Package: optimalThreshold (via r-universe)

October 23, 2024

Type Package

Title Bayesian Methods for Optimal Threshold Estimation

Version 1.0

SystemRequirements JAGS 4.x.y

Depends R (>= 3.1.2)

Date 2020-01-09

Description Functions to estimate the optimal threshold of diagnostic markers or treatment selection markers. The optimal threshold is the marker value that maximizes the utility of the marker based-strategy (for diagnostic or treatment selection) in a given population. The utility function depends on the type of marker (diagnostic or treatment selection), but always takes into account the preferences of the patients or the physician in the decision process. For estimating the optimal threshold, ones must specify the distributions of the marker in different groups (defined according to the type of marker, diagnostic or treatment selection) and provides data to estimate the parameters of these distributions. Ones must also provide some features of the target populations (disease prevalence or treatment efficacies) as well as the preferences of patients or physicians. The functions rely on Bayesian inference which helps producing several indicators derived from the optimal threshold. See Blangero, Y, Rabilloud, M, Ecochard, R, and Subtil, F (2019) <doi:10.1177/0962280218821394> for the original article that describes the estimation method for treatment selection markers and Subtil, F, and Rabilloud, M (2019) <doi:10.1002/bimj.200900242> for diagnostic markers.

License GPL (>= 2.0)

Collate Package.R import_package.R scaledT.R ClassUnions.R ClassFitNormalDist.R ClassFitLogNormalDist.R ClassFitGammaDist.R ClassFitStudentDist.R ClassFitLogisticDist.R ClassFitUserDefinedDist.R ClassUndefined.R ClassNormalDist.R ClassLogNormalDist.R ClassGammaDist.R ClassStudentDist.R ClassLogisticDist.R

ClassUnionsDist.R ClassCompoundDist.R ClassTrtSelOptThresh.R
ClassTrtSelRelUtility.R ClassDiagOptThresh.R
ClassDiagRelUtility.R cdf.R gradient.R hessian.R ARS.R
samplePosteriorDist.R global.R

RoxygenNote 6.1.1

Imports ars, rjags, HDInterval, mgcv, utils, coda, grDevices, methods, stats, graphics

NeedsCompilation no

Author Yoann Blangero [aut, cre]

Maintainer Yoann Blangero <yblangero@gmail.com>

Repository CRAN

Date/Publication 2020-01-13 16:10:05 UTC

Contents

allowedDist-class . . . 3
allowedFitDist-class . . . 3
cdf . . . 4
compoundEvtInnovDist-class . . . 6
compoundEvtRefDist-class . . . 6
compoundNoEvtInnovDist-class . . . 7
compoundNoEvtRefDist-class . . . 8
credibleIntervals . . . 9
decisionCurve . . . 10
densCurves . . . 11
diagOptThresh-class . . . 12
diagRelUtility-class . . . 13
diagThresh . . . 14
estimates . . . 16
fit . . . 17
fitGammaDist-class . . . 21
fitLogisticDist-class . . . 21
fitLogNormalDist-class . . . 22
fitNormalDist-class . . . 23
fitStudentDist-class . . . 23
fitUserDefinedDist-class . . . 24
gammaDist-class . . . 25
gradient . . . 25
hessian . . . 27
logisticDist-class . . . 29
logNormalDist-class . . . 30
mcmc.listOrNull-class . . . 30
normalDist-class . . . 30
plot-methods . . . 31
plot.diagRelUtility . . . 32
plot.trtSelRelUtility . . . 33

riskCurves 34
samplePosteriorDist 36
show-methods 37
studentDist-class 38
summary-methods 39
trtSelOptThresh-class 40
trtSelRelUtility-class 42
trtSelThresh 42
undefined-class 45

Index **47**

`allowedDist-class` *An S4 union class to represent allowed dist object.*

Description

This S4 union class describes what types of distribution are supported by the `optimalThreshold` package.

Details

Five theoretical univariate and continuous distributions are supported by the package internally (normal, log-normal, gamma, scaled t, and logistic). However, it is possible for the user to define its own distribution type using the `fit` function, and passing the result to one of the two main functions: `trtSelThresh`, and `diagThresh`.

See Also

`fit`, `trtSelThresh`, and `diagThresh`.

`allowedFitDist-class` *An S4 union class to represent allowed fitDist objects.*

Description

This S4 union class describes what types of distribution may be fitted by the user in the `optimalThreshold` package.

Details

Five theoretical types of distribution fit are supported by the package internally (normal, log-normal, gamma, scaled t, and logistic). However, it is possible for the user to fit a personalized distribution using the `fit` function, and passing the result to one of the two main functions: `trtSelThresh`, and `diagThresh`. The 'undefined' type is used to indicate which distribution must be expressed as a function of the three others when estimating the optimal threshold of a treatment selection marker (see References for more details).

References

Blangero, Y, Rabilloud, M, Ecochard, R, and Subtil, F. A Bayesian method to estimate the optimal threshold of a marker used to select patients' treatment. *Statistical Methods in Medical Research*. 2019. Subtil, F, and Rabilloud, M. A Bayesian method to estimate the optimal threshold of a longitudinal marker. *Biometrical Journal*. 2010.

See Also

`fit`, `trtSelThresh`, and `diagThresh`.

`cdf` *Cumulative distribution function of a specified distribution*

Description

The `cdf` function returns the cumulative distribution function relative to the S4 object passed in its argument. See details to know on what kind of S4 objects this function could be applied.

Usage

```
cdf(object)

## S4 method for signature 'normalDist'
cdf(object)

## S4 method for signature 'logNormalDist'
cdf(object)

## S4 method for signature 'gammaDist'
cdf(object)

## S4 method for signature 'studentDist'
cdf(object)

## S4 method for signature 'logisticDist'
cdf(object)

## S4 method for signature 'compoundEvtRefDist'
cdf(object)

## S4 method for signature 'compoundNoEvtRefDist'
cdf(object)

## S4 method for signature 'compoundEvtInnovDist'
cdf(object)

## S4 method for signature 'compoundNoEvtInnovDist'
cdf(object)
```

Arguments

`object`	Any S4 object for which a `cdf` method is defined. Should match with the definition of an S4 distribution object as defined in the `optimalThreshold` package.

Details

This method can be applied to the S4 distribution objects that are supported in the `optimalThreshold` package: `normalDist`, `logNormalDist`, `gammaDist`, `studentDist`, `logisticDist`, and `userDefinedDist`. These methods are applied internally, and you have no need to use it outside of the main functions `trtSelThresh` and `diagThresh`.

- Normal distribution: the `cdf` method applied to a `normalDist` object is simply the `pnorm` function (see help on this function to have more details).
- Log-normal distribution: the `cdf` method applied to a `logNormalDist` object is simply the `plnorm` function (see help on this function to have more details).
- Gamma distribution: the `cdf` method applied to a `gammaDist` object is simply the `pgamma` function (see help on this function to have more details).
- Scaled t distribution: the scaled t distribution with `df` = n, `mu` = μ, and `sd` = σ has density:

$$f(x) = (\Gamma((n+1)/2)/(\sqrt{n\pi}\Gamma(n/2))(1 + ((x-\mu)/\sigma)^2/n)^{-}((n+1)/2))/\sigma$$

- Logistic distribution: the `cdf` method applied to a `logisticDist` object is simply the `plogis` function (see help on this function to have more details).
- User-defined distribution: the `cdf` method applied to a `userDefinedDist` object is simply the cumulative distribution function provided by the user when fitting a user-defined distribution with the `fit` function.

The S4 objects `compoundEvtRefDist`, `compoundNoEvtRefDist`, `compoundEvtInnovDist`, and `compoundNoEvtInnovDist` are created internally. The `cdf` function applied to these objects is defined dynamically depending on what types of distribution are fitted. The definition of the `cdf` function relies on the expression of the randomization constraint of a clinical trial that enforces the distribution of the marker in each treatment arm to be identical (see References for more details).

Value

Returns the cumulative distribution function of the specified distribution.

References

Blangero, Y, Rabilloud, M, Ecochard, R, and Subtil, F. A Bayesian method to estimate the optimal threshold of a marker used to select patients' treatment. *Statistical Methods in Medical Research*. 2019.

See Also

`trtSelThresh`, `diagThresh`, `pnorm`, `plnorm`, `pgamma`, `plogis`, `fit`.

`compoundEvtInnovDist-class`

An S4 class to represent a compound distribution.

Description

This S4 class describes the 'compound' distribution that is fitted to the marker values of patients that developed the event of interest in the innovative treatment arm when the type of distribution is set to 'undefined' in the `fit` function.

Details

You never have to create this class manually. This class is created internally when an undefined distribution is fitted on the marker values.

Slots

`EvtRefDist` This slot is an object that describes the distribution of the marker for patients that developed the event in the reference arm. This object must be an "allowedDist" class object.

`NoEvtRefDist` This slot is an object that describes the distribution of the marker for patients that did not develop the event in the reference arm. This object must be an "allowedDist" class object.

`NoEvtInnovDist` This slot is an object that describes the distribution of the marker for patients that did not develop the event in the innovative arm. This object must be an "allowedDist" class object.

`r0` Mean risk of event occurrence in the reference arm. Numeric argument.

`r1` Mean risk of event occurrence in the innovative arm. Numeric argument.

See Also

`fit` for more details on how to fit an undefined distribution.

`compoundEvtRefDist-class`

An S4 class to represent a compound distribution.

Description

This S4 class describes the 'compound' distribution that is fitted to the marker values of patients that developed the event of interest in the reference treatment arm when the type of distribution is set to 'undefined' in the `fit` function.

Details

You never have to create this class manually. This class is created internally when an undefined distribution is fitted on the marker values.

Slots

`NoEvtRefDist` This slot is an object that describes the distribution of the marker for patients that did not develop the event in the reference arm. This object must be an "allowedDist" class object.

`EvtInnovDist` This slot is an object that describes the distribution of the marker for patients that developed the event in the innovative arm. This object must be an "allowedDist" class object.

`NoEvtInnovDist` This slot is an object that describes the distribution of the marker for patients that did not develop the event in the innovative arm. This object must be an "allowedDist" class object.

`r0` Mean risk of event occurrence in the reference arm. Numeric argument.

`r1` Mean risk of event occurrence in the innovative arm. Numeric argument.

See Also

`fit` for more details on how to fit an undefined distribution.

---

compoundNoEvtInnovDist-class

An S4 class to represent a compound distribution.

Description

This S4 class describes the 'compound' distribution that is fitted to the marker values of patients that did not develop the event of interest in the innovative treatment arm when the type of distribution is set to 'undefined' in the `fit` function.

Details

You never have to create this class manually. This class is created internally when an undefined distribution is fitted on the marker values.

Slots

`EvtRefDist` This slot is an object that describes the distribution of the marker for patients that developed the event in the reference arm. This object must be an "allowedDist" class object.

`NoEvtRefDist` This slot is an object that describes the distribution of the marker for patients that did not develop the event in the reference arm. This object must be an "allowedDist" class object.

`EvtInnovDist` This slot is an object that describes the distribution of the marker for patients that developed the event in the innovative arm. This object must be an "allowedDist" class object.

`r0` Mean risk of event occurrence in the reference arm. Numeric argument.

`r1` Mean risk of event occurrence in the innovative arm. Numeric argument.

See Also

`fit` for more details on how to fit an undefined distribution.

`compoundNoEvtRefDist-class`
An S4 class to represent a compound distribution.

Description

This S4 class describes the 'compound' distribution that is fitted to the marker values of patients that did not develop the event of interest in the reference treatment arm when the type of distribution is set to 'undefined' in the `fit` function.

Details

You never have to create this class manually. This class is created internally when an undefined distribution is fitted on the marker values.

Slots

`EvtRefDist` This slot is an object that describes the distribution of the marker for patients that developed the event in the reference arm. This object must be an "allowedDist" class object.

`EvtInnovDist` This slot is an object that describes the distribution of the marker for patients that developed the event in the innovative arm. This object must be an "allowedDist" class object.

`NoEvtInnovDist` This slot is an object that describes the distribution of the marker for patients that did not develop the event in the innovative arm. This object must be an "allowedDist" class object.

`r0` Mean risk of event occurrence in the reference arm. Numeric argument.

`r1` Mean risk of event occurrence in the innovative arm. Numeric argument.

See Also

`fit` for more details on how to fit an undefined distribution.

credibleIntervals *Credible intervals estimation*

Description

Credible intervals estimation

Usage

```
credibleIntervals(object, ...)

## S4 method for signature 'trtSelOptThresh'
credibleIntervals(object, alpha = 0.05,
  hpd = FALSE)

## S4 method for signature 'diagOptThresh'
credibleIntervals(object, alpha = 0.05,
  hpd = FALSE)
```

Arguments

object	a trtSelOptThresh or a diagOptThresh S4 class object from which the credible intervals of several indicators (including the optimal threshold) must be calculated.
...	other arguments passed to the method.
alpha	alpha parameter for the confidence level required.
hpd	logical value to specify whether the function has to return Highest Posterior Density interval or not.

Details

This function calculates the credible intervals of several indicators depending on the type of marker under study (treatment selection or diagnostic marker). The user may specify the alpha risk for the confidence level (default is 5

Value

Returns a matrix with the credible intervals of several indicators.

Returns the credible intervals of several indicators.

References

Gelman, A, et al. 2014. *Bayesian Data Analysis*. 3rd edition, CRC Press, Boca Raton, section 2.3.

See Also

trtSelThresh, diagThresh, and hdi.

`decisionCurve` *Decision curve plot*

Description

This S4 method allows to plot the decision curve associated with a treatment selection marker for several treatment/event cost ratios, or with a diagnostic marker for several risk thresholds.

Usage

```
decisionCurve(object, r, ...)

## S4 method for signature 'trtSelOptThresh'
decisionCurve(object, r, alpha = 0.05)

## S4 method for signature 'diagOptThresh'
decisionCurve(object, r)
```

Arguments

`object`	a `trtSelOptThresh` or a `diagOptThresh` S4 class object from which the decision curve must be plotted.
`r`	Ratio of treatment/event costs (for treatment selection markers), or risk preference (for diagnostic marker). Numeric argument.
`...`	other arguments passed to methods.
`alpha`	alpha parameter for the confidence level required.

Details

This function plots the decision curves according to the type of marker under study (treatment selection or diagnostic) and the `r` argument. For treatment selection markers, it plots two graphs: the first one is a classical decision curves graph comparing the utilities of: the marker under study, the perfect marker, the strategy 'Treat everyone with the reference treatment', and the strategy 'Treat everyone with the innovative treatment'; the second one is the relative decision curve that plots the relative utility of the marker under study (0 meaning that using the marker to guide treatment decisions is not better than treating everyone with the overall best treatment, and 1 meaning that the marker under study has the same utility as the perfect marker). The decision curves are calculated using the mean risk of event in each treatment arm provided in the `trtSelOptThresh` object. For diagnostic markers it calculates the expected benefit of the marker and compares it with the strategies "Treat everyone" and "Do not treat anyone". The decision curve is calculated for a population with a disease prevalence provided in the `diagOptThresh` object.

Value

Returns an object of class `trtSelRelUtility` if applied to a `trtSelOptThresh` object, and an object of class `diagRelUtility` if applied to a `diagOptThresh` object.

References

Blangero, Y, Rabilloud, M, Ecochard, R, and Subtil, F. A Bayesian method to estimate the optimal threshold of a marker used to select patients' treatment. *Statistical Methods in Medical Research*. 2019. Huang, Y, Laber, EB, and Janes H. Characterizing expected benefits of biomarkers in treatment selection. *Biostatistics*. 2015; 16(2): 383-399. Vickers, AJ, and Elkin, EB. Decision curve analysis: a novel method for evaluating prediction models. *Medical Decision Making*. 2006; 26(6): 565-574.

See Also

`trtSelThresh` and `diagThresh` for some examples on how to use this function.

`densCurves` *Density curves plot*

Description

This function plots the density curves of the marker values in each treatment arm (treatment selection marker), or for diseased and non-diseased patients (diagnostic marker).

Usage

```
densCurves(x0, x1, type, ylab = "Density", xlab = "Marker values",
  main = "Density curves", col0 = "blue", col1 = "green", lty0 = 1,
  lty1 = 1, pos.legend = "topright", ...)
```

Arguments

`x0`	a numeric vector containing the marker values of patients in the reference arm (treatment selection marker), or non-diseased patients (diagnostic marker).
`x1`	a numeric vector containing the marker values of patients in the innovative arm (treatment selection marker), or diseased patients (diagnostic marker).
`type`	a character argument that specifies the type of the marker ("treatment selection" for treatment selection marker or "diagnostic" for diagnostic marker).
`ylab`	label of the Y-axis.
`xlab`	label of the X-axis.
`main`	title of the graph.
`col0`	color of the density curve in the reference arm (treatment selection marker), or for non-diseased patients (diagnostic marker).
`col1`	color of the density curve in the innovative arm (treatment selection marker), or for diseased patients (diagnostic marker).
`lty0`	type of the line for the density curve in the reference arm (treatment selection marker), or for non-diseased patients (diagnostic marker).

lty1	type of the line for the density curve in the innovative arm (treatment selection marker), or for diseased patients (diagnostic marker).
pos.legend	the x and y co-ordinates to be used to position the legend. They can be specified by keyword or in any way which is accepted by xy.coords.
...	other arguments to be passed to the plot function.

Details

When assessing treatment selection markers and estimating their optimal threshold, it is necessary that the randomization constraint be respected. If the density curves of the marker are different when comparing the two treatment arms, then it is likely that the trtSelThresh function will provide a threshold that do not correspond to the true optimal threshold. When assessing diagnostic markers, it is necessary to define the decision rule (classically high values of the marker are associated with a worst outcome). This decision rule may be checked with the density curves of the marker for diseased and non-diseased patients.

Value

None

Examples

```
### Plotting density curves for a treatment selection marker
# Data generation
x0E <- rnorm(100, 2, 1)
x0Eb <- rnorm(200, 4, 1)
x1E <- rnorm(100, 4, 1)
x1Eb <- rnorm(200, 2, 1)
densCurves(x0 = c(x0E, x0Eb), x1 = c(x1E, x1Eb), type = "treatment selection")

### Plotting density curves for a diagnostic marker
# Data generation
xE <- rnorm(30, 3, 1)
xEb <- rnorm(90, 1, 1)
densCurves(x0 = xEb, x1 = xE, type = "diagnostic")
```

---

diagOptThresh-class *An S4 class to describe the optimal threshold of a diagnostic marker.*

Description

An S4 class to describe the optimal threshold of a diagnostic marker.

Details

You never have to create this class manually. This class is created internally when the diagThresh function is used.

Slots

`optThresh` This slot is an object that takes in argument the sampled optimal threshold values. Numeric argument.

`p` Prevalence specified by the user. Numeric argument.

`r` Risk threshold preference. Numeric argument.

`N` Sample size.

`xEvt` This slot is an object that takes in argument the marker values in the subgroup of patients that developed the event. Numeric argument.

`xNoEvt` This slot is an object that takes in argument the marker values in the subgroup of patients that did not develop the event. Numeric argument.

`lowEvt` This slot is a logical argument that specifies whether the low values of the marker are associated with the presence of the disease or not.

`mcmcChainEvt` This slot is an object that takes in argument the sampled distribution objects in the subgroup of patients that developed the event. list argument.

`mcmcChainNoEvt` This slot is an object that takes in argument the sampled distribution objects in the subgroup of patients that did not develop the event. list argument.

`tabMCMCChain` This slot is an object that takes in argument all the distribution parameters that were sampled using the MCMC algorithm. mcmc.listOrNull argument.

`paraNamesUserDefined` This slot is an object that takes in argument the list of the distribution parameter names defined by the user in a 'fitUserDefinedDist' object. list argument.

`cdfUserDefined` This slot is an object that takes in argument the list of cumulative distribution functions defined by the user in 'fitUserDefinedDist' objects. list argument.

`gradientUserDefined` This slot is an object that takes in argument the list of gradient functions defined by the user in 'fitUserDefinedDist' objects. list argument.

`hessianUserDefined` This slot is an object that takes in argument the list of hessian functions defined by the user in 'fitUserDefinedDist' objects. list argument.

`percentNA` This slot is a numeric object that indicates the percentage of NA values contained in the 'optThresh' slot.

See Also

`diagThresh` for more details on how to estimate the optimal threshold of a diagnostic marker.

`diagRelUtility-class` *An S4 class to sum up the results from the decisionCurve methods.*

Description

An S4 class to sum up the results from the decisionCurve methods.

Details

You never have to create this class manually. This class is created internally when the `decisionCurve` method is applied to a 'diagOptThresh' object.

Slots

`U` This slot is a matrix of the marker-based utility according to `r`.

`UNoTreat` This slot is a matrix of the utility of the 'No Treat' strategy according to `r`.

`UTreatAll` This slot is a matrix of the utility of the 'Treat All' strategy according to `r`.

`r` Risk threshold preference. Numeric argument.

See Also

`decisionCurve` for more details on how to plot the decision curves.

---

diagThresh *Estimation of the optimal threshold of a diagnostic marker*

Description

This function produces a sample of the posterior distribution of the optimal threshold of a diagnostic marker. The optimal threshold is defined as the marker value that maximized the utility of using the marker to make the diagnosis and treat the patient (treat or not the patient). The utility function takes into account the proportions of patients well classified and miss-classified (through the sensitivity and specificity), the prevalence of the disease in the target population, and the cost and benefits of treating wrongly or rightly the subject. To calculate the utility function, the user needs to specify:

- the distribution of the marker in the subject with and without the disease (see the `fit` function)
- the prevalence of the disease in the target population
- the cost of treating subject without the disease and the benefit of treating a patient with the disease (see Details).

The optimal threshold and its credible interval are calculated using a Monte Carlo approach.

Usage

```
diagThresh(EvtDist = NULL, NoEvtDist = NULL, p, r, lowEvt = FALSE,
  le.MCMC = 1000, hessTol = 10^(-6), plot = FALSE,
  progress.bar = NULL, seed = NULL)
```

Arguments

`EvtDist`	an object of class allowedFitDist that summarizes the distribution fitted to the marker values of patients with the disease of interest. This class of objects is obtained using the `fit` function.
`NoEvtDist`	an object of class allowedFitDist that summarizes the distribution fitted to the marker values of patients without the disease of interest. This class of objects is obtained using the `fit` function.
`p`	the prevalence of the disease in the target population.
`r`	the risk threshold preference (see Details).

lowEvt	logical argument that specifies whether low values of the marker are associated with the presence of the disease or not.
le.MCMC	length of the desired MCMC chain.
hessTol	a numeric value used in the optimization algorithm to control the tolerance threshold of the hessian value at the optimal threshold estimate.
plot	a logical value indicating whether routine graphics must be produced.
progress.bar	a character string indicating whether the user wishes to print a progress bar during the function process.
seed	a numerical value used to fix the random seed.

Details

The r value can be defined as the probability of disease above which a patient or a physician would accept the treatment. The value (1-r)/r can be interpreted as the NB/NC ratio, i.e. the number of subjects without the disease a physician would accept to treat wrongly to be able to detect and treat one diseased patient.

Value

Returns an object of class diagOptThresh.

References

Subtil, F, and Rabilloud. A Bayesian method to estimate the optimal threshold of a longitudinal marker. *Biometrical Journal*. 2010; 52(3): 333-347.

Examples

```
#Simulating data from two gaussian distributions:
xE <- rnorm(100) # distribution of the marker in diseased patients
xEb <- rnorm(400, 2) # distribution of the marker in the subjects without the disease

#When working with real data. You can check the decision rule (whether low or high
#value of the marker are associated with the disease) using the densCurves function:
densCurves(x0 = xEb, x1 = xE, type = "diagnostic")

#Fit normal distributions on the two groups:
fitE <- fit(xE, "norm")
fitEb <- fit(xEb, "norm")

#Apply the main function to estimate the optimal threshold:

res <- diagThresh(fitE, fitEb, p = 0.2, r = 0.3, lowEvt = TRUE, le.MCMC = 5000,
                  plot = TRUE, progress.bar = "text")

#You can summarize the results using the summary() function:
summary(res,method = "median")

#You can extract the estimates and CI bounds of each indicator presented in the summary:
estimates(res, method = "median")
```

```
credibleIntervals(res)

#Plot the decision curves (this function is time-consuming):
dCres <- decisionCurve(res, r = seq(0, 0.5, length.out = 10))

#You can plot again the decision curves by applying the plot method to dCres,
#this function is a lot faster than the previous one. It also has more options
#to customize the plot:
plot(dCres)
```

`estimates` *Indicator estimates*

Description

This function calculates the punctual estimates of several indicators.

Usage

```
estimates(object, ...)

## S4 method for signature 'trtSelOptThresh'
estimates(object, method = "median")

## S4 method for signature 'diagOptThresh'
estimates(object, method = "median")
```

Arguments

`object`	a `trtSelOptThresh` S4 class object from which the decision curve must be plotted.
`...`	other arguments to pass to estimates method.
`method`	which method to use: median, mean or mode (median is the default).

Details

This function calculates the punctual estimates of several indicators (median, mean, or mode) depending on the type of marker under study (treatment selection or diagnostic marker).

Value

Returns a list of several indicator estimates.

Returns the estimates of several indicators.

`fit` *Specify which distribution to fit on the marker values*

Description

This function is a wrapper to create an S4 object to specify a distribution to fit the marker values.

Usage

```
fit(x, distr, ini = NULL, thin = NULL, burnin = NULL, model = NULL,
  paraNames = NULL, mcmcList = NULL, cdf = NULL, gradient = NULL,
  hessian = NULL)
```

Arguments

`x`	a vector of marker values (NA values allowed, see Details).
`distr`	a character that specifies the distribution to fit (normal, log-normal, scaled t, gamma, logistic, user-defined or undefined, see Details).
`ini`	specification of initial values for the parameters of the marker distribution in the form of a list. Each list must be named. A list should be provided for each MCMC chain. NULL for "norm" and "lnorm".
`thin`	the thinning interval between consecutive observations. NULL for "norm" and "lnorm".
`burnin`	a positive integer that defines the length of the burn-in iterations when performing the MCMC algorithm. NULL for "norm" and "lnorm".
`model`	a character string used to define the model. Must match with the definition of a model compatible with JAGS. Necessary only for the t and logistic distributions (see Details).
`paraNames`	a string vector containing the names of the parameters of the submitted distribution. Should be provided only for "user" defined distribution.
`mcmcList`	an object of class mcmc.list where each list contains an MCMC chain. To be provided only for "user" defined distribution.
`cdf`	a function that characterizes the cumulative distribution. To be provided only for "user" defined distribution (see Details).
`gradient`	a function that characterizes the density distribution. To be provided only for "user" defined distribution (see Details).
`hessian`	a function that characterizes the first derivative of the probability density function. To be provided only for "user" defined distribution (see Details).

Details

This function allows the user to specify which distribution should be fitted to the marker values. If NA values are present in the x argument passed to the function, a warning is produced. However, the user should not discard the NA values from the original data because the length of the x argument is calculated internally to to estimate the mean risk of event occurrence in each treatment arm. So NA values are managed internally by the function. Five theoretical distributions are implemented by the package: normal, log-normal, gamma, scaled t, and logistic. This is here that the user must specify which of the four distributions must be of type 'undefined' (or in other words which distribution must be expressed as a function of the three other distributions and mean risks of event). The user may also define its own theoretical distribution. The details for each theoretical distribution are provided hereafter:

- Fit a normal distribution: when specifying `distr="norm"` you fit a normal distribution to the marker values passed to the x argument of the function. Non-informative priors are used ($p(\mu, \sigma^2) \propto (\sigma^2)^(- 1)$). Posterior values of the normal distribution parameters are sampled directly from the exact posterior distributions. If you don't want to use non-informative priors, see the explanation on how to fit a user-defined distribution.
- Fit a log-normal distribution: when specifying `distr="lnorm"` you fit a log-normal distribution to the marker values passed to the x argument of the function. Non-informative priors are used ($p(\mu, \sigma^2) \propto (\sigma^2)^(- 1)$). Posterior values of the log-normal distribution parameters are sampled directly from the exact posterior distributions. If you don't want to use non-informative priors, see the explanation on how to fit a user-defined distribution.
- Fit a gamma distribution: when specifying `distr="gamma"` you fit a gamma distribution to the marker values passed to the x argument of the function. Non-informative priors are used ($p(shape, scale) \propto 1/scale$). Posterior values of the gamma distribution parameters are sampled using the ARS method. This method requires that the user specifies a list of initial values passed to the `ini` argument of the function. Each element of this list must be a list with one element named 'shape'. It also requires the `thin` of the MCMC chain, and the length of the burnin phase passed to the `burnin` argument. If you don't want to use non-informative priors, see the explanation on how to fit a user-defined distribution.
- Fit a scaled t distribution: when specifying `distr="t"` you fit a scaled t distribution to the marker values passed to the x argument of the function. Posterior values of the scaled t distribution parameters are sampled using an MCMC algorithm through the JAGS software, so the function requires the user to provide the JAGS model as a character string through the `model` argument of the function. If `NULL`, a model with vague priors is provided to the function automatically:

$$mu\ U(min(x), max(x))$$

$$log(sd)\ U(-10, 10)$$

$$1/df\ U(0, 1)$$

  This method requires that the user specifies a list of initial values passed to the `ini` argument of the function. Each element of this list must be a list with three elements named 'mu', 'sd', and 'df'. It also requires the `thin` of the MCMC chain, and the length of the burnin phase passed to the `burnin` argument.
- Fit a logistic distribution: when specifying `distr="logis"` you fit a logistic distribution to the marker values passed to the x argument of the function. Posterior values of the logistic distribution parameters are sampled using a MCMC algorithm through the JAGS software,

so the function requires the user to provide the JAGS model as a character string through the `model` argument of the function. If `NULL`, a model with vague priors is provided to the function automatically:

$$location\ U(min(x), max(x))$$

$$log(scale)\ U(-10, 10)$$

This method requires that the user specifies a list of initial values passed to the `ini` argument of the function. Each element of this list must be a list with two elements named 'location', and 'scale'. It also requires the `thin` of the MCMC chain, and the length of the burnin phase passed to the `burnin` argument.

- Fit a user-defined distribution: when specifying `distr="user"` you fit a user-defined distribution to the marker values passed to the `x` argument of the function. First of all, the user must give the parameters name in the argument `paraNames` of the function using a character vector. Then, the user provides a posterior sample of the parameters of the distribution obtained using JAGS or another software through an object of class `mcmc.list` to the argument `mcmcList` of the function (this implies that the user performed the Bayesian inference himself). Note that the names passed to the `mcmc.list` object must match with the names given in the `paraNames` argument. Then, the user must specify the `cdf`, `gradient`, and `hessian` functions associated with the fitted distribution. The `cdf` function is the cumulative distribution function that is fitted to the marker values, the `gradient` function is its first derivative which corresponds to the probability density function fitted to the marker values, and the `hessian` function is the second derivative of `cdf`. When the fitted distribution is a supported distribution (e.g. a normal distribution with informative priors), the user may use the `getMethod(cdf,"normalDist")` function to use the standard method for normal distribution used in the package. When the fitted distribution is not supported, the user must specify directly the `cdf` as `function(x,mu,sd) pnorm(x,mu,sd)` (if we keep the example of the normal distribution). The same idea may be used for the `gradient` and `hessian` functions (see the examples to have more details).
- Specify which marker distribution is expressed as a function of the three others and the mean risks of event using `distr="undefined"`.

Value

Returns an object to be passed to the `trtSelThresh` and `diagThresh` functions.

See Also

`trtSelThresh` and `diagThresh`.

Examples

```
#Fit a normal distribution
x <- rnorm(250)
fitX <- fit(x, "norm")

#Fit a log-normal distribution
x <- rlnorm(250)
fitX <- fit(x, "lnorm")

#Fit a gamma distribution
```

```
x <- rgamma(250, shape = 2, scale = 1.2)
fitX <- fit(x, "gamma",
            ini = list(list(shape = 1),
                       list(shape = 2),
                       list(shape = 3)),
            thin = 1, burnin = 1000)

#Fit a scaled t distribution
x <- optimalThreshold:::rt.scaled(250, df = 4, mean = 2.5, sd = 2)
fitX <- fit(x, "t",
            ini = list(list(mu = 1, sd = 1, df = 2),
                       list(mu = 2, sd = 2, df = 4),
                       list(mu = 3, sd = 3, df = 6)),
            thin = 1, burnin = 1000, model = NULL)

#Fit a logistic distribution
x <- rlogis(250)
fitX <- fit(x, "logis",
            ini = list(list(location = 0.3, scale = 0.5),
                       list(location = 1, scale = 1),
                       list(location = 2, scale = 2)),
            thin = 1, burnin = 1000, model = NULL)

#Specify which distribution is 'undefined'
x <- rnorm(250)
fitX <- fit(x, "undefined")

#Fit a user-defined normal distribution with informative priors
library(rjags)
x <- rnorm(250, mean = 2, sd = 1)
model <- "model
  {
   mu ~ dunif(0, 4)
   log_sd ~ dunif(-1, 1)
   sd <- exp(log_sd)
   tau <- 1 / (sd^2)
   for (i in 1:N)
   {
    x[i] ~ dnorm(mu, tau)
   }
  }
  "
modelJAGS <- jags.model(file = textConnection(model), data = list(x = x, N = length(x)),
                   inits = list(list(mu = 1, log_sd = -0.5),list(mu = 3.5, log_sd = 0.5)),
                        n.chains = 2, quiet = TRUE)
update(modelJAGS, 1000, progress.bar = "text")
mcmcpara <- coda.samples(modelJAGS, c("mu", "log_sd"), n.iter = 2000, thin = 1)
varnames(mcmcpara) <- c("mu", "sd")
mcmcpara[[1]][, "sd"] <- exp(mcmcpara[[1]][, "sd"])
mcmcpara[[2]][, "sd"] <- exp(mcmcpara[[2]][, "sd"])
fitX <- fit(x, "user", paraNames = varnames(mcmcpara), mcmcList = mcmcpara,
            cdf = function(x, mu, sd) pnorm(x, mu, sd),
            gradient = getMethod(gradient, "normalDist"),
```

```
hessian = function(x, mu, sd) ((mu - x) / sd^2) * dnorm(x, mu, sd))
```

`fitGammaDist-class` *An S4 class to fit a gamma distribution on a vector of marker values.*

Description

This class allows to fit a gamma distribution on the marker values `x`.

Details

This class is automatically created when the user applies the `fit` function with the argument `distr="gamma"`. You never have to create manually this class, it is created internally.

Slots

`x` This slot takes in argument the marker values. Numeric argument.

`n` Length of x vector (including NA values). Numeric argument.

`ini` This slot is a list of initial values passed to the MCMC algorithm. List argument.

`thin` This slot is a strictly positive integer value that specifies the 'thin' applied to the MCMC algorithm.

`burnin` This slot is a positive integer value that specifies the length of the burnin period in the MCMC algorithm.

`mcmc` This slot allows the main function to k,now whether an MCMC algorithm must be performed to sample the distribution parameters from their posterior distribution.

See Also

`fit` for more details on how to fit a gamma distribution.

`fitLogisticDist-class` *An S4 class to fit a logistic distribution on a vector of marker values.*

Description

This class allows to fit a logistic distribution on the marker values `x`.

Details

This class is automatically created when the user applies the `fit` function with the argument `distr="logis"`. You never have to create manually this class, it is created internally.

Slots

`x` This slot takes in argument the marker values. Numeric argument.

`n` Length of x vector (including NA values). Numeric argument.

`ini` This slot is a list of initial values passed to the MCMC algorithm. List argument.

`thin` This slot is a strictly positive integer value that specifies the 'thin' applied to the MCMC algorithm.

`burnin` This slot is a positive integer value that specifies the length of the burnin period in the MCMC algorithm.

`model` This slot is a character string that specifies the model passed to the JAGS software to perform the MCMC algorithm.

`mcmc` This slot allows the main function to k,now whether an MCMC algorithm must be performed to sample the distribution parameters from their posterior distribution.

See Also

`fit` for more details on how to fit a logistic distribution.

`fitLogNormalDist-class` *An S4 class to fit a log-normal distribution on a vector of marker values.*

Description

This class allows to fit a log-normal distribution on the marker values `x`.

Details

This class is automatically created when the user applies the `fit` function with the argument `distr="lnorm"`. You never have to create manually this class, it is created internally.

Slots

`x` This slot takes in argument the marker values. Numeric argument.

`n` Length of x vector (including NA values). Numeric argument.

`mcmc` This slot allows the main function to k,now whether an MCMC algorithm must be performed to sample the distribution parameters from their posterior distribution.

See Also

`fit` for more details on how to fit a log-normal distribution.

fitNormalDist-class *An S4 class to fit a normal distribution on a vector of marker values.*

Description

This class allows to fit a normal distribution on the marker values `x`.

Details

This class is automatically created when the user applies the `fit` function with the argument `distr="norm"`. You never have to create manually this class, it is created internally.

Slots

`x` This slot takes in argument the marker values. Numeric argument.

`n` Length of x vector (including NA values). Numeric argument.

`mcmc` This slot allows the main function to k,now whether an MCMC algorithm must be performed to sample the distribution parameters from their posterior distribution.

See Also

`fit` for more details on how to fit a normal distribution.

fitStudentDist-class *An S4 class to fit a Student distribution on a vector of marker values.*

Description

This class allows to fit a t distribution on the marker values `x`.

Details

This class is automatically created when the user applies the `fit` function with the argument `distr="t"`. You never have to create manually this class, it is created internally.

Slots

`x` This slot takes in argument the marker values. Numeric argument.

`n` Length of x vector (including NA values). Numeric argument.

`ini` This slot is a list of initial values passed to the MCMC algorithm. List argument.

`thin` This slot is a strictly positive integer value that specifies the 'thin' applied to the MCMC algorithm.

`burnin` This slot is a positive integer value that specifies the length of the burnin period in the MCMC algorithm.

`model` This slot is a character string that specifies the model passed to the JAGS software to perform the MCMC algorithm.

`mcmc` This slot allows the main function to know whether an MCMC algorithm must be performed to sample the distribution parameters from their posterior distribution.

See Also

`fit` for more details on how to fit a t distribution.

`fitUserDefinedDist-class`

An S4 class to fit a user-defined distribution on a vector of marker values.

Description

This class allows to fit a user-defined distribution on the marker values `x`.

Details

This class is automatically created when the user applies the `fit` function with the argument `distr="user"`. You never have to create manually this class, it is created internally.

Slots

`x` This slot takes in argument the marker values. Numeric argument.

`n` Length of x vector (including NA values). Numeric argument.

`paraNames` This slot is a character vector of distribution parameter names.

`mcmcList` This slot is an mcmc.list object summing up all the sampled parameters of the user-defined distribution.

`cdf` This slot is a function that describes the cumulative distribution function of the user-defined distribution.

`gradient` This slot is a function that describes the probability density function of the user-defined distribution.

`hessian` This slot is a function that describes the fisrt derivative of the probability density function of the user-defined distribution.

`mcmc` This slot allows the main function to k,now whether an MCMC algorithm must be performed to sample the distribution parameters from their posterior distribution.

See Also

`fit` for more details on how to fit a user-defined distribution.

gammaDist-class *An S4 class to represent a gamma distribution.*

Description

This S4 class describes the gamma distribution that is fitted to the marker values. The gamma distribution is characterized by the `shape` and the `scale` parameters.

Details

You never have to create this class manually. This class is created internally when a gamma distribution is fitted on the marker values.

Slots

`shape` shape parameter. Must be positive.

`scale` scale parameter. Must be strictly positive.

See Also

`fit` for more details on how to fit a gamma distribution.

gradient *Probability density function of a specified distribution*

Description

The `gradient` function returns the probability density function relative to the S4 object passed in its argument. See details to know on what kind of S4 objects this function could be applied.

Usage

```
gradient(object)

## S4 method for signature 'normalDist'
gradient(object)

## S4 method for signature 'logNormalDist'
gradient(object)

## S4 method for signature 'gammaDist'
gradient(object)

## S4 method for signature 'studentDist'
gradient(object)
```

```
## S4 method for signature 'logisticDist'
gradient(object)

## S4 method for signature 'compoundEvtRefDist'
gradient(object)

## S4 method for signature 'compoundNoEvtRefDist'
gradient(object)

## S4 method for signature 'compoundEvtInnovDist'
gradient(object)

## S4 method for signature 'compoundNoEvtInnovDist'
gradient(object)
```

Arguments

object	Any S4 object for which a gradient method is defined. Should match with the definition of an S4 distribution object as defined in the optimalThreshold package.

Details

This method can be applied to the S4 distribution objects that are supported in the optimalThreshold package: normalDist, logNormalDist, gammaDist, studentDist, logisticDist, and userDefinedDist. These methods are applied internally, and you have no need to use it outside of the main functions trtSelThresh and diagThresh.

- Normal distribution: the gradient method applied to a normalDist object is simply the dnorm function (see help on this function to have more details).
- Log-normal distribution: the gradient method applied to a logNormalDist object is simply the dlnorm function (see help on this function to have more details).
- Gamma distribution: the gradient method applied to a gammaDist object is simply the dgamma function (see help on this function to have more details).
- Scaled t distribution: the scaled t distribution with df = n, mu = μ, and sd = σ has density:

$$f(x) = (\Gamma((n+1)/2)/(\sqrt{n\pi}\Gamma(n/2))(1 + ((x-\mu)/\sigma)^2/n)^{-}((n+1)/2))/\sigma$$

- Logistic distribution: the gradient method applied to a logisticDist object is simply the dlogis function (see help on this function to have more details).
- User-defined distribution: the gradient method applied to a userDefinedDist object is simply the gradient function provided by the user when fitting a user-defined distribution with the fit function.

The S4 objects compoundEvtRefDist, compoundNoEvtRefDist, compoundEvtInnovDist, and compoundNoEvtInnovDist are created internally. The gradient function applied to these objects is defined dynamically depending on what types of distribution are fitted. The definition of the gradient function relies on the expression of the randomization constraint of a clinical trial that enforces the distribution of the marker in each treatment arm to be identical (see References for more details).

Value

Returns the probability density function of the specified distribution.

References

Blangero, Y, Rabilloud, M, Ecochard, R, and Subtil, F. A Bayesian method to estimate the optimal threshold of a marker used to select patients' treatment. *Statistical Methods in Medical Research*. 2019.

See Also

`trtSelThresh`, `dnorm`, `dlnorm`, `dgamma`, `dlogis`, `fit`

`hessian` *Second derivative of the cumulative distribution function of a specified distribution*

Description

The `hessian` function returns the second derivative of the cumulative distribution function relative to the S4 object passed in its argument. See details to know on what kind of S4 objects this function could be applied.

Usage

```
hessian(object)

## S4 method for signature 'normalDist'
hessian(object)

## S4 method for signature 'logNormalDist'
hessian(object)

## S4 method for signature 'gammaDist'
hessian(object)

## S4 method for signature 'studentDist'
hessian(object)

## S4 method for signature 'logisticDist'
hessian(object)

## S4 method for signature 'compoundEvtRefDist'
hessian(object)

## S4 method for signature 'compoundNoEvtRefDist'
hessian(object)
```

```
## S4 method for signature 'compoundEvtInnovDist'
hessian(object)

## S4 method for signature 'compoundNoEvtInnovDist'
hessian(object)
```

Arguments

object	A distribution object.

Details

This method can be applied to the S4 distribution objects that are supported in the optimalThreshold package: normalDist, logNormalDist, gammaDist, studentDist, logisticDist, and userDefinedDist. These methods are applied internally, and you have no need to use it outside of the main functions trtSelThresh and diagThresh.

- Normal distribution: the hessian method applied to a normalDist object is simply the second derivative of the cumulative distribution function of a normal distribution, with mu=μ and sd=σ, and expressed as:
$$f'(x) = ((\mu - x)/\sigma^2) * f(x)$$
- Log-normal distribution: the hessian method applied to a logNormalDist object is simply the second derivative of the cumulative distribution function of a log-normal distribution, with mu=μ and sd=σ, and expressed as:
$$f'(x) = (((\mu - \log(x))/(x * \sigma^2)) - 1/x) * f(x)$$
- Gamma distribution: the hessian method applied to a gammaDist object is simply the second derivative of the cumulative distribution function of a gamma distribution, with shape=α and scale=β, and expressed as:
$$f'(x) = ((\alpha - 1)/x - 1/\beta) * f(x)$$
- Scaled t distribution: the hessian method applied to a studentDist object is simply the second derivative of the cumulative distribution function of a t scaled distribution, with df=n, mu=μ and sd=σ, and expressed as:
$$f'(x) = (-(n+1)) * ((x-\mu)/(\sigma^2 * (n + ((x-\mu)/\sigma)^2))) * f(x)$$
- Logistic distribution: the hessian method applied to a logisticDist object is simply the second derivative of the cumulative distribution function of a logistic distribution, with location=μ, and scale=σ, and expressed as:
$$f'(x) = ((\exp(-(x-\mu)/\sigma)^2 - 1)/(\sigma * (1 + \exp(-(x-\mu)/\sigma))^2)) * f(x)$$
- User-defined distribution: the hessin method applied to a userDefinedDist object is simply the hessian function provided by the user when fitting a user-defined distribution with the fit function.

The S4 objects `compoundEvtRefDist`, `compoundNoEvtRefDist`, `compoundEvtInnovDist`, and `compoundNoEvtInnovDist` are created internally. The `hessian` function applied to these objects is defined dynamically depending on what types of distribution are fitted. The definition of the `hessian` function relies on the expression of the randomization constraint of a clinical trial that enforces the distribution of the marker in each treatment arm to be identical (see References for more details).

Value

Returns the second derivative of the cumulative distribution function of the specified distribution.

References

Blangero, Y, Rabilloud, M, Ecochard, R, and Subtil, F. A Bayesian method to estimate the optimal threshold of a marker used to select patients' treatment. *Statistical Methods in Medical Research*. 2019.

See Also

`trtSelThresh`, `fit`

`logisticDist-class` *An S4 class to represent a logistic distribution.*

Description

This S4 class describes the logistic distribution that is fitted to the marker values. The logistic distribution is characterized by the `location` and the `scale` parameters.

Details

You never have to create this class manually. This class is created internally when a logistic distribution is fitted on the marker values.

Slots

`location` location parameter.

`scale` scale parameter. Must be strictly positive.

See Also

`fit` for more details on how to fit a logistic distribution.

logNormalDist-class *An S4 class to represent a log-normal distribution.*

Description

This S4 class describes the log-normal distribution that is fitted to the marker values. The log-normal distribution is characterized by the `mu` and the `sd` parameters.

Details

You never have to create this class manually. This class is created internally when a log-normal distribution is fitted on the marker values.

Slots

`mu` mu parameter.

`sd` standard deviation parameter. Must be strictly positive.

See Also

`fit` for more details on how to fit a log-normal distribution.

mcmc.listOrNull-class *An S4 union class to merge mcmc.list and NULL types of object.*

Description

An S4 union class to merge mcmc.list and NULL types of object.

normalDist-class *An S4 class to represent a normal distribution.*

Description

This S4 class describes the normal distribution that is fitted to the marker values. The normal distribution is characterized by the `mu` and the `sd` parameters.

Details

You never have to create this class manually. This class is created internally when a normal distribution is fitted on the marker values.

Slots

`mu` mu parameter.

`sd` standard deviation parameter. Must be strictly positive.

See Also

`fit` for more details on how to fit a normal distribution.

`plot-methods` *Plot method*

Description

Plot method

Usage

```
## S4 method for signature 'trtSelOptThresh'
plot(x, y,
  main = "MCMC sample distribution of optimal threshold",
  col = "gray85", border.col = "darkgrey",
  xlab = "Optimal threshold estimate", yaxs = "i", freq = FALSE,
  breaks = seq(min(x@optThresh, na.rm = TRUE), max(x@optThresh, na.rm =
  TRUE), length.out = 20), ...)

## S4 method for signature 'diagOptThresh'
plot(x, y,
  main = "MCMC sample distribution of optimal threshold",
  col = "gray85", border.col = "darkgrey",
  xlab = "Optimal threshold estimate", yaxs = "i", freq = FALSE,
  breaks = seq(min(x@optThresh, na.rm = TRUE), max(x@optThresh, na.rm =
  TRUE), length.out = 20), ...)
```

Arguments

`x`	a `trtSelOptThresh` or a `diagOptThresh` object.
`y`	unused parameter.
`main`	an overall title for the plot.
`col`	the color of the histogram.
`border.col`	the color of the histogram border.
`xlab`	a label for the x axis of the plot.
`yaxs`	The style of axis interval calculation to be used for the y-axis.
`freq`	logical; if TRUE, the histogram graphic is a representation of frequencies; if FALSE, probability densities are plotted (so that the histogram has a total area of one).

breaks	one of: • a vector giving the breakpoints between histogram cells, • a function to compute the vector of breakpoints, • a single number giving the number of cells for the histogram, • a character string naming an algorithm to compute the number of cells, • a function to compute the number of cells. In the last three cases the number is a suggestion only; as the breakpoints will be set to pretty values, the number is limited to 1e6 (with a warning if it was larger). If breaks is a function, the x vector is supplied to it as the only argument (and the number of breaks is only limited by the amount of available memory).
...	other graphical parameters.

Value

None

`plot.diagRelUtility` *Plot the decision curves of a diagnostic marker*

Description

Plot the decision curves of a diagnostic marker

Usage

```
## S4 method for signature 'diagRelUtility'
plot(x, y, main = "Decision curves",
  lty = 1, lwd = 1, xlim = range(x@r), ylim = c(min(x@U,
  x@UNoTreat, x@UTreatAll), max(x@U, x@UNoTreat, x@UTreatAll)),
  ylab = "Expected benefit", xlab = "r", col.U = "black",
  col.UNoTreat = "blue", col.UTreatAll = "green", ...)
```

Arguments

x	a `diagRelUtility` object.
y	unused parameter.
main	an overall title for the plot.
lty	the line type. Line types can either be specified as an integer (0=blank, 1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash) or as one of the character strings "blank", "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash", where "blank" uses 'invisible lines' (i.e., does not draw them).
lwd	the line width, a *positive* number, defaulting to 1. The interpretation is device-specific, and some devices do not implement line widths less than one. (See the help on the device for details of the interpretation).

xlim	the x limits of the plot.
ylim	the x limits of the plot.
ylab	a label for the y axis.
xlab	a label for the x axis
col.U	color of the utility curve for the marker-based strategy.
col.UNoTreat	color of the utility curve for the "No Treat" strategy.
col.UTreatAll	color of the utility curve for the "Treat All" strategy.
...	other graphical parameters.

Value

None

`plot.trtSelRelUtility` *Plot the decision curves of a treatment selection marker*

Description

Plot the decision curves of a treatment selection marker

Usage

```
## S4 method for signature 'trtSelRelUtility'
plot(x, y, which = c(1, 2), alpha = 0.05,
  conf.int = TRUE, main1 = "Decision curves (unscaled)",
  main2 = "Decision curve (scaled)", lty = 1, lwd = 1,
  xlim = range(x@r), ylim1 = c(min(x@U, x@UT0, x@UT1, x@Up), max(x@U,
  x@UT0, x@UT1, x@Up)), ylim2 = c(0, 1), ylab1 = "Utility",
  ylab2 = "Relative utility", xlab = "r ratio", col.U = "black",
  col.Up = "red", col.UT0 = "blue", col.UT1 = "green",
  col.RU = "black", col.conf.int = "black", add = FALSE,
  legend1 = TRUE, ...)
```

Arguments

x	a `trtSelRelUtility` object.
y	unused parameter.
which	indicates which graph should be plotted. Default is both graphs.
alpha	alpha risk for the confidence intervals.
conf.int	a logical value indicating whether the confidence intervals should be plotted for the relative utility curve.
main1	an overall title for the first plot.
main2	an overall title for the second plot.

lty	the line type. Line types can either be specified as an integer (0=blank, 1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash) or as one of the character strings "blank", "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash", where "blank" uses 'invisible lines' (i.e., does not draw them).
lwd	the line width, a *positive* number, defaulting to 1. The interpretation is device-specific, and some devices do not implement line widths less than one. (See the help on the device for details of the interpretation).
xlim	the x limits of the plots.
ylim1	the y limits of the first plot.
ylim2	the y limits of the second plot.
ylab1	a label for the y axis of the first plot.
ylab2	a label for the y axis of the second plot.
xlab	a label for the x axis of the plots.
col.U	color of the utility curve for the marker-based strategy.
col.Up	color of the utility curve for the perfect marker-based strategy.
col.UT0	color of the utility curve for the "Treat All with the reference treatment" strategy.
col.UT1	color of the utility curve for the "Treat All with the innovative treatment" strategy.
col.RU	color of the relative utility curve.
col.conf.int	color of the confidence intervals.
add	a logical value indicating whether the relative utility curve should superimpose with an existing graph. Only works when `which` = 2.
legend1	a logical value indicating whether a legend should be added to the first plot.
...	other graphical parameters.

Value

None

`riskCurves` *Marker-by-treatment predictiveness curves plot*

Description

This function plots the marker-by-treatment predictiveness curves for treatment selection markers, corresponding to the risk of event in each treatment arm in function of the cumulative distribution of the marker.

Usage

```
riskCurves(x0E, x0Eb, x1E, x1Eb, ylab = "Predicted risk of event",
  xlab = "Empirical cumulative distribution function of the marker",
  main = "Marker-by-treatment predictiveness curves")
```

Arguments

x0E	a numeric vector of the marker values for patients in the reference arm that developed the event.
x0Eb	a numeric vector of the marker values for patients in the reference arm that did not develop the event.
x1E	a numeric vector of the marker values for patients in the innovative arm that developed the event.
x1Eb	a numeric vector of the marker values for patients in the innovative arm that did not develop the event.
ylab	label of the Y-axis.
xlab	label of the X-axis.
main	title of the graph.

Details

The function uses regression splines to plot the marker-by-treatment predictiveness curves. This graph may be used to check graphically the strength of the marker-by-treatment interaction, and to know whether low values of the marker are associated with a better response of the reference treatment (this information is needed in the `trtSelThresh` function).

Value

None

References

Janes, H, Pepe, MS, Bossuyt, PM, and Barlow, WE. Measuring the performance of markers for guiding treatment decisions. *Annals of Internal Medicine*. 2011; 154(4): 253-259.

See Also

`gam` for more details about regression splines.

Examples

```
x0E <- rnorm(100, 2, 1)
x0Eb <- rnorm(100, 4, 1)
x1E <- rnorm(100, 4, 1)
x1Eb <- rnorm(100, 2, 1)
riskCurves(x0E, x0Eb, x1E, x1Eb)
```

samplePosteriorDist *Sample in the posterior distribution of the parameters of a given theoretical distribution.*

Description

The samplePosteriorDist function samples the parameters of a given theoretical distribution using explicit posterior distribution (if it exists), or a Markov Chain Monte Carlo (MCMC) algorithm when the posterior distribution is unknown. See details to know on what kind of S4 objects this function could be applied.

Usage

```
samplePosteriorDist(object, K, ...)

## S4 method for signature 'fitNormalDist'
samplePosteriorDist(object, K, n)

## S4 method for signature 'fitLogNormalDist'
samplePosteriorDist(object, K, n)

## S4 method for signature 'fitGammaDist'
samplePosteriorDist(object, K, do.pb, seed)

## S4 method for signature 'fitStudentDist'
samplePosteriorDist(object, K, do.pb, seed)

## S4 method for signature 'fitLogisticDist'
samplePosteriorDist(object, K, do.pb, seed)
```

Arguments

object	A distribution object.
K	A numerical value indicating the length of the sample.
...	other parameters passed to methods.
n	number of MCMC chains.
do.pb	Indicates whther progressing bar or not
seed	seed for the random number generator. Integer.

Details

This method can be applied to the S4 distribution objects that are supported in the optimalThreshold package: fitNormalDist, fitLogNormalDist, fitGammaDist, fitStudentDist, and fitLogisticDist. These methods are applied internally, and you have no need to use it outside of the main function optThresEst. See below to have details on the expression of the samplePosteriorDist function according to the type of distribution.

- Normal distribution: a noninformative prior is used for the parameters of the normal distribution (`mu`=μ, and `sd`=σ). The σ^2 parameter is sampled from an inverse Chi-squared distribution, and the μ parameter is sampled from a normal distribution with known variance. So, sampling in the posterior distribution of μ and σ does not involve an MCMC algorithm (see References for more details and justification).
- Log-normal distribution: a noninformative prior is used for the parameters of the log-normal distribution (`mu`=μ, and `sd`=σ). The σ^2 parameter is sampled from an inverse Chi-squared distribution, and the μ parameter is sampled from a normal distribution with known variance. So, sampling in the posterior distribution of μ and σ does not involve an MCMC algorithm (see References for more details and justification).
- Gamma distribution: a noninformative prior is used for the parameters of the gamma distribution (`shape`=α, and `scale`=β). The parameters are sampled using an adaptive rejection sampling (ARS) algorithm. The β parameter is sampled at the first iteration from an inverse gamma distribution using the initial value of the α parameter provided by the user. Then the ARS algorithm is performed to sample α from its posterior distribution (see References for more details and justification).
- Scaled t distribution: a vague prior is used for the parameters of the scaled t distribution as a default. However, the user can write its own JAGS model to use different priors (see the `fit` function for more details). Sampling from the posterior distribution of the parameters of a scaled t distribution requires JAGS to be installed.
- Logistic distribution: a vague prior is used for the parameters of the logistic distribution as a default. However, the user can write its own JAGS model to use different priors (see the `fit` function for more details). Sampling from the posterior distribution of the parameters of a logistic distribution requires JAGS to be installed.

Value

Returns an object of class list.

References

Gelman, A, et al. 2014. *Bayesian Data Analysis*. 3rd edition, CRC Press, Boca Raton, section 2.8.
Sook, Y, and Oh, M. Bayesian estimation of the two-parameter Gamma distribution. *Communications in Statistics - Simulation and Computation*. 2006; 35: 285-293.

See Also

`trtSelThresh`

---

show-methods *Show method*

Description

Show some of the slots of a `trtSelOptThresh` or a `diagOptThresh` objects.

Usage

```
## S4 method for signature 'trtSelOptThresh'
show(object)

## S4 method for signature 'diagOptThresh'
show(object)
```

Arguments

object	a trtSelOptThresh or a diagOptThresh S4 object.

Value

None

---

studentDist-class *An S4 class to represent a scaled Student distribution.*

Description

This S4 class describes the scaled t distribution that is fitted to the marker values. The scaled t distribution is characterized by the df (degrees of freedom), the mu, and the sd parameters.

Details

You never have to create this class manually. This class is created internally when a scaled t distribution is fitted on the marker values.

Slots

df degrees of freedom (> 0, maybe non-integer).

mu mu parameter.

sd standard deviation parameter. Must be strictly positive.

See Also

fit for more details on how to fit a t distribution.

summary-methods *An S4 method that summarizes the results of a* trtSelOpthThresh *or a* diagOpthThresh *object.*

Description

An S4 method that summarizes the results of a trtSelOpthThresh or a diagOpthThresh object.

Usage

```
## S4 method for signature 'trtSelOptThresh'
summary(object, alpha = 0.05,
  method = "median")

## S4 method for signature 'diagOptThresh'
summary(object, alpha = 0.05,
  method = "median")
```

Arguments

object	a trtSelOptThresh S4 class object for which a summary is desired.
alpha	alpha parameter for the confidence level required.
method	which method to use: median, mean or mode (median is the default).

Details

This function presents the results stocked in a trtSelOpthThresh object, or in a diagOpthThresh object. For a trtSelOpthThresh object it prints:

- The decision rule: is the reference treatment recommended for low values of the marker?
- The median (default), mean, or mode risk of event occurrence in each treatment arm, and their credible interval.
- Some summary statistics of the marker under study (min, max, quartiles and mean)
- The optimal threshold estimate and its credible interval (percentile and highest posterior density).
- The median (default), mean, or mode risk in each arm under the marker-based strategy.
- The median (default), mean, or mode benefit estimate under each treatment arm.
- The percentage of NA values returned during the optimal threshold estimation process.

For a diagOpthThresh object, it prints:

- The decision rule: is the reference treatment recommended for low values of the marker?
- The median (default), mean, or mode risk of event occurrence in each treatment arm, and their credible interval.
- Some summary statistics of the marker under study (min, max, quartiles and mean)

- The optimal threshold estimate and its credible interval (percentile and highest posterior density).
- The median (default), mean, or mode risk in each arm under the marker-based strategy.
- The median (default), mean, or mode benefit estimate under each treatment arm.
- The percentage of NA values returned during the optimal threshold estimation process.

Value

This function returns an object of class 'summaryTrtSelOptThresh'.

This function returns an object of class 'summaryDiagOptThresh'.

References

Blangero, Y, Rabilloud, M, Ecochard, R, and Subtil, F. A Bayesian method to estimate the optimal threshold of a marker used to select patients' treatment. *Statistical Methods in Medical Research*. 2019.

Subtil, F, and Rabilloud. A Bayesian method to estimate the optimal threshold of a longitudinal biomarker. *Biometrical Journal*. 2010.

See Also

`trtSelThresh` for more details on how to estimate the optimal threshold of a treatment selection marker.

`diagThresh` for more details on how to estimate the optimal threshold of a diagnostic marker.

---

trtSelOptThresh-class *An S4 class to describe the optimal threshold of a treatment selection marker.*

Description

An S4 class to describe the optimal threshold of a treatment selection marker.

Details

You never have to create this class manually. This class is created internally when the `trtSelThresh` function is used.

Slots

`optThresh` This slot is an object that takes in argument the sampled optimal threshold values. Numeric argument.

`r0` This slot is an object that takes in argument the sampled mean risks of event occurrence in the reference arm. Numeric argument.

`r1` This slot is an object that takes in argument the sampled mean risks of event occurrence in the innovative arm. Numeric argument.

`xEvtRef` This slot is an object that takes in argument the marker values in the subgroup of patients that developed the event in the reference arm. Numeric argument.

`xNoEvtRef` This slot is an object that takes in argument the marker values in the subgroup of patients that did not develop the event in the reference arm. Numeric argument.

`xEvtInnov` This slot is an object that takes in argument the marker values in the subgroup of patients that developed the event in the innovative arm. Numeric argument.

`xNoEvtInnov` This slot is an object that takes in argument the marker values in the subgroup of patients that did not develop the event in the innovative arm. Numeric argument.

`lowRef` This slot is a logical argument that specifies whether the reference treatment is recommended for low values of the marker.

`toxRef` This slot is a logical argument that specifies whether the reference treatment is the most toxic treatment option at equal efficacy with the innovative treatment.

`markerBasedRiskRef` This slot is an object that takes in argument the sampled mean risks of event occurrence in the reference treatment under the marker-based allocation rule. Numeric argument.

`markerBasedRiskInnov` This slot is an object that takes in argument the sampled mean risks of event occurrence in the innovative treatment under the marker-based allocation rule. Numeric argument.

`mcmcChainEvtRef` This slot is an object that takes in argument the sampled distribution objects in the subgroup of patients that developed the event in the reference arm. list argument.

`mcmcChainNoEvtRef` This slot is an object that takes in argument the sampled distribution objects in the subgroup of patients that did not develop the event in the reference arm. list argument.

`mcmcChainEvtInnov` This slot is an object that takes in argument the sampled distribution objects in the subgroup of patients that developed the event in the innovative arm. list argument.

`mcmcChainNoEvtInnov` This slot is an object that takes in argument the sampled distribution objects in the subgroup of patients that did not develop the event in the innovative arm. list argument.

`tabMCMCChain` This slot is an object that takes in argument all the distribution parameters that were sampled using the MCMC algorithm. mcmc.listOrNull argument.

`paraNamesUserDefined` This slot is an object that takes in argument the list of the distribution parameter names defined by the user in a 'fitUserDefinedDist' object. list argument.

`cdfUserDefined` This slot is an object that takes in argument the list of cumulative distribution functions defined by the user in 'fitUserDefinedDist' objects. list argument.

`gradientUserDefined` This slot is an object that takes in argument the list of gradient functions defined by the user in 'fitUserDefinedDist' objects. list argument.

`hessianUserDefined` This slot is an object that takes in argument the list of hessian functions defined by the user in 'fitUserDefinedDist' objects. list argument.

`percentNA` This slot is a numeric object that indicates the percentage of NA values contained in the 'optThresh' slot.

See Also

`trtSelThresh` for more details on how to estimate the optimal threshold of a treatment selection marker.

trtSelRelUtility-class

An S4 class to the results from the decisionCurve methods.

Description

An S4 class to the results from the decisionCurve methods.

Details

You never have to create this class manually. This class is created internally when the `decisionCurve` method is applied to an 'optThresh' object.

Slots

`RU` This slot is a matrix of the relative utility according to the r ratios.

`U` This slot is a matrix of the marker-based utility according to the r ratios.

`UT0` This slot is a matrix of the reference treatment utility according to the r ratios.

`UT1` This slot is a matrix of the innovative treatment utility according to the r ratios.

`Up` This slot is a matrix of perfect marker utility according to the r ratios.

`r` Ratio of treatment/event costs. Numeric argument.

See Also

`decisionCurve` for more details on how to plot the decision curves.

trtSelThresh *Estimation of the optimal threshold of a treatment selection marker*

Description

This function produces a sample of the posterior distribution of the optimal threshold of a treatment selection marker. The optimal threshold is defined as the marker value that maximized the utility of using the marker to decide between two treatment options (innovative and reference one). The utility function takes into account the efficacy of the treatment options as well as treatment induced toxicities. The estimation of the utility function needs data from a clinical trial about the two treatment options, in which the success of a treatment is defined by the absence of an event of interest in a given post-treatment interval (binary data). For the time being, the package cannot estimate the optimal threshold in case of censored data about the occurrence of the event in the given post-treatment interval. To calculate the utility function, the user needs to specify:

- the distribution of the marker in the four groups defined by the treatment option and the outcome; in fact only three distributions need to be specified, the fourth one being derived from the three others and the mean risks of event in the two treatment arms through the randomization constraint (the distribution of the marker being the same in both treatment arms; see the `fit` function for more details),

- and the mean risks of the event in the two treatment arms. The user must also specify: the cost of the innovative treatment relative to the cost of the event (see Details).

The optimal threshold and its credible interval are calculated using a Monte Carlo approach.

Usage

```
trtSelThresh(EvtRefDist = NULL, NoEvtRefDist = NULL,
  EvtInnovDist = NULL, NoEvtInnovDist = NULL, mRiskRef = NULL,
  mRiskInnov = NULL, lowRef = TRUE, toxRef = TRUE, r = 0,
  le.MCMC = 1000, hessTol = 10^(-6), plot = FALSE,
  progress.bar = NULL, seed = NULL)
```

Arguments

EvtRefDist	an object of class allowedFitDist that summarizes the distribution fitted to the marker values of patients that developed the event of interest in the reference arm. This class of objects is obtained using the fit function.
NoEvtRefDist	an object of class allowedFitDist that summarizes the distribution fitted to the marker values of patients that did not develop the event of interest in the reference arm. This class of objects is obtained using the fit function.
EvtInnovDist	an object of class allowedFitDist that summarizes the distribution fitted to the marker values of patients that developed the event of interest in the innovative arm. This class of objects is obtained using the fit function.
NoEvtInnovDist	an object of class allowedFitDist that summarizes the distribution fitted to the marker values of patients that did not develop the event of interest in the innovative arm. This class of objects is obtained using the fit function.
mRiskRef	an object of class mcmc.list provided by the user. It must be a sample of the posterior distribution of the mean risk of event in the reference treatment arm. If NULL, the function samples values in the posterior distribution of the mean risk of event in the reference arm assuming Jeffrey's prior (Beta(0.5,0.5)), and estimating the mean risk using the number of marker values specified in each treatment arm.
mRiskInnov	an object of class mcmc.list provided by the user. It must be a sample of the posterior distribution of the mean risk of event in the innovative treatment arm. If NULL, the function samples values in the posterior distribution of the mean risk of event in the innovative arm assuming Jeffrey's prior (Beta(0.5,0.5)), and estimating the mean risk using the number of marker values specified in each treatment arm..
lowRef	a logical value indicating whether low values of the marker are associated with low (TRUE) or high (FALSE) risk under the reference treatment arm.
toxRef	a logical value indicating whether the reference treatment arm (TRUE) or the innovative treatment arm (FALSE) must be preferred at equal efficacy taking into account toxicity.
r	a numeric value indicating the cost ratio between the most harmful treatment and the event (see Details).
le.MCMC	length of the desired MCMC chain.

hessTol	tolerance for the hessian value of the utility function at the optimal threshold.
plot	a logical value indicating whether routine graphics must be produced.
progress.bar	a character string indicating whether the user wishes to print a progress bar during the function process.
seed	a numerical value used to fix the random seed.

Details

When toxRef==FALSE then Janes et al. (2014) defined the costs of event and treatment as:

	Y=0	Y=1
Z=0	0	C_Y
Z=1	C_Z	$C_Z + C_Y$

When toxRef==TRUE it is defined as :

	Y=0	Y=1
Z=0	C_Z	$C_Z + C_Y$
Z=1	0	C_Y

According to the value of toxRef, the r ratio is simply $r = C_Z/C_Y$. The r ratio can also be indirectly specified by the absolute difference in risk of event between the two treatments above which a physician would recommend the use of the most harmful treatment. The inverse of the r ratio can also be interpreted as the number of patients for whom the physician is ready to give the most harmful treatment to prevent one additional case compared with the less harmful treatment.

Value

Returns an object of class trtSelOptThresh.

References

Blangero, Y, Rabilloud, M, Ecochard, R, and Subtil, F. A Bayesian method to estimate the optimal threshold of a marker used to select patients' treatment. *Statistical Methods in Medical Research*. 2019.

Examples

```
#Simulating data from four gaussian distributions,
#with mean risks equal to 0.5 in each arm:
x0E <- rnorm(250) # reference arm, event
x0Eb <- rnorm(250, 2) # reference arm, no event
x1E <- rnorm(250, 2) # innovative arm, event
x1Eb <- rnorm(250) # innovative arm, no event

#When working with real data. You can check the randomization constraint using the
#densCurves function:
densCurves(x0 = c(x0E, x0Eb), x1 = c(x1E, x1Eb), type = "treatment selection")
```

```
#You can also use the riskCurves function to know if low values of the marker are associated
#with a better response under the reference treatment or not:
library(mgcv)
riskCurves(x0E, x0Eb, x1E, x1Eb)

#Fit normal distributions on three groups. And let the last one (1E) be undefined (derived
#indirectly using the randomization constraint):
fit0E <- fit(x0E, "norm")
fit0Eb <- fit(x0Eb, "norm")
fit1E <- fit(x1E, "undefined")
fit1Eb <- fit(x1Eb, "norm")

#Apply the main function to estimate the optimal threshold:
# first case: the mean risks of event in the two treatment arms are left unspecified (are
# determined by the number of marker measurements in the fit0E, fi0Eb, fit1E, fit1Eb)

res <- trtSelThresh(fit0E, fit0Eb, fit1E, fit1Eb,
                    lowRef = FALSE, toxRef = FALSE, r = 0.02, le.MCMC = 5000, plot = TRUE,
                     progress.bar = "text")

# second case: the mean risks of event in the two treatment arms are given through mcmc.lists
# that correspond to their posterior distributions (see the fit man page for examples on how
# to generate posterior distributions manually)

#You can summarize the results using the summary() function:
summary(res, method = "median")

#You can extract the estimates and CI bounds of each indicator presented in the summary:
estimates(res, method = "median")
credibleIntervals(res)

#Plot the decision curves (this function is time-consuming):
dCres <- decisionCurve(res, r = seq(0, 0.2, length.out = 6))

#You can plot again the decision curves by applying the plot method to dCres,
#this function is a lot faster than the previous one. It also has more options
#to customize the plot:
plot(dCres)
plot(dCres, which = 1)
plot(dCres, which = 2)
```

`undefined-class` *An S4 class to represent an 'undefined' distribution.*

Description

This class allows to fit an undefined distribution on the marker values `x`.

Details

This class is automatically created when the user applies the `fit` function with the argument `distr="undefined"`. You never have to create manually this class, it is created internally.

Slots

`x` a vector of marker values.

`n` Length of x vector (including NA values). Numeric argument.

`mcmc` This slot allows the main function to k,now whether an MCMC algorithm must be performed to sample the distribution parameters from their posterior distribution.

See Also

`fit` for more details on how to fit an undefined distribution.

Index

allowedDist *(allowedDist-class)*, 3
allowedDist-class, 3
allowedFitDist *(allowedFitDist-class)*, 3
allowedFitDist-class, 3

cdf, 4
cdf,compoundEvtInnovDist-method *(cdf)*, 4
cdf,compoundEvtRefDist-method *(cdf)*, 4
cdf,compoundNoEvtInnovDist-method *(cdf)*, 4
cdf,compoundNoEvtRefDist-method *(cdf)*, 4
cdf,gammaDist-method *(cdf)*, 4
cdf,logisticDist-method *(cdf)*, 4
cdf,logNormalDist-method *(cdf)*, 4
cdf,normalDist-method *(cdf)*, 4
cdf,studentDist-method *(cdf)*, 4
cdf.compoundEvtInnovDist *(cdf)*, 4
cdf.compoundEvtRefDist *(cdf)*, 4
cdf.compoundNoEvtInnovDist *(cdf)*, 4
cdf.compoundNoEvtRefDist *(cdf)*, 4
cdf.gammaDist *(cdf)*, 4
cdf.logisticDist *(cdf)*, 4
cdf.logNormalDist *(cdf)*, 4
cdf.normalDist *(cdf)*, 4
cdf.studentDist *(cdf)*, 4
compoundEvtInnovDist *(compoundEvtInnovDist-class)*, 6
compoundEvtInnovDist-class, 6
compoundEvtRefDist *(compoundEvtRefDist-class)*, 6
compoundEvtRefDist-class, 6
compoundNoEvtInnovDist *(compoundNoEvtInnovDist-class)*, 7
compoundNoEvtInnovDist-class, 7
compoundNoEvtRefDist *(compoundNoEvtRefDist-class)*, 8
compoundNoEvtRefDist-class, 8
credibleIntervals, 9
credibleIntervals,diagOptThresh-method *(credibleIntervals)*, 9
credibleIntervals,trtSelOptThresh-method *(credibleIntervals)*, 9
credibleIntervals.diagOptThresh *(credibleIntervals)*, 9
credibleIntervals.trtSelOptThresh *(credibleIntervals)*, 9

decisionCurve, 10, *14*, *42*
decisionCurve,diagOptThresh-method *(decisionCurve)*, 10
decisionCurve,trtSelOptThresh-method *(decisionCurve)*, 10
decisionCurve.diagOptThresh *(decisionCurve)*, 10
decisionCurve.trtSelOptThresh *(decisionCurve)*, 10
densCurves, 11
dgamma, *27*
diagOptThresh *(diagOptThresh-class)*, 12
diagOptThresh-class, 12
diagRelUtility *(diagRelUtility-class)*, 13
diagRelUtility-class, 13
diagThresh, *3–5*, *9*, *11*, *13*, 14, *19*, *40*
dlnorm, *27*
dlogis, *27*
dnorm, *27*

estimates, 16
estimates,diagOptThresh-method *(estimates)*, 16
estimates,trtSelOptThresh-method *(estimates)*, 16
estimates.diagOptThresh *(estimates)*, 16
estimates.trtSelOptThresh *(estimates)*, 16

fit, *3–8*, *14*, 17, *21–25*, *27*, *29–31*, *38*, *42*, *43*, *46*

fitGammaDist *(fitGammaDist-class)*, 21
fitGammaDist-class, 21
fitLogisticDist
(fitLogisticDist-class), 21
fitLogisticDist-class, 21
fitLogNormalDist
(fitLogNormalDist-class), 22
fitLogNormalDist-class, 22
fitNormalDist *(fitNormalDist-class)*, 23
fitNormalDist-class, 23
fitStudentDist *(fitStudentDist-class)*,
23
fitStudentDist-class, 23
fitUserDefinedDist
(fitUserDefinedDist-class), 24
fitUserDefinedDist-class, 24

gam, *35*
gammaDist *(gammaDist-class)*, 25
gammaDist-class, 25
gradient, 25
gradient,compoundEvtInnovDist-method
(gradient), 25
gradient,compoundEvtRefDist-method
(gradient), 25
gradient,compoundNoEvtInnovDist-method
(gradient), 25
gradient,compoundNoEvtRefDist-method
(gradient), 25
gradient,gammaDist-method *(gradient)*, 25
gradient,logisticDist-method
(gradient), 25
gradient,logNormalDist-method
(gradient), 25
gradient,normalDist-method *(gradient)*,
25
gradient,studentDist-method *(gradient)*,
25
gradient.compoundEvtInnovDist
(gradient), 25
gradient.compoundEvtRefDist *(gradient)*,
25
gradient.compoundNoEvtInnovDist
(gradient), 25
gradient.compoundNoEvtRefDist
(gradient), 25
gradient.gammaDist *(gradient)*, 25
gradient.logisticDist *(gradient)*, 25
gradient.logNormalDist *(gradient)*, 25
gradient.normalDist *(gradient)*, 25
gradient.studentDist *(gradient)*, 25

hdi, *9*
hessian, 27
hessian,compoundEvtInnovDist-method
(hessian), 27
hessian,compoundEvtRefDist-method
(hessian), 27
hessian,compoundNoEvtInnovDist-method
(hessian), 27
hessian,compoundNoEvtRefDist-method
(hessian), 27
hessian,gammaDist-method *(hessian)*, 27
hessian,logisticDist-method *(hessian)*,
27
hessian,logNormalDist-method *(hessian)*,
27
hessian,normalDist-method *(hessian)*, 27
hessian,studentDist-method *(hessian)*, 27
hessian.compoundEvtInnovDist *(hessian)*,
27
hessian.compoundEvtRefDist *(hessian)*, 27
hessian.compoundNoEvtInnovDist
(hessian), 27
hessian.compoundNoEvtRefDist *(hessian)*,
27
hessian.gammaDist *(hessian)*, 27
hessian.logisticDist *(hessian)*, 27
hessian.logNormalDist *(hessian)*, 27
hessian.normalDist *(hessian)*, 27
hessian.studentDist *(hessian)*, 27

logisticDist *(logisticDist-class)*, 29
logisticDist-class, 29
logNormalDist *(logNormalDist-class)*, 30
logNormalDist-class, 30

mcmc.listOrNull
(mcmc.listOrNull-class), 30
mcmc.listOrNull-class, 30

normalDist *(normalDist-class)*, 30
normalDist-class, 30

pgamma, *5*
plnorm, *5*
plogis, *5*
plot,diagOptThresh-method
(plot-methods), 31

plot,diagRelUtility-method
(plot.diagRelUtility), 32
plot,trtSelOptThresh-method
(plot-methods), 31
plot,trtSelRelUtility-method
(plot.trtSelRelUtility), 33
plot-methods, 31
plot.diagOptThresh (plot-methods), 31
plot.diagRelUtility, 32
plot.trtSelOptThresh (plot-methods), 31
plot.trtSelRelUtility, 33
pnorm, *5*

riskCurves, 34

samplePosteriorDist, 36
samplePosteriorDist,fitGammaDist-method
(samplePosteriorDist), 36
samplePosteriorDist,fitLogisticDist-method
(samplePosteriorDist), 36
samplePosteriorDist,fitLogNormalDist-method
(samplePosteriorDist), 36
samplePosteriorDist,fitNormalDist-method
(samplePosteriorDist), 36
samplePosteriorDist,fitStudentDist-method
(samplePosteriorDist), 36
samplePosteriorDist.fitGammaDist
(samplePosteriorDist), 36
samplePosteriorDist.fitLogisticDist
(samplePosteriorDist), 36
samplePosteriorDist.fitLogNormalDist
(samplePosteriorDist), 36
samplePosteriorDist.fitNormalDist
(samplePosteriorDist), 36
samplePosteriorDist.fitStudentDist
(samplePosteriorDist), 36
show,diagOptThresh-method
(show-methods), 37
show,trtSelOptThresh-method
(show-methods), 37
show-methods, 37
show.diagOptThresh (show-methods), 37
show.trtSelOptThresh (show-methods), 37
studentDist (studentDist-class), 38
studentDist-class, 38
summary,diagOptThresh-method
(summary-methods), 39
summary,trtSelOptThresh-method
(summary-methods), 39
summary-methods, 39
summary.diagOptThresh
(summary-methods), 39
summary.trtSelOptThresh
(summary-methods), 39

trtSelOptThresh
(trtSelOptThresh-class), 40
trtSelOptThresh-class, 40
trtSelRelUtility
(trtSelRelUtility-class), 42
trtSelRelUtility-class, 42
trtSelThresh, *3–5*, *9*, *11*, *19*, *27*, *29*, *37*, *40*, *41*, 42

undefined (undefined-class), 45
undefined-class, 45