Package ‘predictNMB’

June 3, 2023

Type Package
Title Evaluate Clinical Prediction Models by Net Monetary Benefit
Version 0.2.1
Description Estimates when and where a model-guided treatment strategy may outperform a treat-all or treat-none approach by Monte Carlo simulation and evaluation of the Net Monetary Benefit. Details can be viewed in Parsons et al. (2023) <doi:10.21105/joss.05328>.
License GPL (>= 3)
Encoding UTF-8
RoxygenNote 7.2.3
Imports assertthat, cutpointr, dplyr, ggplot2, magrittr, pmsampsize, rlang, scales, stats, tibble, tidyr
Suggests spelling, covr, flextable, knitr, parallel, pbapply, rmarkdown, testthat (>= 3.0.0), vdiffr, withr
Config/testthat/edition 3
VignetteBuilder knitr
URL https://docs.ropensci.org/predictNMB/
BugReports https://github.com/ropensci/predictNMB/issues
Depends R (>= 3.5.0)
Language en-US
NeedsCompilation no
Author Rex Parsons [aut, cre] (https://orcid.org/0000-0002-6053-8174), Robin Blythe [aut] (https://orcid.org/0000-0002-3643-4332), Adrian Barnett [aut] (https://orcid.org/0000-0001-6339-0374), Emi Tanaka [rev] (Emi Tanaka reviewed predictNMB for rOpenSci, see https://github.com/ropensci/software-review/issues/566), Tinula Kariyawasam [rev] (Tinula Kariyawasam reviewed predictNMB for rOpenSci, see https://github.com/ropensci/software-review/issues/566), Susanna Cramb [ctb] (https://orcid.org/0000-0001-9041-9531), Steven McPhail [ctb] (https://orcid.org/0000-0002-1463-662X)
Maintainer  Rex Parsons <rex.parsons94@gmail.com>
Repository  CRAN
Date/Publication  2023-06-03 07:40:02 UTC

R topics documented:

- autoplot.predictNMBscreen
- autoplot.predictNMBsim
- ce_plot
- ce_plot.predictNMBsim
- do_nmb_sim
- evaluate_cutpoint_cost
- evaluate_cutpoint_nmb
- evaluate_cutpoint_qalys
- get_inbuilt_cutpoint
- get_inbuilt_cutpoint_methods
- get_nmb_sampler
- get_sample
- get_thresholds
- print.predictNMBscreen
- print.predictNMBsim
- screen_simulation_inputs
- summary.predictNMBscreen
- summary.predictNMBsim
- theme_sim

Index  26

autoplot.predictNMBscreen

Create plots of from screened predictNMB simulations.

Description

Create plots of from screened predictNMB simulations.

Usage

```r
# S3 method for class 'predictNMBscreen'
autoplot(
  object,
  x_axis_var = NULL,
  constants = list(),
  what = c("nmb", "inb", "cutpoints", "qalys", "costs"),
  inb_ref_col = NA,
  plot_range = TRUE,
)```
autoplot.predictNMBscreen

```r
plot_conf_level = TRUE,
plot_line = TRUE,
plot_alpha = 0.5,
dodge_width = 0,
conf.level = 0.95,
methods_order = NULL,
rename_vector,
...)
```

**Arguments**

- **object**
  A `predictNMBscreen` object.

- **x_axis_var**
  The desired screened factor to be displayed along the x axis. For example, if the simulation screen was used with many values for event rate, this could be "event_rate". Defaults to the first detected, varied input.

- **constants**
  Named vector. If multiple inputs were screened in this object, this argument can be used to modify the selected values for all those except the input that’s varying along the x-axis. See the summarising methods vignette.

- **what**
  What to summarise: one of "nmb", "inb", "cutpoints", "qalys" or "costs". Defaults to "nmb".

- **inb_ref_col**
  Which cutpoint method to use as the reference strategy when calculating the incremental net monetary benefit. See `do_nmb_sim` for more information.

- **plot_range** logical. Whether or not to plot the range of the distribution as a thin line. Defaults to TRUE.

- **plot_conf_level** logical. Whether or not to plot the confidence region of the distribution as a thicker line. Defaults to TRUE.

- **plot_line** logical. Whether or not to connect the medians of the distributions for each method along the x-axis. Defaults to TRUE.

- **plot_alpha** Alpha value (transparency) of all plot elements. Defaults to 0.5.

- **dodge_width** The dodge width of plot elements. Can be used to avoid excessive overlap between methods. Defaults to 0.

- **conf.level** The confidence level of the interval. Defaults to 0.95 (coloured area of distribution represents 95% CIs).

- **methods_order** The order (left to right) to display the cutpoint methods.

- **rename_vector** A named vector for renaming the methods in the summary. The values of the vector are the default names and the names given are the desired names in the output.

- **...** Additional (unused) arguments.

**Details**

This plot method works with `predictNMBscreen` objects that are created using `screen_simulation_inputs()`. Can be used to visualise distributions from many different simulations and assign a varying input to the x-axis of the plot.
Value

Returns a ggplot object.

Examples

get_nmb <- function() c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)
sim_screen_obj <- screen_simulation_inputs(
  n_sims = 50, n_valid = 10000, sim_auc = seq(0.7, 0.9, 0.1),
  event_rate = c(0.1, 0.2, 0.3),
  fx_nmb_training = get_nmb, fx_nmb_evaluation = get_nmb,
  cutpoint_methods = c("all", "none", "youden", "value_optimising")
)

autoplot(sim_screen_obj)

autoplot(
  sim_screen_obj,
  x_axis_var = "event_rate",
  constants = c(sim_auc = 0.8),
  dodge_width = 0.02,
  rename_vector = c(
    "Value-Optimising" = "value_optimising",
    "Treat-None" = "none",
    "Treat-All" = "all",
    "Youden Index" = "youden"
  )
)

--

autoplot.predictNMBsim

Create plots of from predictNMB simulations.

Description

Create plots of from predictNMB simulations.

Usage

## S3 method for class 'predictNMBsim'
autoplot(
  object,
  what = c("nmb", "inb", "cutpoints", "qalys", "costs"),
  inb_ref_col = NA,
  conf.level = 0.95,
  methods_order = NULL,
  n_bins = 40,
  label_wrap_width = 12,
)
Arguments

- **object**: A `predictNMBsim` object.
- **what**: What to summarise: one of "nmb", "inb", "cutpoints", "qalys" or "costs". Defaults to "nmb".
- **inb_ref_col**: Which cutpoint method to use as the reference strategy when calculating the incremental net monetary benefit. See `do_nmb_sim()` for more information.
- **conf.level**: The confidence level of the interval. Defaults to 0.95 (coloured area of distribution represents 95% CIs).
- **methods_order**: The order (left to right) to display the cutpoint methods.
- **n_bins**: The number of bins used when constructing histograms. Defaults to 40.
- **label_wrap_width**: The number of characters in facet labels at which the label is wrapped. Default is 12.
- **fill_cols**: Vector containing the colours used for fill aesthetic of histograms. The first colour represents the area outside of the confidence region, second colour shows the confidence region. Defaults to `c("grey50", "#ADD8E6")`.
- **median_line_size**: Size of line used to represent the median of distribution. Defaults to 2.
- **median_line_alpha**: Alpha (transparency) for line used to represent the median of distribution. Defaults to 0.5.
- **median_line_col**: Colour of line used to represent the median of distribution. Defaults to "black".
- **rename_vector**: A named vector for renaming the methods in the summary. The values of the vector are the default names and the names given are the desired names in the output.
- **...**: Additional (unused) arguments.

Details

This plot method works with `predictNMBsim` objects that are created using `do_nmb_sim()`. Can be used to visualise distributions from simulations for different cutpoint methods.

Value

Returns a `ggplot` object.
Examples

get_nmb <- function() c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)
sim_obj <- do_nmb_sim(
    sample_size = 200, n_sims = 50, n_valid = 10000, sim_auc = 0.7,
    event_rate = 0.1, fx_nmb_training = get_nmb, fx_nmb_evaluation = get_nmb,
    cutpoint_methods = c("all", "none", "youden", "value_optimising")
)

autoplot(
    sim_obj,
    rename_vector = c(
        "Value- Optimising" = "value_optimising",
        "Treat- None" = "none",
        "Treat- All" = "all",
        "Youden Index" = "youden"
    )
) + theme_sim()

ce_plot

Create a cost-effectiveness plot.

Description

Create a cost-effectiveness plot.

Usage

ce_plot(
    object,
    ref_col,
    wtp,
    show_wtp = TRUE,
    methods_order = NULL,
    rename_vector,
    shape = 21,
    wtp_linetype = "dashed",
    add_prop_ce = FALSE,
    ...
)

Arguments

object A predictNMBsim object.
ref_col Which cutpoint method to use as the reference strategy when calculating the incremental net monetary benefit. Often sensible to use a "all" or "none" approach for this.
ce_plot

wtp A numeric. The willingness to pay (WTP) value used to create a WTP threshold line on the plot (if show_wtp = TRUE). Defaults to the WTP stored in the predictNMBsim object.

show_wtp A logical. Whether or not to show the willingness to pay threshold.

methods_order The order (within the legend) to display the cutpoint methods.

rename_vector A named vector for renaming the methods in the summary. The values of the vector are the default names and the names given are the desired names in the output.

shape The shape used for ggplot2::geom_point(). Defaults to 21 (hollow circles). If shape = "method" or shape = "cost-effective" (only applicable when show_wtp = TRUE), then the shape will be mapped to that aesthetic.

wtp_linetype The linetype used for ggplot2::geom_abline() when making the WTP. Defaults to "dashed".

add_prop_ce Whether to append the proportion of simulations for that method which were cost-effective (beneath the WTP threshold) to their labels in the legend. Only applicable when show_wtp = TRUE.

... Additional (unused) arguments.

Details

This plot method works with predictNMBsim objects that are created using do_nmb_sim(). Can be used to visualise the simulations on a cost-effectiveness plot (costs vs effectiveness).

Value

Returns a ggplot object.

Examples

get_nmb_evaluation <- get_nmb_sampler(
  qalys_lost = function() rnorm(1, 0.33, 0.03),
  wtp = 28000,
  high_risk_group_treatment_effect = function() exp(rnorm(n = 1, mean = log(0.58), sd = 0.43)),
  high_risk_group_treatment_cost = function() rnorm(n = 1, mean = 161, sd = 49)
)

sim_obj <- do_nmb_sim(
  sample_size = 200, n_sims = 50, n_valid = 10000, sim_auc = 0.7,
  event_rate = 0.1, fx_nmb_training = get_nmb_evaluation, fx_nmb_evaluation = get_nmb_evaluation
)

ce_plot(sim_obj, ref_col = "all")
ce_plot.predictNMBsim  Create a cost-effectiveness plot.

Description

Create a cost-effectiveness plot.

Usage

## S3 method for class 'predictNMBsim'
ce_plot(
  object,
  ref_col,
  wtp,
  show_wtp = TRUE,
  methods_order = NULL,
  rename_vector,
  shape = 21,
  wtp_linetype = "dashed",
  add_prop_ce = FALSE,
  ...
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>A predictNMBsim object.</td>
</tr>
<tr>
<td>ref_col</td>
<td>Which cutpoint method to use as the reference strategy when calculating the incremental net monetary benefit. Often sensible to use a &quot;all&quot; or &quot;none&quot; approach for this.</td>
</tr>
<tr>
<td>wtp</td>
<td>A numeric. The willingness to pay (WTP) value used to create a WTP threshold line on the plot (if show_wtp = TRUE). Defaults to the WTP stored in the predictNMBsim object.</td>
</tr>
<tr>
<td>show_wtp</td>
<td>A logical. Whether or not to show the WTP threshold.</td>
</tr>
<tr>
<td>methods_order</td>
<td>The order (within the legend) to display the cutpoint methods.</td>
</tr>
<tr>
<td>rename_vector</td>
<td>A named vector for renaming the methods in the summary. The values of the vector are the default names and the names given are the desired names in the output.</td>
</tr>
<tr>
<td>shape</td>
<td>The shape used for ggplot2::geom_point(). Defaults to 21 (hollow circles). If shape = &quot;method&quot; or shape = &quot;cost-effective&quot; (only applicable when show_wtp = TRUE), then the shape will be mapped to that aesthetic.</td>
</tr>
<tr>
<td>wtp_linetype</td>
<td>The linetype used for ggplot2::geom_abline() when making the WTP.Defaults to &quot;dashed&quot;.</td>
</tr>
<tr>
<td>add_prop_ce</td>
<td>Whether to append the proportion of simulations for that method which were cost-effective (beneath the WTP threshold) to their labels in the legend. Only applicable when show_wtp = TRUE.</td>
</tr>
<tr>
<td>...</td>
<td>Additional (unused) arguments.</td>
</tr>
</tbody>
</table>
**do_nmb_sim**

*Do the predictNMB simulation, evaluating the net monetary benefit (NMB) of the simulated model.*

**Details**

This plot method works with `predictNMBsim` objects that are created using `do_nmb_sim()`. Can be used to visualise the simulations on a cost-effectiveness plot (costs vs. effectiveness).

**Value**

Returns a `ggplot` object.

**Examples**

```r
get_nmb_evaluation <- get_nmb_sampler(
  qalys_lost = function() rnorm(1, 0.33, 0.03),
  wtp = 28000,
  high_risk_group_treatment_effect = function() exp(rnorm(n = 1, mean = log(0.58), sd = 0.43)),
  high_risk_group_treatment_cost = function() rnorm(n = 1, mean = 161, sd = 49)
)

sim_obj <- do_nmb_sim(
  sample_size = 200, n_sims = 50, n_valid = 10000, sim_auc = 0.7,
  event_rate = 0.1, fx_nmb_training = get_nmb_evaluation, fx_nmb_evaluation = get_nmb_evaluation
)

ce_plot(sim_obj, ref_col = "all")
```

---

**do_nmb_sim**

*Do the predictNMB simulation, evaluating the net monetary benefit (NMB) of the simulated model.*

**Description**

Do the predictNMB simulation, evaluating the net monetary benefit (NMB) of the simulated model.

**Usage**

```r
do_nmb_sim(
  sample_size,
  n_sims,
  n_valid,
  sim_auc,
  event_rate,
  cutpoint_methods = get_inbuilt_cutpoint_methods(),
  fx_nmb_training,
  fx_nmb_evaluation,
  meet_min_events = TRUE,
  min_events = NA,
  show_progress = FALSE,
)```
do_nmb_sim

cl = NULL
)

Arguments

sample_size  Sample size of training set. If missing, a sample size calculation will be performed and the calculated size will be used.
n_sims        Number of simulations to run.
n_valid       Sample size for evaluation set.
sim_auc       Simulated model discrimination (AUC).
event_rate    Simulated event rate of the binary outcome being predicted. Also known as prevalence.
cutpoint_methods
A value or vector of cutpoint methods to include. Defaults to use the inbuilt methods:
  • "all" = treat all patients (cutpoint = 0)
  • "none" = treat no patients (cutpoint = 1)
  • "value_optimising" = select the cutpoint that maximises NMB
  • "youden" = select cutpoint based on the Youden index, also known as the J-index (sensitivity + specificity - 1)
  • "cost_minimising" = select the cutpoint that minimises expected value of costs
  • "prod_sens_spec" = product of sensitivity and specificity (sensitivity * specificity)
  • "roc01" = selects the closest threshold to the (0,1) point on the ROC curve
User-defined cutpoint methods can be used by passing the name of a function that takes the following arguments:
  • predicted (predicted probabilities)
  • actual (the actual, binary outcome)
  • nmb (a named vector containing NMB values assigned to each predicted class (i.e. c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)))

See ?get_thresholds for an example of a user-defined cutpoint function.

fx_nmb_training
Function or NMBsampler that returns a named vector of NMB assigned to classifications used for obtaining cutpoint on training set.

fx_nmb_evaluation
Function or NMBsampler that returns a named vector of NMB assigned to classifications used for obtaining cutpoint on evaluation set.

meet_min_events
Whether or not to incrementally add samples until the expected number of events (sample_size * event_rate) is met. (Applies to sampling of training data only.)
evaluate_cutpoint_cost

min_events  The minimum number of events to include in the training sample. If less than this number are included in sample of size sample_size, additional samples are added until the min_events is met. The default (NA) will use the expected value given the event_rate and the sample_size.

show_progress  Logical. Whether to display a progress bar. Requires the pbapply package.

cl  A cluster made using parallel::makeCluster(). If a cluster is provided, the simulation will be done in parallel.

Details

This function runs a simulation for a given set of inputs that represent a healthcare setting using model-guided interventions.

The arguments fx_nmb_training and fx_nmb_evaluation should be functions that capture the treatment being used, its costs and effectiveness, and the costs of the outcome being treated/prevented.

Both of these are functions that return a named vector of NMB values when called and are used for obtaining and evaluating cutpoints, respectively. For example, the following function returns the appropriately named vector.

get_nmb <- function() c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)

There is a helper function, get_nmb_sampler(), to help you create these.

Value

Returns a predictNMBsim object.

Examples

get_nmb <- function() c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)
do_nmb_sim(
  sample_size = 200, n_sims = 50, n_valid = 10000, sim_auc = 0.7,
  event_rate = 0.1, fx_nmb_training = get_nmb, fx_nmb_evaluation = get_nmb
)

evaluate_cutpoint_cost

Evaluates a cutpoint by returning the mean treatment cost per sample.

Description

Evaluates a cutpoint by returning the mean treatment cost per sample.

Usage

evaluate_cutpoint_cost(predicted, actual, pt, nmb)
evaluate_cutpoint_nmb

Evaluates a cutpoint by returning the mean NMB per sample.

Description

Evaluates a cutpoint by returning the mean NMB per sample.

Usage

evaluate_cutpoint_nmb(predicted, actual, pt, nmb)

Arguments

predicted A vector of predicted probabilities.
actual A vector of actual outcomes.
pt The probability threshold to be evaluated.
nmb A named vector containing NMB assigned to each classification.
evaluate_cutpoint_qalys

**Value**

Returns a numeric value representing the NMB for that cutpoint and data.

**Examples**

evaluate_cutpoint_nmb(
    predicted = runif(1000),
    actual = sample(c(0, 1), size = 1000, replace = TRUE),
    pt = 0.1,
    nmb = c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)
)

evaluate_cutpoint_qalys

*Evaluates a cutpoint by returning the mean QALYs lost per sample.*

**Description**

Evaluates a cutpoint by returning the mean QALYs lost per sample.

**Usage**

evaluate_cutpoint_qalys(predicted, actual, pt, nmb)

**Arguments**

- **predicted**: A vector of predicted probabilities.
- **actual**: A vector of actual outcomes.
- **pt**: The probability threshold to be evaluated.
- **nmb**: A named vector containing NMB assigned to each classification and the treatment effects and QALYS lost due to the event of interest.

**Value**

Returns a numeric value representing the mean QALYs for that cutpoint and data.

**Examples**

evaluate_cutpoint_qalys(
    predicted = runif(1000),
    actual = sample(c(0, 1), size = 1000, replace = TRUE),
    pt = 0.1,
    nmb = c("qalys_lost" = 5,
             "low_risk_group_treatment_effect" = 0,
             "high_risk_group_treatment_effect" = 0.5)
)
)
get_inbuilt_cutpoint  Get a cutpoint using the methods inbuilt to predictNMB

Description

Get a cutpoint using the methods inbuilt to predictNMB

Usage

get_inbuilt_cutpoint(predicted, actual, nmb, method)

Arguments

- `predicted`: A vector of predicted probabilities
- `actual`: A vector of actual outcomes
- `nmb`: A named vector containing NMB assigned to each classification
- `method`: A cutpoint selection method to be used; methods that can be used as the method argument

Value

Returns a selected cutpoint (numeric).

Examples

```r
## get the list of available methods:
get_inbuilt_cutpoint_methods()

## get the cutpoint that maximises the Youden index for a given set of
## probabilities and outcomes
get_inbuilt_cutpoint(
  predicted = runif(1000),
  actual = sample(c(0, 1), size = 1000, replace = TRUE),
  method = "youden"
)
```

get_inbuilt_cutpoint_methods  Get a vector of all the inbuilt cutpoint methods

Description

Get a vector of all the inbuilt cutpoint methods
**Usage**

```r
get_inbuilt_cutpoint_methods()
```

**Value**

Returns a vector cutpoint methods that can be used in `do_nmb_sim()`.

**Examples**

```r
get_inbuilt_cutpoint_methods()
```

---

**get_nmb_sampler**

_Make a NMB sampler for use in `do_nmb_sim()` or `screen_simulation_inputs()`_

**Description**

Make a NMB sampler for use in `do_nmb_sim()` or `screen_simulation_inputs()`.

**Usage**

```r
get_nmb_sampler(
  outcome_cost,
  wtp,
  qalys_lost,
  high_risk_group_treatment_effect,
  high_risk_group_treatment_cost,
  low_risk_group_treatment_effect = 0,
  low_risk_group_treatment_cost = 0,
  use_expected_values = FALSE,
  nboot = 10000
)
```

**Arguments**

- **outcome_cost**: The cost of the outcome. Must be provided if `wtp` and `qalys_lost` are not. Or can be used in addition to these arguments to represent additional cost to the health burden.
- **wtp**: Willingness-to-pay.
- **qalys_lost**: Quality-adjusted life years (QALYs) lost due to healthcare event being predicted.
- **high_risk_group_treatment_effect**: The effect of the treatment provided to patients given high risk prediction. Can be a number of a function. Provide a function to incorporate uncertainty.
- **high_risk_group_treatment_cost**: The cost of the treatment provided to patients given high risk prediction. Can be a number of a function. Provide a function to incorporate uncertainty.
get_sample

**Description**

Samples data for a prediction model with a specified AUC and prevalence.
get_thresholds

Usage

get_sample(auc, n_samples, prevalence, min_events = 0)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>auc</td>
<td>The Area Under the (receiver operating characteristic) Curve.</td>
</tr>
<tr>
<td>n_samples</td>
<td>Number of samples to draw.</td>
</tr>
<tr>
<td>prevalence</td>
<td>Prevalence or event rate of the binary outcome as a proportion (0.1 = 10%).</td>
</tr>
<tr>
<td>min_events</td>
<td>Minimum number of events required in the sample.</td>
</tr>
</tbody>
</table>

Value

Returns a data.frame.

Examples

get_sample(0.7, 1000, 0.1)

get_thresholds

Gets probability thresholds given predicted probabilities, outcomes and NMB.

Description

Gets probability thresholds given predicted probabilities, outcomes and NMB.

Usage

get_thresholds(predicted, actual, nmb, cutpoint_methods = NULL)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>predicted</td>
<td>A vector of predicted probabilities.</td>
</tr>
<tr>
<td>actual</td>
<td>A vector of actual outcomes.</td>
</tr>
<tr>
<td>nmb</td>
<td>A named vector containing NMB assigned to true positives, true negatives, false positives and false negatives</td>
</tr>
<tr>
<td>cutpoint_methods</td>
<td>Which cutpoint method(s) to return. The default (NULL) uses all the inbuilt methods.</td>
</tr>
</tbody>
</table>

Value

Returns a list.
Examples

# get thresholds using default (all inbuilt) cutpoint methods
get_thresholds(
  predicted = runif(1000),
  actual = sample(c(0, 1), size = 1000, replace = TRUE),
  nmb = c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)
)

# get cutpoints using user-defined functions
# These functions must take the \code{predicted} and \code{actual}
# as arguments. They can also take \code{nmb} (named vector containing NMB
# with values for TP, FP, TN, FN).
fx_roc01 <- function(predicted, actual, ...) {
  cutpointr::cutpointr(
    x = predicted, class = actual, method = cutpointr::minimize_metric,
    metric = cutpointr::roc01,
    silent = TRUE
  )[["optimal_cutpoint"]]
}

fx_sum_sens_spec <- function(predicted, actual, ...) {
  cutpointr::cutpointr(
    x = predicted, class = actual, method = cutpointr::maximize_metric,
    metric = cutpointr::sum_sens_spec,
    silent = TRUE
  )[["optimal_cutpoint"]]
}

get_thresholds(
  predicted = runif(1000),
  actual = sample(c(0, 1), size = 1000, replace = TRUE),
  cutpoint_methods = c("fx_roc01", "fx_sum_sens_spec"),
  nmb = c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)
)

# get a combination of cutpoints from both user-defined functions and
# inbuilt methods
get_thresholds(
  predicted = runif(1000),
  actual = sample(c(0, 1), size = 1000, replace = TRUE),
  cutpoint_methods = c(  
    "fx_roc01",  
    "fx_sum_sens_spec",  
    "youden",  
    "all",  
    "none"
  ),
  nmb = c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)
)
print.predictNMBscreen

*Print a summary of a predictNMBscreen object*

**Description**

Print a summary of a predictNMBscreen object

**Usage**

```r
## S3 method for class 'predictNMBscreen'
print(x, ...)
```

**Arguments**

- `x` A `predictNMBscreen` object.
- `...` Optional, ignored arguments.

**Value**

`print(x)` returns `x` invisibly.

**Examples**

```r
get_nmb <- function() c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)
sim_screen_obj <- screen_simulation_inputs(
  n_sims = 50, n_valid = 10000, sim_auc = seq(0.7, 0.9, 0.1),
  event_rate = 0.1,
  fx_nmb_training = get_nmb, fx_nmb_evaluation = get_nmb
)
print(sim_screen_obj)
```

---

print.predictNMBsim

*Print a summary of a predictNMBsim object*

**Description**

Print a summary of a predictNMBsim object

**Usage**

```r
## S3 method for class 'predictNMBsim'
print(x, ...)
```

**Examples**

```r
get_nmb <- function() c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)
sim_screen_obj <- screen_simulation_inputs(
  n_sims = 50, n_valid = 10000, sim_auc = seq(0.7, 0.9, 0.1),
  event_rate = 0.1,
  fx_nmb_training = get_nmb, fx_nmb_evaluation = get_nmb
)
print(sim_screen_obj)
```
Arguments

x A `predictNMBsim` object.

... Optional, ignored arguments.

Value

`print(x)` returns `x` invisibly.

Examples

```r
get_nmb <- function() c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)
sim_obj <- do_nmb_sim(
  sample_size = 200, n_sims = 50, n_valid = 10000, sim_auc = 0.7,
  event_rate = 0.1, fx_nmb_training = get_nmb, fx_nmb_evaluation = get_nmb
)
print(sim_obj)
```

Description

Runs `do_nmb_sim` with a range of inputs.

Usage

```r
screen_simulation_inputs(
  sample_size, n_sims, n_valid, sim_auc, event_rate,
  cutpoint_methods = get_inbuilt_cutpoint_methods(),
  fx_nmb_training, fx_nmb_evaluation,
  pair_nmb_train_and_evaluation_functions = FALSE,
  meet_min_events = TRUE,
  min_events = NA,
  show_progress = FALSE,
  cl = NULL
)
```
Arguments

sample_size  A value (or vector of values): Sample size of training set. If missing, a sample size calculation will be performed and the calculated size will be used.
n_sims  A value (or vector of values): Number of simulations to run.
n_valid  A value (or vector of values): Sample size for evaluation set.
sim_auc  A value (or vector of values): Simulated model discrimination (AUC).
event_rate  A value (or vector of values): simulated event rate of the binary outcome being predicted.
cutpoint_methods  cutpoint methods to include. Defaults to use the inbuilt methods. This doesn’t change across calls to do_nmb_sim().
fx_nmb_training  A function or NMBsampler (or list of) that returns named vector of NMB assigned to classifications use for obtaining cutpoint on training set.
fx_nmb_evaluation  A function or NMBsampler (or list of) that returns named vector of NMB assigned to classifications use for obtaining cutpoint on evaluation set.
pair_nmb_train_and_evaluation_functions  logical. Whether or not to pair the lists of functions passed for fx_nmb_training and fx_nmb_evaluation. If two treatment strategies are being used, it may make more sense to pair these because selecting a value-optimising or cost-minimising threshold using one strategy but evaluating another is likely unwanted.
meet_min_events  Whether or not to incrementally add samples until the expected number of events (sample_size * event_rate) is met. (Applies to sampling of training data only.)
min_events  A value: the minimum number of events to include in the training sample. If less than this number are included in sample of size sample_size, additional samples are added until the min_events is met. The default (NA) will use the expected value given the event_rate and the sample_size.
show_progress  Logical. Whether to display a progress bar.
cl  A cluster made using parallel::makeCluster(). If a cluster is provided, the simulation will be done in parallel.

Value

Returns a predictNMBscreen object.

Examples

# Screen for optimal cutpoints given increasing values of # model discrimination (sim_auc)

get_nmb <- function() c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)
sim_screen_obj <- screen_simulation_inputs(
  n_sims = 50, n_valid = 10000, sim_auc = seq(0.7, 0.9, 0.1),
  event_rate = 0.1, fx_nmb_training = get_nmb, fx_nmb_evaluation = get_nmb
)

summary.predictNMBscreen

Create table summaries of predictNMBscreen objects.

Description

Create table summaries of predictNMBscreen objects.

Usage

## S3 method for class 'predictNMBscreen'
summary(
  object,
  what = c("nmb", "inb", "cutpoints"),
  inb_ref_col = NULL,
  agg_functions = list(median = function(x) {
    round(stats::median(x), digits = 2)
  }, `95% CI` = function(x) {
    paste0(round(stats::quantile(x, probs = c(0.025, 0.975))), collapse = " to ")
  }),
  rename_vector,
  show_full_inputs = FALSE,
  ...
)

Arguments

- `object`: A predictNMBscreen object.
- `what`: What to summarise: one of "nmb", "inb" or "cutpoints". Defaults to "nmb".
- `inb_ref_col`: Which cutpoint method to use as the reference strategy when calculating the incremental net monetary benefit. See do_nmb_sim for more information.
- `agg_functions`: A named list of functions to use to aggregate the selected values. Defaults to the median and 95% interval.
- `rename_vector`: A named vector for renaming the methods in the summary. The values of the vector are the default names and the names given are the desired names in the output.
- `show_full_inputs`: A logical. Whether or not to include the inputs used for simulation alongside aggregations.
- `...`: Additional, ignored arguments.
Summary of predictNMBsim objects.

Details

Table summaries will be based on the `what` argument. Using "nmb" returns the simulated values for NMB, with no reference group; "inb" returns the difference between simulated values for NMB and a set strategy defined by `inb_ref_col`; "cutpoints" returns the cutpoints selected (0, 1).

Value

Returns a tibble.

Examples

```r
# perform screen with increasing values of model discrimination (sim_auc)

get_nmb <- function() c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)
sim_screen_obj <- screen_simulation_inputs(
  n_sims = 50, n_valid = 10000, sim_auc = seq(0.7, 0.9, 0.1),
  event_rate = 0.1, fx_nmb_training = get_nmb, fx_nmb_evaluation = get_nmb,
  cutpoint_methods = c("all", "none", "youden", "value_optimising")
)
summary(
sim_screen_obj,
rename_vector = c(
  "Value_Optimising" = "value_optimising",
  "Treat_None" = "none",
  "Treat_All" = "all",
  "Youden_Index" = "youden"
)
)
```

---

**Summary of predictNMBsim objects.**

Description

Create table summaries of predictNMBsim objects.

Usage

```r
## S3 method for class 'predictNMBsim'
summary(
  object,
  what = c("nmb", "inb", "cutpoints"),
inb_ref_col = NULL,
agg_functions = list(median = function(x) {
  round(stats::median(x), digits = 2)
}))
```
\textbf{summary.predictNMBsim}

\begin{verbatim}
}, '95% CI' = function(x) {
paste0(round(stats::quantile(x, probs = c(0.025, 0.975)), digits = 1), collapse = " to ")
}), rename_vector,
...
)

\textbf{Arguments}

\begin{itemize}
  \item \textbf{object} \hspace{1cm} A \texttt{predictNMBsim} object.
  \item \textbf{what} \hspace{1cm} What to summarise: one of "nmb", "inb" or "cutpoints". Defaults to "nmb".
  \item \textbf{inb_ref_col} \hspace{1cm} Which cutpoint method to use as the reference strategy when calculating the incremental net monetary benefit. See \texttt{do_nmb_sim} for more information.
  \item \textbf{agg_functions} \hspace{1cm} A named list of functions to use to aggregate the selected values. Defaults to the median and 95\% interval.
  \item \textbf{rename_vector} \hspace{1cm} A named vector for renaming the methods in the summary. The values of the vector are the default names and the names given are the desired names in the output.
  \item ... \hspace{1cm} Additional, ignored arguments.
\end{itemize}

\textbf{Details}

Table summaries will be based on the what argument. Using "nmb" returns the simulated values for NMB, with no reference group; "inb" returns the difference between simulated values for NMB and a set strategy defined by \texttt{inb_ref_col}; "cutpoints" returns the cutpoints selected (0, 1).

\textbf{Value}

Returns a \texttt{tibble}.

\textbf{Examples}

\begin{verbatim}
# perform simulation with \texttt{do_nmb_sim()}

get_nmb <- function() c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)
sim_obj <- do_nmb_sim(
  sample_size = 200, n_sims = 50, n_valid = 10000, sim_auc = 0.7,
  event_rate = 0.1, fx_nmb_training = get_nmb, fx_nmb_evaluation = get_nmb,
  cutpoint_methods = c("all", "none", "youden", "value_optimising")
)
summary(
  sim_obj,
  rename_vector = c(
    "Value_Optimising" = "value_optimising",
    "Treat_None" = "none",
    "Treat_All" = "all",
    "Youden_Index" = "youden"
  )
)
\end{verbatim}
theme_sim

Returns a ggplot2 theme that reduces clutter in an autoplot() of a predictNMBsim object.

Description

Returns a ggplot2 theme that reduces clutter in an autoplot() of a predictNMBsim object.

Usage

theme_sim()

Value

Returns a ggplot2 theme.

Examples

get_nmb <- function() c("TP" = -3, "TN" = 0, "FP" = -1, "FN" = -4)
sim_obj <- do_nmb_sim(
  sample_size = 200, n_sims = 50, n_valid = 10000, sim_auc = 0.7,
  event_rate = 0.1, fx_nmb_training = get_nmb, fx_nmb_evaluation = get_nmb
)

autoplot(sim_obj) + theme_sim()
Index

autoplot.predictNMBscreen, 2
autoplot.predictNMBsim, 4

ce_plot, 6
ce_plot.predictNMBsim, 8

do_nmb_sim, 9

evaluate_cutpoint_cost, 11
evaluate_cutpoint_nmb, 12
evaluate_cutpoint_qalys, 13

get_inbuilt_cutpoint, 14
get_inbuilt_cutpoint_methods, 14
get_nmb_sampler, 15
get_sample, 16
get_thresholds, 17

print.predictNMBscreen, 19
print.predictNMBsim, 19

screen_simulation_inputs, 20
summary.predictNMBscreen, 22
summary.predictNMBsim, 23

theme_sim, 25