biomarker response measured at a fixed timepoint post-randomization and with a univariate baseline surrogate measure allowed to be observed in only a subset of trial participants with an observed biomarker response (see the flexible three-phase sampling design in the paper for details). Bootstrap-based procedures are available for pointwise and simultaneous confidence intervals and testing of four relevant hypotheses. Summary and plotting functions are provided for estimation results.

License GPL-2

URL https://github.com/mjuraska/pssmooth

Encoding UTF-8

LazyData true

Imports graphics, stats, osDesign, np, chngpt, MASS

Suggests knitr, rmarkdown

RoxygenNote 6.1.1

NeedsCompilation no

Author Michal Juraska [aut, cre]

Maintainer Michal Juraska <mjuraska@fredhutch.org>

Repository CRAN

Date/Publication 2019-01-22 21:50:03 UTC

R topics documented:

`bootRiskCurve` *Bootstrap Estimation of Conditional Clinical Endpoint Risk under Placebo and Treatment Given Biomarker Response to Treatment in a Baseline Surrogate Measure Three-Phase Sampling Design*

Description

Estimates $P\{Y(z) = 1|S(1) = s_1\}$, $z = 0, 1$, on a grid of s_1 values in bootstrap resamples (see `riskCurve` for notation introduction). Cases ($Y = 1$) and controls ($Y = 0$) are sampled separately yielding a fixed number of cases and controls in each bootstrap sample. Consequentially, the number of controls with available phase 2 data varies across bootstrap samples.

Usage

```
bootRiskCurve(formula, bsm, tx, data, pstype = c("continuous",
  "ordered"), bsmtype = c("continuous", "ordered"), bwtype = c("fixed",
  "generalized_nn", "adaptive_nn"), hinge = FALSE, weights = NULL,
  psGrid = NULL, iter, seed = NULL, saveFile = NULL,
  saveDir = NULL)
```

Arguments

`formula` a formula object with the binary clinical endpoint on the left of the ~ operator. The first listed variable on the right must be the biomarker response at $t0$ and all variables that follow, if any, are discrete baseline covariates specified in all fitted models that condition on them. Interactions and transformations of the baseline covariates are allowed. All terms in the formula must be evaluable in the data frame `data`.

`bsm` a character string specifying the variable name in `data` representing the baseline surrogate measure

`tx` a character string specifying the variable name in `data` representing the treatment group indicator

`data` a data frame with one row per randomized participant endpoint-free at t_0 that contains at least the variables specified in `formula`, `bsm` and `tx`. Values of `bsm` and the biomarker at t_0 that are unavailable are represented as `NA`.

`pstype` a character string specifying whether the biomarker response shall be treated as a `continuous` (default) or `ordered` categorical variable in the kernel density/probability estimation

`bsmtype` a character string specifying whether the baseline surrogate measure shall be treated as a `continuous` (default) or `ordered` categorical variable in the kernel density/probability estimation

`bwtype` a character string specifying the bandwidth type for continuous variables in the kernel density estimation. The options are `fixed` (default) for fixed bandwidths, `generalized_nn` for generalized nearest neighbors, and `adaptive_nn` for adaptive nearest neighbors. As noted in the documentation of the function `npcdensbw` in the `np` package: "Adaptive nearest-neighbor bandwidths change with each sample realization in the set when estimating the density at the point x. Generalized nearest-neighbor bandwidths change with the point at which the density is estimated, x. Fixed bandwidths are constant over the support of x."

`hinge` a logical value (`FALSE` by default) indicating whether a hinge model (Fong et al., 2017) shall be used for modeling the effect of $S(z)$ on the clinical endpoint risk. A hinge model specifies that variability in $S(z)$ below the hinge point does not associate with the clinical endpoint risk. The hinge point is reestimated in each bootstrap sample.

`weights` either a numeric vector of weights or a character string specifying the variable name in `data` representing weights applied to observations in the phase 2 subset in order to make inference about the target population of all randomized participants endpoint-free at t_0. The weights reflect that the case:control ratio in the phase 2 subset is different from that in the target population and are passed on to GLMs in the estimation of the hinge point. If `NULL` (default and recommended), weights for cases and controls are recalculated separately in each study group *within each bootstrap sample*; otherwise the same specified vector of weights is used in each bootstrap sample.

`psGrid` a numeric vector of $S(1)$ values at which the conditional clinical endpoint risk in each study group is estimated. If `NULL` (default), a grid of values spanning the range of observed values of the biomarker will be used.

`iter` the number of bootstrap iterations

`seed` a seed of the random number generator supplied to `set.seed` for reproducibility

`saveFile` a character string specifying the name of an `.RData` file storing the output list. If `NULL` (default), the output list will only be returned.

`saveDir` a character string specifying a path for the output directory. If `NULL` (default), the output list will only be returned; otherwise, if `saveFile` is specified, the output list will also be saved as an `.RData` file in the specified directory.

Value

If `saveFile` and `saveDir` are both specified, the output list (named `bList`) is saved as an `.RData` file; otherwise it is returned only. The output object is a list with the following components:

- `psGrid`: a numeric vector of $S(1)$ values at which the conditional clinical endpoint risk is estimated in the components `plaRiskCurveBoot` and `txRiskCurveBoot`

- plaRiskCurveBoot: a length(psGrid)-by-iter matrix of estimates of $P\{Y(0) = 1|S(1) = s_1\}$ for s_1 in psGrid, with columns representing bootstrap samples
- txRiskCurveBoot: a length(psGrid)-by-iter matrix of estimates of $P\{Y(1) = 1|S(1) = s_1\}$ for s_1 in psGrid, with columns representing bootstrap samples
- cpointPboot: if hinge=TRUE, a numeric vector of estimates of the hinge point in the placebo group in each bootstrap sample
- cpointTboot: if hinge=TRUE, a numeric vector of estimates of the hinge point in the treatment group in each bootstrap sample

References

Fong, Y., Huang, Y., Gilbert, P. B., and Permar, S. R. (2017), chngpt: threshold regression model estimation and inference, *BMC Bioinformatics*, 18.

See Also

riskCurve, summary.riskCurve and plotMCEPcurve

Examples

```
n <- 500
Z <- rep(0:1, each=n/2)
S <- MASS::mvrnorm(n, mu=c(2,2,3), Sigma=matrix(c(1,0.9,0.7,0.9,1,0.7,0.7,0.7,1), nrow=3))
p <- pnorm(drop(cbind(1,Z,(1-Z)*S[,2],Z*S[,3]) %*% c(-1.2,0.2,-0.02,-0.2)))
Y <- sapply(p, function(risk){ rbinom(1,1,risk) })
X <- rbinom(n,1,0.5)
# delete S(1) in placebo recipients
S[Z==0,3] <- NA
# delete S(0) in treatment recipients
S[Z==1,2] <- NA
# generate the indicator of being sampled into the phase 2 subset
phase2 <- rbinom(n,1,0.4)
# delete Sb, S(0) and S(1) in controls not included in the phase 2 subset
S[Y==0 & phase2==0,] <- c(NA,NA,NA)
# delete Sb in cases not included in the phase 2 subset
S[Y==1 & phase2==0,1] <- NA
data <- data.frame(X,Z,S[,1],ifelse(Z==0,S[,2],S[,3]),Y)
colnames(data) <- c("X","Z","Sb","S","Y")
qS <- quantile(data$S, probs=c(0.05,0.95), na.rm=TRUE)
grid <- seq(qS[1], qS[2], length.out=3)

out <- bootRiskCurve(formula=Y ~ S + factor(X), bsm="Sb", tx="Z", data=data,
                     psGrid=grid, iter=1, seed=10)

# alternatively, to save the .RData output file (no '<-' needed):
bootRiskCurve(formula=Y ~ S + factor(X), bsm="Sb", tx="Z", data=data,
              psGrid=grid, iter=1, seed=10, saveFile="out.RData", saveDir="./")
```

plotMCEPcurve *Plotting of the Estimated Marginal Causal Effect Predictiveness Curve*

Description

Plots point estimates and, if available, pointwise and simultaneous Wald-type bootstrap confidence intervals for the specified marginal causal effect predictiveness (mCEP) curve.

Usage

```
plotMCEPcurve(object, confLevel = 0.95, hingePoint = NULL,
  title = NULL, xLab = NULL, yLab = NULL, yLim = NULL,
  pType = c("l", "p"))
```

Arguments

`object`	an object returned by `summary.riskCurve`
`confLevel`	the confidence level (0.95 by default) of pointwise and simultaneous confidence intervals
`hingePoint`	the hinge point estimate (`NULL` by default)
`title`	a character string specifying the plot title
`xLab`	a character string specifying the x-axis label (`NULL` by default)
`yLab`	a character string specifying the y-axis label (`NULL` by default)
`yLim`	a numeric vector of length 2 specifying the y-axis range (`NULL` by default)
`pType`	a character string specifying the type of plot. Possible options are `"l"` for lines (default) and `"p"` for points.

Value

None. The function is called solely for plot generation.

See Also

`riskCurve`, `bootRiskCurve` and `summary.riskCurve`

Examples

```
n <- 500
Z <- rep(0:1, each=n/2)
S <- MASS::mvrnorm(n, mu=c(2,2,3), Sigma=matrix(c(1,0.9,0.7,0.9,1,0.7,0.7,0.7,1), nrow=3))
p <- pnorm(drop(cbind(1,Z,(1-Z)*S[,2],Z*S[,3]) %*% c(-1.2,0.2,-0.02,-0.2)))
Y <- sapply(p, function(risk){ rbinom(1,1,risk) })
X <- rbinom(n,1,0.5)
# delete S(1) in placebo recipients
S[Z==0,3] <- NA
# delete S(0) in treatment recipients
```

```
S[Z==1,2] <- NA
# generate the indicator of being sampled into the phase 2 subset
phase2 <- rbinom(n,1,0.3)
# delete Sb, S(0) and S(1) in controls not included in the phase 2 subset
S[Y==0 & phase2==0,] <- c(NA,NA,NA)
# delete Sb in cases not included in the phase 2 subset
S[Y==1 & phase2==0,1] <- NA
data <- data.frame(X,Z,S[,1],ifelse(Z==0,S[,2],S[,3]),Y)
colnames(data) <- c("X","Z","Sb","S","Y")
qS <- quantile(data$S, probs=c(0.05,0.95), na.rm=TRUE)
grid <- seq(qS[1], qS[2], length.out=3)

out <- riskCurve(formula=Y ~ S + factor(X), bsm="Sb", tx="Z", data=data, psGrid=grid)
boot <- bootRiskCurve(formula=Y ~ S + factor(X), bsm="Sb", tx="Z", data=data,
                      psGrid=grid, iter=2, seed=10)
sout <- summary(out, boot, contrast="te")
plotMCEPcurve(sout)
```

riskCurve — *Estimation of Conditional Clinical Endpoint Risk under Placebo and Treatment Given Biomarker Response to Treatment in a Baseline Surrogate Measure Three-Phase Sampling Design*

Description

Estimates $P\{Y(z) = 1|S(1) = s_1\}$, $z = 0, 1$, on a grid of s_1 values following the estimation method of Juraska, Huang, and Gilbert (2018), where Z is the treatment group indicator ($Z = 1$, treatment; $Z = 0$, placebo), $S(z)$ is a continuous or ordered categorical univariate biomarker under assignment to $Z = z$ measured at fixed time t_0 after randomization, and Y is a binary clinical endpoint ($Y = 1$, disease; $Y = 0$, no disease) measured after t_0. The estimator employs the generalized product kernel density/probability estimation method of Hall, Racine, and Li (2004) implemented in the np package. The risks $P\{Y(z) = 1|S(z) = s_1, X = x\}$, $z = 0, 1$, where X is a vector of discrete baseline covariates, are estimated by fitting inverse probability-weighted logistic regression models using the osDesign package.

Usage

```
riskCurve(formula, bsm, tx, data, pstype = c("continuous", "ordered"),
  bsmtype = c("continuous", "ordered"), bwtype = c("fixed",
  "generalized_nn", "adaptive_nn"), hinge = FALSE, weights = NULL,
  psGrid = NULL, saveFile = NULL, saveDir = NULL)
```

Arguments

`formula` a formula object with the binary clinical endpoint on the left of the ~ operator. The first listed variable on the right must be the biomarker response at $t0$ and all variables that follow, if any, are discrete baseline covariates specified in all fitted models that condition on them. Interactions and transformations of the baseline covariates are allowed. All terms in the formula must be evaluable in the data frame `data`.

`bsm` a character string specifying the variable name in `data` representing the baseline surrogate measure

`tx` a character string specifying the variable name in `data` representing the treatment group indicator

`data` a data frame with one row per randomized participant endpoint-free at t_0 that contains at least the variables specified in `formula`, `bsm` and `tx`. Values of `bsm` and the biomarker at t_0 that are unavailable are represented as `NA`.

`pstype` a character string specifying whether the biomarker response shall be treated as a `continuous` (default) or `ordered` categorical variable in the kernel density/probability estimation

`bsmtype` a character string specifying whether the baseline surrogate measure shall be treated as a `continuous` (default) or `ordered` categorical variable in the kernel density/probability estimation

`bwtype` a character string specifying the bandwidth type for continuous variables in the kernel density estimation. The options are `fixed` (default) for fixed bandwidths, `generalized_nn` for generalized nearest neighbors, and `adaptive_nn` for adaptive nearest neighbors. As noted in the documentation of the function `npcdensbw` in the `np` package: "Adaptive nearest-neighbor bandwidths change with each sample realization in the set when estimating the density at the point x. Generalized nearest-neighbor bandwidths change with the point at which the density is estimated, x. Fixed bandwidths are constant over the support of x."

`hinge` a logical value (`FALSE` by default) indicating whether a hinge model (Fong et al., 2017) shall be used for modeling the effect of $S(z)$ on the clinical endpoint risk. A hinge model specifies that variability in $S(z)$ below the hinge point does not associate with the clinical endpoint risk.

`weights` either a numeric vector of weights or a character string specifying the variable name in `data` representing weights applied to observations in the phase 2 subset in order to make inference about the target population of all randomized participants endpoint-free at t_0. The weights reflect that the case:control ratio in the phase 2 subset is different from that in the target population and are passed on to GLMs in the estimation of the hinge point. If `NULL` (default), weights for cases and controls are calculated separately in each study group.

`psGrid` a numeric vector of $S(1)$ values at which the conditional clinical endpoint risk in each study group is estimated. If `NULL` (default), a grid of values spanning the range of observed values of the biomarker will be used.

`saveFile` a character string specifying the name of an `.RData` file storing the output list. If `NULL` (default), the output list will only be returned.

`saveDir` a character string specifying a path for the output directory. If `NULL` (default), the output list will only be returned; otherwise, if `saveFile` is specified, the output list will also be saved as an `.RData` file in the specified directory.

Value

If saveFile and saveDir are both specified, the output list (named oList) is saved as an .RData file; otherwise it is returned only. The output object (of class riskCurve) is a list with the following components:

- psGrid: a numeric vector of $S(1)$ values at which the conditional clinical endpoint risk is estimated in the components plaRiskCurve and txRiskCurve
- plaRiskCurve: a numeric vector of estimates of $P\{Y(0) = 1|S(1) = s_1\}$ for s_1 in psGrid
- txRiskCurve: a numeric vector of estimates of $P\{Y(1) = 1|S(1) = s_1\}$ for s_1 in psGrid
- fOptBandwidths: a conbandwidth object returned by the call of the function npcdensbw containing the optimal bandwidths, selected by likelihood cross-validation, in the kernel estimation of the conditional density of $S(1)$ given the baseline surrogate measure and any other specified baseline covariates
- gOptBandwidths: a conbandwidth object returned by the call of the function npcdensbw or npudensbw containing the optimal bandwidths, selected by likelihood cross-validation, in the kernel estimation of the conditional density of $S(0)$ given any specified baseline covariates or the marginal density of $S(0)$ if no baseline covariates are specified in formula
- cpointP: if hinge=TRUE, the estimate of the hinge point in the placebo group
- cpointT: if hinge=TRUE, the estimate of the hinge point in the treatment group

References

Fong, Y., Huang, Y., Gilbert, P. B., and Permar, S. R. (2017), chngpt: threshold regression model estimation and inference, *BMC Bioinformatics*, 18.

Hall, P., Racine, J., and Li, Q. (2004), Cross-validation and the estimation of conditional probability densities, *JASA* 99(468), 1015-1026.

Juraska, M., Huang, Y., and Gilbert, P. B. (2018), Inference on treatment effect modification by biomarker response in a three-phase sampling design, Biostatistics, kxy074, https://doi.org/10.1093/biostatistics/kxy074.

See Also

bootRiskCurve, summary.riskCurve and plotMCEPcurve

Examples

```
n <- 500
Z <- rep(0:1, each=n/2)
S <- MASS::mvrnorm(n, mu=c(2,2,3), Sigma=matrix(c(1,0.9,0.7,0.9,1,0.7,0.7,0.7,1), nrow=3))
p <- pnorm(drop(cbind(1,Z,(1-Z)*S[,2],Z*S[,3]) %*% c(-1.2,0.2,-0.02,-0.2)))
Y <- sapply(p, function(risk){ rbinom(1,1,risk) })
X <- rbinom(n,1,0.5)
# delete S(1) in placebo recipients
S[Z==0,3] <- NA
# delete S(0) in treatment recipients
S[Z==1,2] <- NA
# generate the indicator of being sampled into the phase 2 subset
```

```
phase2 <- rbinom(n,1,0.4)
# delete Sb, S(0) and S(1) in controls not included in the phase 2 subset
S[Y==0 & phase2==0,] <- c(NA,NA,NA)
# delete Sb in cases not included in the phase 2 subset
S[Y==1 & phase2==0,1] <- NA
data <- data.frame(X,Z,S[,1],ifelse(Z==0,S[,2],S[,3]),Y)
colnames(data) <- c("X","Z","Sb","S","Y")
qS <- quantile(data$S, probs=c(0.05,0.95), na.rm=TRUE)
grid <- seq(qS[1], qS[2], length.out=3)

out <- riskCurve(formula=Y ~ S + factor(X), bsm="Sb", tx="Z", data=data, psGrid=grid)

# alternatively, to save the .RData output file (no '<-' needed):
riskCurve(formula=Y ~ S + factor(X), bsm="Sb", tx="Z", data=data, saveFile="out.RData",
          saveDir="./")
```

summary.riskCurve *Summary of Point and Interval Estimation of a Marginal Causal Effect Predictiveness Curve*

Description

Summarizes point estimates and pointwise and simultaneous Wald-type bootstrap confidence intervals for a specified marginal causal effect predictiveness (mCEP) curve (see, e.g., Juraska, Huang, and Gilbert (2018) for the definition).

Usage

```
## S3 method for class 'riskCurve'
summary(object, boot = NULL, contrast = c("te",
  "rr", "logrr", "rd"), confLevel = 0.95, ...)
```

Arguments

object	an object of class riskCurve, typically returned by riskCurve
boot	an object returned by bootRiskCurve. If NULL (default), only point estimates are reported.
contrast	a character string specifying the mCEP curve. It must be one of te (treatment efficacy), rr (relative risk), logrr (log relative risk), and rd (risk difference [placebo minus treatment]).
confLevel	the confidence level of pointwise and simultaneous confidence intervals
...	for other methods

Value

A data frame containing point and possibly interval estimates of the specified mCEP curve.

References

Juraska, M., Huang, Y., and Gilbert, P. B. (2018), Inference on treatment effect modification by biomarker response in a three-phase sampling design, Biostatistics, kxy074, https://doi.org/10.1093/biostatistics/kxy074.

See Also

riskCurve and bootRiskCurve

Examples

```
n <- 500
Z <- rep(0:1, each=n/2)
S <- MASS::mvrnorm(n, mu=c(2,2,3), Sigma=matrix(c(1,0.9,0.7,0.9,1,0.7,0.7,0.7,1), nrow=3))
p <- pnorm(drop(cbind(1,Z,(1-Z)*S[,2],Z*S[,3]) %*% c(-1.2,0.2,-0.02,-0.2)))
Y <- sapply(p, function(risk){ rbinom(1,1,risk) })
# delete S(1) in placebo recipients
S[Z==0,3] <- NA
# delete S(0) in treatment recipients
S[Z==1,2] <- NA
# generate the indicator of being sampled into the phase 2 subset
phase2 <- rbinom(n,1,0.4)
# delete Sb, S(0) and S(1) in controls not included in the phase 2 subset
S[Y==0 & phase2==0,] <- c(NA,NA,NA)
# delete Sb in cases not included in the phase 2 subset
S[Y==1 & phase2==0,1] <- NA
data <- data.frame(Z,S[,1],ifelse(Z==0,S[,2],S[,3]),Y)
colnames(data) <- c("Z","Sb","S","Y")
qS <- quantile(data$S, probs=c(0.05,0.95), na.rm=TRUE)
grid <- seq(qS[1], qS[2], length.out=2)

out <- riskCurve(formula=Y ~ S, bsm="Sb", tx="Z", data=data, psGrid=grid)
boot <- bootRiskCurve(formula=Y ~ S, bsm="Sb", tx="Z", data=data,
                      psGrid=grid, iter=2, seed=10)
summary(out, boot, contrast="te")
```

testConstancy *Testing of the Null Hypotheses of a Flat and a Constant Marginal Causal Effect Predictiveness Curve*

Description

Computes a two-sided p-value either from the test of $\{H_0^1 : mCEP(s_1) = CE$ for all $s_1\}$, where CE is the overall causal treatment effect on the clinical endpoint, or from the test of $\{H_0^2 : mCEP(s_1) = c$ for all s_1 in the interval limS1 and a specified constant $c\}$, each against a general alternative hypothesis. The testing procedures are described in Juraska, Huang, and Gilbert (2018) and are based on the simultaneous estimation method of Roy and Bose (1953).

Usage

```
testConstancy(object, boot, contrast = c("te", "rr", "logrr", "rd"),
  null = c("H01", "H02"), overallPlaRisk = NULL,
  overallTxRisk = NULL, MCEPconstantH02 = NULL, limS1 = NULL)
```

Arguments

`object` an object returned by `riskCurve`

`boot` an object returned by `bootRiskCurve`

`contrast` a character string specifying the mCEP curve. It must be one of `te` (treatment efficacy), `rr` (relative risk), `logrr` (log relative risk), and `rd` (risk difference [placebo minus treatment]).

`null` a character string specifying the null hypothesis to be tested; one of `H01` and `H02` as introduced above

`overallPlaRisk` a numeric value of the estimated overall clinical endpoint risk in the placebo group. It is required when `null` equals `H01`.

`overallTxRisk` a numeric value of the estimated overall clinical endpoint risk in the treatment group. It is required when `null` equals `H01`.

`MCEPconstantH02`
the constant c in the null hypothesis H_0^2. It is required when `null` equals `H02`.

`limS1` a numeric vector of length 2 specifying an interval that is a subset of the support of $S(1)$ and that is used in the evaluation of the null hypothesis H_0^2. If `NULL` (default), then H_0^2 is evaluated for all s_1.

Value

A numeric value representing the two-sided p-value from the test of either H_0^1 or H_0^2.

References

Juraska, M., Huang, Y., and Gilbert, P. B. (2018), Inference on treatment effect modification by biomarker response in a three-phase sampling design, Biostatistics, kxy074, https://doi.org/10.1093/biostatistics/kxy074.

Roy, S. N. and Bose, R. C. (1953), Simultaneous confidence interval estimation, *The Annals of Mathematical Statistics*, 24, 513-536.

See Also

`riskCurve`, `bootRiskCurve` and `testEquality`

Examples

```
n <- 500
Z <- rep(0:1, each=n/2)
S <- MASS::mvrnorm(n, mu=c(2,2,3), Sigma=matrix(c(1,0.9,0.7,0.9,1,0.7,0.7,0.7,1), nrow=3))
p <- pnorm(drop(cbind(1,Z,(1-Z)*S[,2],Z*S[,3]) %*% c(-1.2,0.2,-0.02,-0.2)))
Y <- sapply(p, function(risk){ rbinom(1,1,risk) })
```

```
X <- rbinom(n,1,0.5)
# delete S(1) in placebo recipients
S[Z==0,3] <- NA
# delete S(0) in treatment recipients
S[Z==1,2] <- NA
# generate the indicator of being sampled into the phase 2 subset
phase2 <- rbinom(n,1,0.4)
# delete Sb, S(0) and S(1) in controls not included in the phase 2 subset
S[Y==0 & phase2==0,] <- c(NA,NA,NA)
# delete Sb in cases not included in the phase 2 subset
S[Y==1 & phase2==0,1] <- NA
data <- data.frame(X,Z,S[,1],ifelse(Z==0,S[,2],S[,3]),Y)
colnames(data) <- c("X","Z","Sb","S","Y")
qS <- quantile(data$S, probs=c(0.05,0.95), na.rm=TRUE)
grid <- seq(qS[1], qS[2], length.out=3)

out <- riskCurve(formula=Y ~ S + factor(X), bsm="Sb", tx="Z", data=data, psGrid=grid)
boot <- bootRiskCurve(formula=Y ~ S + factor(X), bsm="Sb", tx="Z", data=data,
                      psGrid=grid, iter=2, seed=10)
fit <- glm(Y ~ Z, data=data, family=binomial)
prob <- predict(fit, newdata=data.frame(Z=0:1), type="response")

testConstancy(out, boot, contrast="te", null="H01", overallPlaRisk=prob[1],
              overallTxRisk=prob[2])
testConstancy(out, boot, contrast="te", null="H02", MCEPconstantH02=0, limS1=c(qS[1],1.5))
```

testEquality — *Testing of the Null Hypothesis of Equal Marginal Causal Effect Predictiveness Curves for Two Biomarkers, Endpoints, or Baseline Covariate Subgroups*

Description

Computes a two-sided p-value either from the test of $\{H_0^3 : mCEP_1(s_1) = mCEP_2(s_1)$ for all s_1 in `limS1`$\}$, where $mCEP_1$ and $mCEP_2$ are each associated with either a different biomarker (measured in the same units) or a different endpoint or both, or from the test of $\{H_0^4 : mCEP(s_1|X = 0) = mCEP(s_1|X = 1)$ for all s_1 in `limS1`$\}$, where X is a baseline dichotomous phase 1 covariate of interest, each against a general alternative hypothesis. The testing procedures are described in Juraska, Huang, and Gilbert (2018) and are based on the simultaneous estimation method of Roy and Bose (1953).

Usage

```
testEquality(object1, object2, boot1, boot2, contrast = c("te", "rr",
  "logrr", "rd"), null = c("H03", "H04"), limS1 = NULL)
```

Arguments

`object1` an object returned by `riskCurve` pertaining to either $mCEP_1(s_1)$ in H_0^3 or $mCEP(s1|X = 0)$ in H_0^4

`object2` an object returned by `riskCurve` pertaining to either $mCEP_2(s_1)$ in H_0^3 or $mCEP(s1|X = 1)$ in H_0^4

`boot1` an object returned by `bootRiskCurve` pertaining to either $mCEP_1(s_1)$ in H_0^3 or $mCEP(s1|X = 0)$ in H_0^4

`boot2` an object returned by `bootRiskCurve` pertaining to either $mCEP_2(s_1)$ in H_0^3 or $mCEP(s1|X = 1)$ in H_0^4

`contrast` a character string specifying the mCEP curve. It must be one of `te` (treatment efficacy), `rr` (relative risk), `logrr` (log relative risk), and `rd` (risk difference [placebo minus treatment]).

`null` a character string specifying the null hypothesis to be tested; one of `H03` and `H04` as introduced above

`limS1` a numeric vector of length 2 specifying an interval that is a subset of the support of $S(1)$. If `NULL` (default), then the specified null hypothesis is evaluated for all s_1.

Value

A numeric value representing the two-sided p-value from the test of either H_0^3 or H_0^4.

References

Juraska, M., Huang, Y., and Gilbert, P. B. (2018), Inference on treatment effect modification by biomarker response in a three-phase sampling design, Biostatistics, kxy074, https://doi.org/10.1093/biostatistics/kxy074.

Roy, S. N. and Bose, R. C. (1953), Simultaneous confidence interval estimation, *The Annals of Mathematical Statistics*, 24, 513-536.

See Also

`riskCurve`, `bootRiskCurve` and `testConstancy`

Examples

```
n <- 500
Z <- rep(0:1, each=n/2)
S <- MASS::mvrnorm(n, mu=c(2,2,3), Sigma=matrix(c(1,0.9,0.7,0.9,1,0.7,0.7,0.7,1), nrow=3))
p <- pnorm(drop(cbind(1,Z,(1-Z)*S[,2],Z*S[,3]) %*% c(-1.2,0.2,-0.02,-0.2)))
Y <- sapply(p, function(risk){ rbinom(1,1,risk) })
X <- rbinom(n,1,0.5)
# delete S(1) in placebo recipients
S[Z==0,3] <- NA
# delete S(0) in treatment recipients
S[Z==1,2] <- NA
# generate the indicator of being sampled into the phase 2 subset
phase2 <- rbinom(n,1,0.4)
```

```
# delete Sb, S(0) and S(1) in controls not included in the phase 2 subset
S[Y==0 & phase2==0,] <- c(NA,NA,NA)
# delete Sb in cases not included in the phase 2 subset
S[Y==1 & phase2==0,1] <- NA
data <- data.frame(X,Z,S[,1],ifelse(Z==0,S[,2],S[,3]),Y)
colnames(data) <- c("X","Z","Sb","S","Y")
qS <- quantile(data$S, probs=c(0.05,0.95), na.rm=TRUE)
grid <- seq(qS[1], qS[2], length.out=3)
out0 <- riskCurve(formula=Y ~ S, bsm="Sb", tx="Z", data=data[data$X==0,], psGrid=grid)
out1 <- riskCurve(formula=Y ~ S, bsm="Sb", tx="Z", data=data[data$X==1,], psGrid=grid)
boot0 <- bootRiskCurve(formula=Y ~ S, bsm="Sb", tx="Z", data=data[data$X==0,],
                       psGrid=grid, iter=2, seed=10)
boot1 <- bootRiskCurve(formula=Y ~ S, bsm="Sb", tx="Z", data=data[data$X==1,],
                       psGrid=grid, iter=2, seed=15)

testEquality(out0, out1, boot0, boot1, contrast="te", null="H04")
```

Index