Package ‘BCEA’

October 28, 2021

Type Package

Title Bayesian Cost Effectiveness Analysis

Version 2.4.1

Date 2021-10-14

Imports dplyr, ggplot2, GrassmannOptim, gridExtra, graphics, MASS, purrr, reshape2, rlang

Depends R (>= 3.5.0)

Suggests coda, grid, INLA, knitr, markdown, Matrix, MCMCvis, mgcv, plotly, rmarkdown, rstan, RColorBrewer, R2jags, splancs, testthat (>= 2.1.0), vdiffr, withr

Additional_repositories `https://inla.r-inla-download.org/R/stable/`

Description Produces an economic evaluation of a sample of suitable variables of cost and effectiveness / utility for two or more interventions, e.g. from a Bayesian model in the form of MCMC simulations. This package computes the most cost-effective alternative and produces graphical summaries and probabilistic sensitivity analysis.

URL `http://www.statistica.it/gianluca/software/bcea/`,
`http://www.statistica.it/gianluca/`,
`https://github.com/giabaio/BCEA`

NeedsCompilation no

RoxygenNote 7.1.2

Encoding UTF-8

Author Gianluca Baio [aut, cre],
Andrea Berardi [aut],
Anna Heath [aut],
Nathan Green [aut]

Maintainer Gianluca Baio `<gianluca@stats.ucl.ac.uk>`

Repository CRAN

Date/Publication 2021-10-28 13:50:05 UTC

R topics documented:

`BCEA-package` *BCEA: A package for Bayesian Cost-Effectiveness Analysis*

Description

A package to post-process the results of a Bayesian health economic model and produce standardised output for the analysis of the results.

Details

Package:	BCEA
Type:	Package
Version:	2.4.1
Date:	2021-02-05
License:	GPL2
LazyLoad:	Yes

BCEA produces a health economic evaluation given a random sample of suitable variables of costs and clinical benefits for two or more interventions, e.g. using results of a Bayesian model (possibly based on MCMC) in the form of simulations from the posterior distributions. Compares one of the interventions (the "reference") to the others ("comparators"). Produces many summaries and plots to analyse the results.

Author(s)

Gianluca Baio, Andrea Berardi, Anna Heath, Nathan Green

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

Baio G., Berardi A., Heath A. (2017). Bayesian Cost Effectiveness Analysis with the R package BCEA. Springer

---

`bcea` *Create Bayesian Cost-Effectiveness Analysis Object*

Description

Cost-effectiveness analysis based on the results of a simulation model for a variable of clinical benefits (e) and of costs (c). Produces results to be post-processed to give the health economic analysis. The output is stored in an object of the class "bcea".

Usage

```
bcea(
  eff,
  cost,
  ref = 1,
  interventions = NULL,
  .comparison = NULL,
  Kmax = 50000,
  wtp = NULL,
  plot = FALSE
)

## Default S3 method:
bcea(
  eff,
  cost,
  ref = NULL,
  interventions = NULL,
  .comparison = NULL,
  Kmax = 50000,
  wtp = NULL,
  plot = FALSE
)

## S3 method for class 'rjags'
bcea(eff, ...)

## S3 method for class 'rstan'
bcea(eff, ...)

## S3 method for class 'bugs'
bcea(eff, ...)
```

Arguments

`eff`	An object containing `nsim` simulations for the variable of clinical effectiveness for each intervention being considered. In general it is a matrix with `nsim` rows

	and `nint` columns. This are partially matched with ‘e’ from previous version of ‘BCEA‘ for back-compatibility.
`cost`	An object containing `nsim` simulations for the variable of cost for each intervention being considered. In general it is a matrix with `nsim` rows and `nint` columns. This are partially matched with ‘c’ from previous version of ‘BCEA‘ for back-compatibility.
`ref`	Defines which intervention (columns of `eff` or `cost`) is considered to be the reference strategy. The default value `ref = 1` means that the intervention associated with the first column of `eff` or `cost` is the reference and the one(s) associated with the other column(s) is(are) the comparators.
`interventions`	Defines the labels to be associated with each intervention. By default and if `NULL`, assigns labels in the form "Intervention1", ... , "InterventionT".
`.comparison`	Selects the comparator, in case of more than two interventions being analysed. Default as NULL plots all the comparisons together. Any subset of the possible comparisons can be selected (e.g., `comparison=c(1,3)` or `comparison = 2`).
`Kmax`	Maximum value of the willingness to pay to be considered. Default value is `k = 50000`. The willingness to pay is then approximated on a discrete grid in the interval `[0,Kmax]`. The grid is equal to `wtp` if the parameter is given, or composed of `501` elements if `wtp = NULL` (the default).
`wtp`	A(n optional) vector including the values of the willingness to pay grid. If not specified then BCEA will construct a grid of 501 values from 0 to `Kmax`. This option is useful when performing intensive computations (e.g. for the EVPPI).
`plot`	A logical value indicating whether the function should produce the summary plot or not.
`...`	Additional arguments

Value

An object of the class "bcea" containing the following elements

`n_sim`	Number of simulations produced by the Bayesian model
`n.comparators`	Number of interventions being analysed
`n.comparisons`	Number of possible pairwise comparisons
`delta.e`	For each possible comparison, the differential in the effectiveness measure
`delta.c`	For each possible comparison, the differential in the cost measure
`ICER`	The value of the Incremental Cost-Effectiveness Ratio
`Kmax`	The maximum value assumed for the willingness to pay threshold
`k`	The vector of values for the grid approximation of the willingness to pay
`ceac`	The value for the Cost-Effectiveness Acceptability Curve, as a function of the willingness to pay
`ib`	The distribution of the Incremental Benefit, for a given willingness to pay
`eib`	The value for the Expected Incremental Benefit, as a function of the willingness to pay

`kstar`	The grid approximation of the break-even point(s)
`best`	A vector containing the numeric label of the intervention that is the most cost-effective for each value of the willingness to pay in the selected grid approximation
`U`	An array including the value of the expected utility for each simulation from the Bayesian model, for each value of the grid approximation of the willingness to pay and for each intervention being considered
`vi`	An array including the value of information for each simulation from the Bayesian model and for each value of the grid approximation of the willingness to pay
`Ustar`	An array including the maximum "known-distribution" utility for each simulation from the Bayesian model and for each value of the grid approximation of the willingness to pay
`ol`	An array including the opportunity loss for each simulation from the Bayesian model and for each value of the grid approximation of the willingness to pay
`evi`	The vector of values for the Expected Value of Information, as a function of the willingness to pay
`interventions`	A vector of labels for all the interventions considered
`ref`	The numeric index associated with the intervention used as reference in the analysis
`comp`	The numeric index(es) associated with the intervention(s) used as comparator(s) in the analysis
`step`	The step size used to form the grid approximation to the willingness to pay
`e`	The `eff` matrix used to generate the object (see Arguments)
`c`	The `cost` matrix used to generate the object (see Arguments)

Author(s)

Gianluca Baio, Andrea Berardi, Nathan Green

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research. doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

Examples

```
# See Baio G., Dawid A.P. (2011) for a detailed description of the
# Bayesian model and economic problem

# Load the processed results of the MCMC simulation model
data(Vaccine)

# Runs the health economic evaluation using BCEA
m <- bcea(
      e=e,
```

```
      c=c,                  # defines the variables of
                            #  effectiveness and cost
      ref=2,                # selects the 2nd row of (e, c)
                            #  as containing the reference intervention
      interventions=treats, # defines the labels to be associated
                            #  with each intervention
      Kmax=50000,           # maximum value possible for the willingness
                            #  to pay threshold; implies that k is chosen
                            #  in a grid from the interval (0, Kmax)
      plot=TRUE             # plots the results
)

# Creates a summary table
summary(
      m,         # uses the results of the economic evaluation
                 #  (a "bcea" object)
      wtp=25000 # selects the particular value for k
)



# Plots the cost-effectiveness plane using base graphics
ceplane.plot(
      m,             # plots the Cost-Effectiveness plane
      comparison=1,  # if more than 2 interventions, selects the
                     #  pairwise comparison
      wtp=25000,     # selects the relevant willingness to pay
                     #  (default: 25,000)
      graph="base"   # selects base graphics (default)
)

# Plots the cost-effectiveness plane using ggplot2
if (requireNamespace("ggplot2")) {
ceplane.plot(
      m,             # plots the Cost-Effectiveness plane
      comparison=1,  # if more than 2 interventions, selects the
                     #  pairwise comparison
      wtp=25000,     # selects the relevant willingness to pay
                     #  (default: 25,000)
      graph="ggplot2"# selects ggplot2 as the graphical engine
)

# Some more options
ceplane.plot(
      m,
      graph="ggplot2",
      pos="top",
      size=5,
      ICER_size=1.5,
      label.pos=FALSE,
      opt.theme=ggplot2::theme(text=ggplot2::element_text(size=8))
)
}
```

```
# Plots the contour and scatterplot of the bivariate
# distribution of (Delta_e,Delta_c)
contour(
      m,            # uses the results of the economic evaluation
                    #  (a "bcea" object)
      comparison=1, # if more than 2 interventions, selects the
                    #  pairwise comparison
      nlevels=4,    # selects the number of levels to be
                    #  plotted (default=4)
      levels=NULL,  # specifies the actual levels to be plotted
                    #  (default=NULL, so that R will decide)
      scale=0.5,    # scales the bandwidths for both x- and
                    #  y-axis (default=0.5)
      graph="base"  # uses base graphics to produce the plot
)

# Plots the contour and scatterplot of the bivariate
#   distribution of (Delta_e,Delta_c)
contour2(
      m,          # uses the results of the economic evaluation
                  #  (a "bcea" object)
      wtp=25000,  # selects the willingness-to-pay threshold
      xlim=NULL,    # assumes default values
      ylim=NULL     # assumes default values
)

# Using ggplot2
if (requireNamespace("ggplot2")) {
contour2(
      m,              # uses the results of the economic evaluation
                      #  (a "bcea" object)
      graph="ggplot2",# selects the graphical engine
      wtp=25000,      # selects the willingness-to-pay threshold
      xlim=NULL,      # assumes default values
      ylim=NULL,      # assumes default values
      label.pos=FALSE # alternative position for the wtp label
)
}

# Plots the Expected Incremental Benefit for the "bcea" object m
eib.plot(m)

# Plots the distribution of the Incremental Benefit
ib.plot(
    m,            # uses the results of the economic evaluation
                  #  (a "bcea" object)
    comparison=1, # if more than 2 interventions, selects the
                  #  pairwise comparison
    wtp=25000,    # selects the relevant willingness
                  #  to pay (default: 25,000)
    graph="base"  # uses base graphics
)
```

```
# Produces a plot of the CEAC against a grid of values for the
# willingness to pay threshold
ceac.plot(m)

# Plots the Expected Value of Information for the "bcea" object m
evi.plot(m)
```

`best_interv_given_k` *Optimal intervention*

Description

Select the best option for each value of willingness to pay.

Usage

```
best_interv_given_k(eib, ref, comp)
```

Arguments

`eib`	Expected incremental benefit
`ref`	Reference group number
`comp`	Comparison group number(s)

Value

Group index

`ceac.plot.bcea` *Cost-Effectiveness Acceptability Curve (CEAC) Plot*

Description

Produces a plot of the Cost-Effectiveness Acceptability Curve (CEAC) against the willingness to pay threshold.

Usage

```
## S3 method for class 'bcea'
ceac.plot(
  he,
  comparison = NULL,
  pos = c(1, 0),
  graph = c("base", "ggplot2", "plotly"),
  ...
)

ceac.plot(he, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`comparison`	Selects the comparator, in case of more than two interventions being analysed. Default as NULL plots all the comparisons together. Any subset of the possible comparisons can be selected (e.g., `comparison=c(1,3)` or `comparison=2`).
`pos`	Parameter to set the position of the legend (only relevant for multiple interventions, ie more than 2 interventions being compared). Can be given in form of a string `(bottom|top)(right|left)` for base graphics and `bottom|top|left|right` for ggplot2. It can be a two-elements vector, which specifies the relative position on the x and y axis respectively, or alternatively it can be in form of a logical variable, with `FALSE` indicating to use the default position and `TRUE` to place it on the bottom of the plot.
`graph`	A string used to select the graphical engine to use for plotting. Should (partial-)match the three options `"base"`, `"ggplot2"` or `"plotly"`. Default value is `"base"`.
`...`	If `graph = "ggplot2"` and a named theme object is supplied, it will be added to the ggplot object. Additional arguments: • `line_colors`: specifies the line colour(s) - all graph types. • `line_types`: specifies the line type(s) as lty numeric values - all graph types. • `area_include`: logical, include area under the CEAC curves - plotly only. • `area_color`: specifies the AUC colour - plotly only.

Details

The CEAC estimates the probability of cost-effectiveness, with respect to a given willingness to pay threshold. The CEAC is used mainly to evaluate the uncertainty associated with the decision-making process, since it enables the quantification of the preference of the compared interventions, defined in terms of difference in utilities. Formally, the CEAC is defined as:

$$\mathrm{CEAC} = P(\mathrm{IB}(\theta) > 0)$$

If the net benefit function is used as utility function, the definition can be re-written as

$$\text{CEAC} = P(k \cdot \Delta_e - \Delta_c > 0)$$

effectively depending on the willingness to pay value k.

Value

`ceac`	If `graph = "ggplot2"` a ggplot object, or if `graph = "plotly"` a plotly object containing the requested plot. Nothing is returned when `graph = "base"`, the default.

The function produces a plot of the cost-effectiveness acceptability curve against the discrete grid of possible values for the willingness to pay parameter. Values of the CEAC closer to 1 indicate that uncertainty in the cost-effectiveness of the reference intervention is very low. Similarly, values of the CEAC closer to 0 indicate that uncertainty in the cost-effectiveness of the comparator is very low.

Author(s)

Gianluca Baio, Andrea Berardi

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`, `plot.bcea`

Examples

```
data("Vaccine")
he <- BCEA::bcea(e, c)
ceac.plot(he)

ceac.plot(he, graph = "base")
ceac.plot(he, graph = "ggplot2")
ceac.plot(he, graph = "plotly")

ceac.plot(he, graph = "ggplot2",
          title = "my title",
          line = list(colors = "green"),
          theme = ggplot2::theme_dark())

## more interventions
he2 <- BCEA::bcea(cbind(e, e - 0.0002), cbind(c, c + 5))
mypalette <- RColorBrewer::brewer.pal(3, "Accent")
ceac.plot(he2, graph = "ggplot2",
          title = "my title",
```

```
          theme = ggplot2::theme_dark(),
          pos = TRUE,
          line = list(colors = mypalette))
ceac.plot(he, graph = "base", title = "my title", line = list(colors = "green"))
ceac.plot(he2, graph = "base")

ceac.plot(he2, graph = "plotly", pos = "bottom")
```

`ceac_matplot` *CEAC Matrix Plot*

Description

CEAC plot using ‘matplot‘ in Base R.

Usage

```
ceac_matplot(he, pos_legend, graph_params, ceac)
```

Arguments

`he`	A bcea object containing the results of the Bayesian modelling and the economic evaluation.
`pos_legend`	Legend position
`graph_params`	Aesthetic ggplot parameters
`ceac`	‘ceac‘ index in ‘he‘

`ceac_plot_graph` *Cost-Effectiveness Acceptability Curve (CEAC) Plot By Graph Device*

Description

Choice of base R, ggplot2 or plotly.

Usage

```
ceac_plot_base(he, pos_legend, graph_params, ...)

## S3 method for class 'pairwise'
ceac_plot_base(he, pos_legend, graph_params, ...)

## S3 method for class 'bcea'
ceac_plot_base(he, pos_legend, graph_params, ...)
```

```
ceac_plot_ggplot(he, pos_legend, graph_params, ...)

## S3 method for class 'pairwise'
ceac_plot_ggplot(he, pos_legend, graph_params, ...)

## S3 method for class 'bcea'
ceac_plot_ggplot(he, pos_legend, graph_params, ...)

ceac_ggplot(he, pos_legend, graph_params, ceac, ...)

ceac_plot_plotly(he, pos_legend = "left", graph_params)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`pos_legend`	Legend position
`graph_params`	Aesthetic ggplot parameters
`...`	Additional arguments
`ceac`	ceac index in he

`ceaf.plot.pairwise` *Cost-Effectiveness Acceptability Frontier (CEAF) plot*

Description

Produces a plot the Cost-Effectiveness Acceptability Frontier (CEAF) against the willingness to pay threshold.

Usage

```
## S3 method for class 'pairwise'
ceaf.plot(mce, graph = c("base", "ggplot2"), ...)

ceaf.plot(mce, ...)
```

Arguments

`mce`	The output of the call to the function `multi.ce`
`graph`	A string used to select the graphical engine to use for plotting. Should (partial-) match the two options `"base"` or `"ggplot2"`. Default value is `"base"`.
`...`	Additional arguments

Value

`ceaf`	A ggplot object containing the plot. Returned only if `graph="ggplot2"`.

Author(s)

Gianluca Baio, Andrea Berardi

References

Baio G, Dawid AP. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`, `multi.ce`

Examples

```
# See Baio G., Dawid A.P. (2011) for a detailed description of the
# Bayesian model and economic problem

# Load the processed results of the MCMC simulation model
data(Vaccine)

# Runs the health economic evaluation using BCEA
m <- bcea(
      e=e,
      c=c,                  # defines the variables of
                            #  effectiveness and cost
      ref=2,                # selects the 2nd row of (e, c)
                            #  as containing the reference intervention
      interventions=treats, # defines the labels to be associated
                            #  with each intervention
      Kmax=50000,           # maximum value possible for the willingness
                            #  to pay threshold; implies that k is chosen
                            #  in a grid from the interval (0, Kmax)
      plot=FALSE            # inhibits graphical output
)


mce <- multi.ce(m)          # uses the results of the economic analysis



ceaf.plot(mce)              # plots the CEAF



ceaf.plot(mce, graph = "g") # uses ggplot2



# Use the smoking cessation dataset
```

```
data(Smoking)
m <- bcea(e, c, ref = 4, intervention = treats, Kmax = 500, plot = FALSE)
mce <- multi.ce(m)
ceaf.plot(mce)
```

`ceef.plot.bcea` *Cost-Effectiveness Efficiency Frontier (CEAF) Plot*

Description

The line connecting successive points on a cost-effectiveness plane which each represent the effect and cost associated with different treatment alternatives. The gradient of a line segment represents the ICER of the treatment comparison between the two alternatives represented by that segment. The cost-effectiveness frontier consists of the set of points corresponding to treatment alternatives that are considered to be cost-effective at different values of the cost-effectiveness threshold. The steeper the gradient between successive points on the frontier, the higher is the ICER between these treatment alternatives and the more expensive alternative would be considered cost-effective only when a high value of the cost-effectiveness threshold is assumed. Points not lying on the cost-effectiveness frontier represent treatment alternatives that are not considered cost-effective at any value of the cost-effectiveness threshold.

Usage

```
## S3 method for class 'bcea'
ceef.plot(
  he,
  comparators = NULL,
  pos = c(1, 1),
  start.from.origins = TRUE,
  threshold = NULL,
  flip = FALSE,
  dominance = TRUE,
  relative = FALSE,
  print.summary = TRUE,
  graph = c("base", "ggplot2"),
  print.plot = TRUE,
  ...
)

ceef.plot(he, ...)
```

Arguments

he	A bcea object containing the results of the Bayesian modelling and the economic evaluation.

`comparators`	Vector specifying the comparators to be included in the frontier analysis. It must have a length > 1. Default as `NULL` includes all the available comparators.
`pos`	Parameter to set the position of the legend (only relevant for multiple interventions, ie more than 2 interventions being compared). Can be given in form of a string `(bottom|top)(right|left)` for base graphics and `bottom|top|left|right` for ggplot2. It can be a two-elements vector, which specifies the relative position on the x and y axis respectively, or alternatively it can be in form of a logical variable, with `FALSE` indicating to use the default position and `TRUE` to place it on the bottom of the plot.
`start.from.origins`	Logical. Should the frontier start from the origins of the axes? The argument is reset to `FALSE` if the average effectiveness and/or costs of at least one comparator are negative.
`threshold`	Specifies if the efficiency should be defined based on a willingness-to-pay threshold value. If set to `NULL` (the default), no conditions are included on the slope increase. If a positive value is passed as argument, to be efficient an intervention also requires to have an ICER for the comparison versus the last efficient strategy not greater than the specified threshold value. A negative value will be ignored with a warning.
`flip`	Logical. Should the axes of the plane be inverted?
`dominance`	Logical. Should the dominance regions be included in the plot?
`relative`	Logical. Should the plot display the absolute measures (the default as `FALSE`) or the differential outcomes versus the reference comparator?
`print.summary`	Logical. Should the efficiency frontier summary be printed along with the graph? See Details for additional information.
`graph`	A string used to select the graphical engine to use for plotting. Should (partial-)match the two options `"base"` or `"ggplot2"`. Default value is `"base"`.
`print.plot`	Logical. Should the efficiency frontier be plotted?
`...`	If `graph_type="ggplot2"` and a named theme object is supplied, it will be added to the ggplot object. Ignored if `graph_type="base"`. Setting the optional argument `include.ICER` to `TRUE` will print the ICERs in the summary tables, if produced.

Details

Back compatibility with BCEA previous versions: The `bcea` objects did not include the generating `e` and `c` matrices in BCEA versions <2.1-0. This function is not compatible with objects created with previous versions. The matrices can be appended to `bcea` objects obtained using previous versions, making sure that the class of the object remains unaltered.

The argument `print.summary` allows for printing a brief summary of the efficiency frontier, with default to `TRUE`. Two tables are plotted, one for the interventions included in the frontier and one for the dominated interventions. The average costs and clinical benefits are included for each intervention. The frontier table includes the slope for the increase in the frontier and the non-frontier table displays the dominance type of each dominated intervention. Please note that the slopes are defined as the increment in the costs for a unit increment in the benefits even if `flip = TRUE` for consistency with the ICER definition. The angle of increase is in radians and depends on the definition of the axes, i.e. on the value given to the `flip` argument.

If the argument `relative` is set to `TRUE`, the graph will not display the absolute measures of costs and benefits. Instead the axes will represent differential costs and benefits compared to the reference intervention (indexed by `ref` in the `bcea` function).

Value

`ceplane`	A ggplot object containing the plot. Returned only if `graph_type="ggplot2"`.

The function produces a plot of the cost-effectiveness efficiency frontier. The dots show the simulated values for the intervention-specific distributions of the effectiveness and costs. The circles indicate the average of each bivariate distribution, with the numbers referring to each included intervention. The numbers inside the circles are black if the intervention is included in the frontier and grey otherwise. If the option `dominance` is set to `TRUE`, the dominance regions are plotted, indicating the areas of dominance. Interventions in the areas between the dominance region and the frontier are in a situation of extended dominance.

Author(s)

Andrea Berardi, Gianluca Baio

References

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

IQWIG (2009). General methods for the Assessment of the Relation of Benefits to Cost, Version 1.0. IQWIG, November 2009.

See Also

`bcea`

Examples

```
## create the bcea object m for the smoking cessation example
data(Smoking, package = "BCEA")
m <- bcea(e, c, ref = 4, Kmax = 500, interventions = treats)

## produce plot
ceef.plot(m, graph = "base")


## tweak the options
## flip axis
ceef.plot(m,
          flip = TRUE,
          dominance = FALSE,
          start.from.origins = FALSE,
          print.summary = FALSE,
          graph = "base")
```

```
## or use ggplot2 instead
if(require(ggplot2)){
ceef.plot(m,
          dominance = TRUE,
          start.from.origins = FALSE,
          pos = TRUE,
          print.summary = FALSE,
          graph = "ggplot2")
 }
```

`ceef.summary` *Summary table for CEEF*

Description

Summary table for CEEF

Usage

```
ceef.summary(he, frontier_data, frontier_params, include.ICER = FALSE, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`frontier_data`	Frontier data
`frontier_params`	Frontier parameters
`include.ICER`	Should we include the ICER? default: FALSE
`...`	Additional arguments

Value

Summary printed to console

`ceef_plot_graph` *Cost-effectiveness Efficiency Frontier Plot By Graph Device*

Description

Choice of base R, ggplot2.

Usage

```
ceef_plot_ggplot(he, frontier_data, frontier_params, ...)

ceef_plot_base(he, frontier_data, frontier_params)
```

Arguments

`he`	A bcea object containing the results of the Bayesian modelling and the economic evaluation.
`frontier_data`	Frontier data
`frontier_params`	Frontier parameters
`...`	Additional arguments

`ceplane.plot.bcea` *Cost-effectiveness Plane Plot*

Description

Produces a scatter plot of the cost-effectiveness plane, together with the sustainability area, as a function of the selected willingness to pay threshold.

Usage

```
## S3 method for class 'bcea'
ceplane.plot(
  he,
  comparison = NULL,
  wtp = 25000,
  pos = c(0, 1),
  graph = c("base", "ggplot2", "plotly"),
  ...
)

ceplane.plot(he, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`comparison`	Selects the comparator, in case of more than two interventions being analysed. Default as `NULL` plots all the comparisons together. Any subset of the possible comparisons can be selected (e.g., `comparison = c(1,3)` or `comparison = 2`).
`wtp`	The value of the willingness to pay parameter. Not used if `graph = "base"` for multiple comparisons.
`pos`	Parameter to set the position of the legend; for a single comparison plot, the ICER legend position. Can be given in form of a string `(bottom|top)(right|left)` for base graphics and `bottom|top|left|right` for ggplot2. It can be a two-elements vector, which specifies the relative position on the x and y axis respectively, or alternatively it can be in form of a logical variable, with `FALSE` indicating to use the default position and `TRUE` to place it on the bottom of the plot. Default value is `c(1,1)`, that is the topright corner inside the plot area.
`graph`	A string used to select the graphical engine to use for plotting. Should (partial-) match the two options `"base"` or `"ggplot2"`. Default value is `"base"`.
`...`	Additional arguments

Details

In the plotly version, point_colors, ICER_colors and area_color can also be specified as rgba colours using either the `toRGB` function or a rgba colour string, e.g. `'rgba(1,1,1,1)'`.

Value

If `graph = "ggplot2"` a ggplot object, or if `graph = "plotly"` a plotly object containing the requested plot. Nothing is returned when `graph = "base"`, the default.

Grey dots show the simulated values for the joint distribution of the effectiveness and cost differentials. The larger red dot shows the ICER and the grey area identifies the sustainability area, i.e. the part of the plan for which the simulated values are below the willingness to pay threshold. The proportion of points in the sustainability area effectively represents the CEAC for a given value of the willingness to pay. If the comparators are more than 2 and no pairwise comparison is specified, all scatterplots are graphed using different colours.

Author(s)

Gianluca Baio, Andrea Berardi

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research. doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`, `ceplane_plot_graph`

Examples

```
## create the bcea object for the smoking cessation example
data(Smoking)

m <- bcea(e, c, ref = 4, Kmax = 500, interventions = treats)

## produce the base plot
ceplane.plot(m, wtp = 200, graph = "base")

## select only one comparator
ceplane.plot(m, wtp = 200, graph = "base", comparison = 3)

## use ggplot2
if (requireNamespace("ggplot2")) {
   ceplane.plot(m, wtp = 200, pos = "right", ICER_size = 2, graph = "ggplot2")
}

## plotly
ceplane.plot(m, wtp = 200, graph = "plotly")
ceplane.plot(m, wtp = 200, comparison = 1, graph = "plotly")
```

`ceplane_geom_params` *Extract Separate Parameter Sets*

Description

Extract Separate Parameter Sets

Usage

```
ceplane_geom_params(...)
```

Arguments

`...`	Additional arguments

`ceplane_ggplot_params` *CE-plane ggplot Parameters*

Description

CE-plane ggplot Parameters

Usage

```
ceplane_ggplot_params(he, wtp, pos_legend, graph_params, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`wtp`	Willingness to pay
`pos_legend`	Position of legend
`graph_params`	Other graphical parameters
`...`	Additional arguments

`ceplane_plot_graph` *Cost-Effectiveness Plane Plot By Graph Device*

Description

Choice of base R, ggplot2 or plotly.

Usage

```
## S3 method for class 'bcea'
ceplane_plot_base(he, wtp = 25000, pos_legend, graph_params, ...)

ceplane_plot_base(he, ...)

## S3 method for class 'bcea'
ceplane_plot_ggplot(he, wtp = 25000, pos_legend, graph_params, ...)

ceplane_plot_ggplot(he, ...)

## S3 method for class 'bcea'
ceplane_plot_plotly(he, wtp = 25000, pos_legend, graph_params, ...)

ceplane_plot_plotly(he, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`wtp`	Willingness to pay threshold; default 25,000
`pos_legend`	Legend position
`graph_params`	Graph parameters in ggplot format
`...`	Additional arguments

Value

For base R returns a plot

For ggplot2 returns ggplot2 object

For plotly returns a plot in the Viewer

Examples

```
# single comparator
data(Vaccine, package = "BCEA")

he <- bcea(e, c)
ceplane.plot(he, graph = "base")

## Not run:
# need to provide all the defaults because thats what
# ceplane.plot() does

graph_params <-  list(xlab = "x-axis label",
                      ylab = "y-axis label",
                      title = "my title",
                      xlim = c(-0.002, 0.001),
                      ylim = c(-13, 5),
                      point = list(sizes = 1,
                                   colors = "darkgrey"),
                      area = list(color = "lightgrey"))

he$delta_e <- as.matrix(he$delta_e)
he$delta_c <- as.matrix(he$delta_c)

BCEA::ceplane_plot_base(he, graph_params = graph_params)

## single non-default comparator


## multiple comparators
data(Smoking, package = "BCEA")

graph_params <-  list(xlab = "x-axis label",
                      ylab = "y-axis label",
                      title = "my title",
                      xlim = c(-1, 2.5),
                      ylim = c(-1, 160),
                      point = list(sizes = 0.5,
                                   colors = grey.colors(3, start = 0.1, end = 0.7)),
                      area = list(color = "lightgrey"))

he <- bcea(e, c, ref = 4, Kmax = 500, interventions = treats)

BCEA::ceplane_plot_base(he,
                       wtp = 200,
                       pos_legend = FALSE,
                       graph_params = graph_params)

## End(Not run)


data(Vaccine)
he <- bcea(e, c)
```

```
ceplane.plot(he, graph = "ggplot2")

data(Smoking)
he <- bcea(e, c, ref = 4, Kmax = 500, interventions = treats)

ceplane.plot(he, graph = "ggplot2")

ceplane.plot(he,
             wtp = 200,
             pos = "right",
             ICER_size = 2,
             graph = "ggplot2")

ceplane.plot(he,
             wtp = 200,
             pos = TRUE,
             graph = "ggplot2")

ceplane.plot(he,
             graph = "ggplot2",
             wtp=200,
             theme = ggplot2::theme_linedraw())
```

`CEriskav_assign` *Cost-effectiveness Analysis Including a Parameter of Risk Aversion*

Description

Extends the standard cost-effectiveness analysis to modify the utility function so that risk aversion of the decision maker is explicitly accounted for.

Default vector of risk aversion parameters: 1e-11, 2.5e-6, 5e-6

Usage

```
CEriskav(he) <- value

## S3 replacement method for class 'bcea'
CEriskav(he) <- value

## Default S3 replacement method:
CEriskav(he) <- value
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.

`value`	A vector of values for the risk aversion parameter. If `NULL`, default values are assigned by R. The first (smallest) value (`r` -> 0) produces the standard analysis with no risk aversion.

Value

An object of the class `CEriskav` containing the following elements:

`Ur`	An array containing the simulated values for all the ”known-distribution” utilities for all interventions, all the values of the willingness to pay parameter and for all the possible values of `r`
`Urstar`	An array containing the simulated values for the maximum ”known-distribution” expected utility for all the values of the willingness to pay parameter and for all the possible values of `r`
`IBr`	An array containing the simulated values for the distribution of the Incremental Benefit for all the values of the willingness to pay and for all the possible values of `r`
`eibr`	An array containing the Expected Incremental Benefit for each value of the willingness to pay parameter and for all the possible values of `r`
`vir`	An array containing all the simulations for the Value of Information for each value of the willingness to pay parameter and for all the possible values of `r`
`evir`	An array containing the Expected Value of Information for each value of the willingness to pay parameter and for all the possible values of `r`
`R`	The number of possible values for the parameter of risk aversion `r`
`r`	The vector containing all the possible values for the parameter of risk aversion `r`

Author(s)

Gianluca Baio

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

bcea

Examples

```
# See Baio G., Dawid A.P. (2011) for a detailed description of the
# Bayesian model and economic problem

# Load the processed results of the MCMC simulation model
```

```
data(Vaccine)

# Runs the health economic evaluation using BCEA
m <- bcea(e=e,c=c,          # defines the variables of
                            #  effectiveness and cost
      ref=2,                # selects the 2nd row of (e, c)
                            #  as containing the reference intervention
      interventions=treats, # defines the labels to be associated
                            #  with each intervention
      Kmax=50000            # maximum value possible for the willingness
                            #  to pay threshold; implies that k is chosen
                            #  in a grid from the interval (0, Kmax)
)

# Define the vector of values for the risk aversion parameter, r, eg:
r <- c(1e-10, 0.005, 0.020, 0.035)

# Run the cost-effectiveness analysis accounting for risk aversion

# uses the results of the economic evaluation
# if more than 2 interventions, selects the
#  pairwise comparison

CEriskav(m) <- r
```

`CEriskav_plot_graph` *Cost-effectiveness Plot Including a Parameter of Risk Aversion Choice of base R, ggplot2.*

Description

Cost-effectiveness Plot Including a Parameter of Risk Aversion

Choice of base R, ggplot2.

CEriskav base R version

CEriskav ggplot2 version

Usage

```
CEriskav_plot_base(he, pos_legend)

CEriskav_plot_ggplot(he, pos_legend)
```

Arguments

`he`	A bcea object containing the results of the Bayesian modelling and the economic evaluation.
`pos_legend`	Legend position

`compute.evppi` *Compute EVPPI*

Description

Compute EVPPI

Usage

```
## S3 method for class 'evppi'
compute(he, fit.full)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`fit.full`	fit.full

Value

list

See Also

`evppi`

`compute_CEAC` *Compute Cost-Effectiveness Acceptability Curve*

Description

Compute Cost-Effectiveness Acceptability Curve

Usage

```
compute_CEAC(ib)
```

Arguments

`ib`	Incremental benefit

Value

Array with dimensions (interv x k)

See Also

`ceac.plot`

`compute_ceaf` *Compute Cost-Effectiveness Acceptability Frontier*

Description

Compute Cost-Effectiveness Acceptability Frontier

Usage

```
compute_ceaf(p_best_interv)
```

Arguments

`p_best_interv`	Probability of being best intervention

`compute_EIB` *Compute Expected Incremental Benefit*

Description

A summary measure useful to assess the potential changes in the decision under different scenarios.

Usage

```
compute_EIB(ib)
```

Arguments

`ib`	Incremental benefit

Details

When considering a pairwise comparison (e.g. in the simple case of a reference intervention $t = 1$ and a comparator, such as the status quo, $t = 0$), it is defined as the difference between the expected utilities of the two alternatives:

$$eib := \mathrm{E}[u(e,c;1)] - \mathrm{E}[u(e,c;0)] = \mathcal{U}^1 - \mathcal{U}^0.$$

Analysis of the expected incremental benefit describes how the decision changes for different values of the threshold. The EIB marginalises out the uncertainty, and does not incorporate and describe explicitly the uncertainty in the outcomes. To overcome this problem the tool of choice is the CEAC.

Value

Array with dimensions (interv x k)

See Also

`ceac.plot`, `compute_CEAC`, `compute_IB`

`compute_eib_cri` *Calculate Credible Intervals*

Description

For expected incremental benefit plot.

Usage

```
compute_eib_cri(he, alpha_cri = 0.05, cri.quantile = TRUE)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`alpha_cri`	Significance level, 0 - 1
`cri.quantile`	Credible interval quantile?; logical

Value

cri

`compute_EVI` *Compute Expected Value of Information*

Description

Compute Expected Value of Information

Usage

```
compute_EVI(ol)
```

Arguments

`ol`	Opportunity loss

Value

EVI

`compute_IB` *Compute Incremental Benefit*

Description

Sample of incremental net monetary benefit for each willingness-to-pay threshold, k, and comparator.

Usage

```
compute_IB(df_ce, k)
```

Arguments

`df_ce`	Dataframe of cost and effectiveness deltas
`k`	Vector of willingness to pay values

Details

Defined as:

$$IB = u(e, c; 1) - u(e, c; 0).$$

If the net benefit function is used as utility function, the definition can be re-written as

$$IB = k \cdot \Delta_e - \Delta_c.$$

Value

Array with dimensions (k x sim x ints)

See Also

`compute_EIB`

`compute_ICER` *Compute Incremental Cost-Effectiveness Ratio*

Description

Defined as

Usage

```
compute_ICER(df_ce)
```

Arguments

`df_ce`	Cost-effectiveness dataframe

Details

$$ICER = \Delta_c/\Delta_e$$

Value

ICER for all comparisons

`compute_kstar` *Compute k^**

Description

Find willingness-to-pay threshold when optimal decision changes.

Usage

```
compute_kstar(k, best, ref)
```

Arguments

`k`	Willingness-to-pay grid approximation of the budget willing to invest (vector)
`best`	Best intervention for each ‘k‘ (int)
`ref`	Reference intervention (int)

Details

$$k^* := \min\{k : IB < 0\}$$

The value of the break-even point corresponds to the ICER and quantifies the point at which the decision-maker is indifferent between the two options.

Value

integer representing intervention

See Also

`ceac.plot`

`compute_ol` *Compute Opportunity Loss*

Description

The difference between the maximum utility computed for the current parameter configuration (e.g. at the current simulation) U^* and the current utility of the intervention associated with the maximum utility overall.

Usage

```
compute_ol(Ustar, U, best)
```

Arguments

`Ustar`	Maximum utility value (sim x k)
`U`	Net monetary benefit (sim x k x interv)
`best`	Best intervention for given willingness-to-pay (k)

Details

In mathematical notation,

$$\text{OL}(\theta) := U^*(\theta) - U(\theta^\tau)$$

where τ is the intervention associated with the overall maximum utility and $U^*(\theta)$ is the maximum utility value among the comparators in the given simulation. The opportunity loss is a non-negative quantity, since $U(\theta^\tau) \leq U^*(\theta)$.

In all simulations where the intervention is more cost-effective (i.e. when incremental benefit is positive), then $\text{OL}(\theta) = 0$ as there would be no opportunity loss, if the parameter configuration were the one obtained in the current simulation.

Value

Array with dimensions (sim x k)

See Also

`compute_vi`

`compute_p_best_interv` *Compute Probability Best Intervention*

Description

Compute Probability Best Intervention

Usage

```
compute_p_best_interv(he)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.

`compute_U` *Compute U Statistic*

Description

Sample of net (monetary) benefit for each willingness-to-pay threshold and intervention.

Usage

```
compute_U(df_ce, k)
```

Arguments

`df_ce`	Cost-effectiveness dataframe
`k`	Willingness to pay vector

Value

Array with dimensions (sim x k x ints)

`compute_Ubar` *Compute NB for mixture of interventions*

Description

Compute NB for mixture of interventions

Usage

```
compute_Ubar(he, value)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`value`	Mixture weights

`compute_Ustar` *Compute Ustar Statistic*

Description

The maximum utility value among the comparators, indicating which intervention produced the most benefits at each simulation.

Usage

```
compute_Ustar(U)
```

Arguments

`U`	Net monetary benefit (sim x k x intervs)

Value

Array with dimensions (sim x k)

`compute_vi` *Compute Value of Information*

Description

The difference between the maximum utility computed for the current parameter configuration U^* and the utility of the intervention which is associated with the maximum utility overall.

Usage

```
compute_vi(Ustar, U)
```

Arguments

`Ustar`	Maximum utility value (sim x k)
`U`	Net monetary benefit (sim x k x interv)

Details

The value of obtaining additional information on the parameter θ to reduce the uncertainty in the decisional process. It is defined as:

$$\mathrm{VI}(\theta) := U^*(\theta) - \mathcal{U}^*$$

with $U^*(\theta)$ the maximum utility value for the given simulation among all comparators and $\mathcal{U}^*(\theta)$ the expected utility gained by the adoption of the cost-effective intervention.

Value

Array with dimensions (sim x k)

See Also

`compute_ol`

`comp_names_from_` *Comparison Names From*

Description

Comparison Names From

Usage

```
comp_names_from_(df_ce)
```

Arguments

`df_ce`	Cost-effectiveness dataframe

`contour.bcea` *Contour Plots for the Cost-Effectiveness Plane*

Description

Contour method for objects in the class bcea. Produces a scatterplot of the cost-effectiveness plane, with a contour-plot of the bivariate density of the differentials of cost (y-axis) and effectiveness (x-axis).

Usage

```
## S3 method for class 'bcea'
contour(
  he,
  scale = 0.5,
  nlevels = 4,
  levels = NULL,
  pos = c(1, 0),
  xlim = NULL,
  ylim = NULL,
  graph = c("base", "ggplot2"),
  ...
)

contour(he, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`scale`	Scales the plot as a function of the observed standard deviation.
`nlevels`	Number of levels to be plotted in the contour.
`levels`	Numeric vector of levels at which to draw contour lines. Will be ignored using `graph="ggplot2"`.
`pos`	Parameter to set the position of the legend (only relevant for multiple interventions, ie more than 2 interventions being compared). Can be given in form of a string `(bottom|top)(right|left)` for base graphics and `bottom|top|left|right` for ggplot2. It can be a two-elements vector, which specifies the relative position on the x and y axis respectively, or alternatively it can be in form of a logical variable, with `FALSE` indicating to use the default position and `TRUE` to place it on the bottom of the plot.
`xlim`	The range of the plot along the x-axis. If NULL (default) it is determined by the range of the simulated values for `delta_e`
`ylim`	The range of the plot along the y-axis. If NULL (default) it is determined by the range of the simulated values for `delta_c`
`graph`	A string used to select the graphical engine to use for plotting. Should (partial-) match the two options `"base"` or `"ggplot2"`. Default value is `"base"`.
`...`	Additional arguments

Value

`ceplane`	A ggplot object containing the plot. Returned only if `graph="ggplot2"`.

Plots the cost-effectiveness plane with a scatterplot of all the simulated values from the (posterior) bivariate distribution of (Δ_e, Δ_c), the differentials of effectiveness and costs; superimposes a contour of the distribution and prints the estimated value of the probability of each quadrant (combination of positive/negative values for both Δ_e and Δ_c)

Author(s)

Gianluca Baio, Andrea Berardi

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`, `ceplane.plot`, `contour2`

Examples

```
data(Vaccine)

# Runs the health economic evaluation using BCEA
m <- bcea(e=e,
          c=c,              # defines the variables of
                            #  effectiveness and cost
      ref=2,                # selects the 2nd row of (e,c)
                            #  as containing the reference intervention
      interventions=treats, # defines the labels to be associated
                            #  with each intervention
      Kmax=50000,           # maximum value possible for the willingness
                            #  to pay threshold; implies that k is chosen
                            #  in a grid from the interval (0,Kmax)
      plot=TRUE             # plots the results
)

contour(m)
contour(m, graph = "ggplot2")

# Plots the contour and scatterplot of the bivariate
# distribution of (Delta_e, Delta_c)
contour(m,          # uses the results of the economic evaluation
                    #  (a "bcea" object)
      comparison=1, # if more than 2 interventions, selects the
                    #  pairwise comparison
      nlevels=4,    # selects the number of levels to be
                    #  plotted (default=4)
      levels=NULL,  # specifies the actual levels to be plotted
                    #  (default=NULL, so that R will decide)
      scale=0.5,    # scales the bandwidths for both x- and
                    #  y-axis (default=0.5)
      graph="base"  # uses base graphics to produce the plot
)
```

`contour2.bcea` *Specialised CE-plane Contour Plot*

Description

Produces a scatterplot of the cost-effectiveness plane, with a contour-plot of the bivariate density of the differentials of cost (y-axis) and effectiveness (x-axis). Also adds the sustainability area (i.e. below the selected value of the willingness-to-pay threshold).

Usage

```
## S3 method for class 'bcea'
contour2(
```

```
  he,
  wtp = 25000,
  xlim = NULL,
  ylim = NULL,
  comparison = NULL,
  graph_type = c("base", "ggplot2"),
  ...
)

contour2(he, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`wtp`	The selected value of the willingness-to-pay. Default is `25000`.
`xlim`	Limits on the x-axis (default=`NULL`, so that R will select appropriate limits).
`ylim`	Limits on the y-axis (default=`NULL`, so that R will select appropriate limits).
`comparison`	The comparison being plotted. Default to `NULL` chooses the first comparison if `graph_type="base"`. If `graph_type="ggplot2"` the default value will choose all the possible comparisons. Any subset of the possible comparisons can be selected (e.g., `comparison=c(1,3)`).
`graph_type`	A string used to select the graphical engine to use for plotting. Should (partial-)match the two options `"base"` or `"ggplot2"`. Default value is `"base"`.
`...`	Arguments to be passed to `ceplane.plot`. See the relative manual page for more details.

Value

`contour`	A ggplot item containing the requested plot. Returned only if `graph_type="ggplot2"`.

Plots the cost-effectiveness plane with a scatterplot of all the simulated values from the (posterior) bivariate distribution of (Δ_e, Δ_c), the differentials of effectiveness and costs; superimposes a contour of the distribution and prints the value of the ICER, together with the sustainability area.

Author(s)

Gianluca Baio, Andrea Berardi

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`, `ceplane.plot`, `contour`

Examples

```
## create the bcea object m for the smoking cessation example
data(Smoking)
m <- bcea(e, c, ref = 4, interventions = treats, Kmax = 500)

## produce the plot
contour2(m,
         wtp = 200,
         graph_type = "base")


## or use ggplot2 to plot multiple comparisons
contour2(m,
         wtp = 200,
         ICER_size = 2,
         graph_type = "ggplot2")
```

`contour_graph` *Contour Cost-Effectiveness Plane*

Description

Choice of base R, ggplot2

Usage

```
contour_base(
  he,
  params,
  scale,
  nlevels,
  levels,
  xlim,
  ylim,
  extra_args,
  comparison = 1
)

contour_ggplot(
  he,
  params,
  scale,
  nlevels,
  levels,
  xlim,
  ylim,
```

```
  extra_args,
  comparison = 1
)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`params`	Plot parameters
`scale`	Scale
`nlevels`	Number of levels
`levels`	levels
`xlim`	x-axis limits
`ylim`	y-axis limits
`extra_args`	additional arguments
`comparison`	Comparison interventions; default 1

`convert_pts_to_mm` *Use from Base R to ggplot*

Description

Use from Base R to ggplot

Usage

```
convert_pts_to_mm(x)
```

Arguments

`x`	points

`createInputs.default` *Create Inputs for EVPI Calculation*

Description

Creates an object containing the matrix with the parameters simulated using the MCMC procedure (using JAGS, BUGS or Stan) and a vector of parameters (strings) that can be used to perform the expected value of partial information analysis. In the process, `createInputs` also checks for linear dependency among columns of the PSA samples or columns having constant values and removes them to only leave the fundamental parameters (to run VoI analysis). This also deals with simulations stored in a `.csv` or `.txt` file (e.g. as obtained using bootstrapping from a non-Bayesian model).

Usage

```
## Default S3 method:
createInputs(inputs, print_is_linear_comb = TRUE)

createInputs(inputs, print_is_linear_comb = TRUE)

## S3 method for class 'rjags'
createInputs(inputs, print_is_linear_comb = TRUE)

## S3 method for class 'bugs'
createInputs(inputs, print_is_linear_comb = TRUE)

## S3 method for class 'stanfit'
createInputs(inputs, print_is_linear_comb = TRUE)

## S3 method for class 'data.frame'
createInputs(inputs, print_is_linear_comb = TRUE)

## S3 method for class 'numeric'
createInputs(inputs, print_is_linear_comb = TRUE)
```

Arguments

`inputs`	A `rjags`, bugs or `stanfit` object, containing the results of a call to either JAGS, (using `R2jags`), BUGS (using `R2WinBUGS`, or Stan (using `rstan`).
`print_is_linear_comb`	Logical indicator. If set to `TRUE` (default) then prints the output of the procedure trying to assess whether there are some parameters that are a linear combination of others (in which case they are removed).

Value

`mat`	Data.frame containing all the simulations for all the monitored parameters
`parameters`	Character vectors of the names of all the monitored parameters

Author(s)

Gianluca Baio, Anna Heath and Mark Strong

See Also

`bcea`, `evppi`

`diag.evppi` *Diagnostic Plots For The Results Of The EVPPI*

Description

The function produces either a residual plot comparing the fitted values from the INLA-SPDE Gaussian Process regression to the residuals. This is a scatter plot of residuals on the y axis and fitted values (estimated responses) on the x axis. The plot is used to detect non-linearity, unequal error variances, and outliers. A well-behaved residual plot supporting the appropriateness of the simple linear regression model has the following characteristics: 1) The residuals bounce randomly around the 0 line. This suggests that the assumption that the relationship is linear is reasonable. 2) The residuals roughly form a horizontal band around the 0 line. This suggests that the variances of the error terms are equal. 3) None of the residual stands out from the basic random pattern of residuals. This suggests that there are no outliers.

Usage

```
diag.evppi(evppi, he, plot_type = c("residuals", "qqplot"), interv = 1)
```

Arguments

`evppi`	A `evppi` object obtained by running the function `evppi` on a `bcea` model.
`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`plot_type`	The type of diagnostics to be performed. It can be the 'residual plot' or the 'qqplot plot'.
`interv`	Specifies the interventions for which diagnostic tests should be performed (if there are many options being compared)

Details

The second possible diagnostic is the qqplot for the fitted value. This is a graphical method for comparing the fitted values distributions with the assumed underlying normal distribution by plotting their quantiles against each other. First, the set of intervals for the quantiles is chosen. A point (x,y) on the plot corresponds to one of the quantiles of the second distribution (y-coordinate) plotted against the same quantile of the first distribution (x-coordinate). If the two distributions being compared are identical, the Q-Q plot follows the 45 degrees line.

Value

Plot

Author(s)

Gianluca Baio, Anna Heath

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`, `evppi`

`eib.plot.bcea` *Expected Incremental Benefit (EIB) Plot*

Description

Produces a plot of the Expected Incremental Benefit (EIB) as a function of the willingness to pay.

Usage

```
## S3 method for class 'bcea'
eib.plot(
  he,
  comparison = NULL,
  pos = c(1, 0),
  size = NULL,
  plot.cri = FALSE,
  graph = c("base", "ggplot2", "plotly"),
  ...
)

eib.plot(he, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`comparison`	Selects the comparator, in case of more than two interventions being analysed. Default as NULL plots all the comparisons together. Any subset of the possible comparisons can be selected (e.g., `comparison=c(1,3)` or `comparison=2`).

`pos`	Parameter to set the position of the legend (only relevant for multiple interventions, ie more than 2 interventions being compared). Can be given in form of a string `(bottom|top)(right|left)` for base graphics and `bottom|top|left|right` for ggplot2. It can be a two-elements vector, which specifies the relative position on the x and y axis respectively, or alternatively it can be in form of a logical variable, with `FALSE` indicating to use the default position and `TRUE` to place it on the bottom of the plot.
`size`	Value (in millimetres) of the size of the willingness to pay label. Used only if `graph="ggplot2"`, otherwise it will be ignored with a message. If set to `NA`, the break-even point line(s) and label(s) are suppressed, with both base graphics and ggplot2.
`plot.cri`	Logical value. Should the credible intervals be plotted along with the expected incremental benefit? Default as `NULL` draws the 95% credible intervals if only one comparison is selected, and does not include them for multiple comparisons. Setting `plot.cri=TRUE` or `plot.cri=FALSE` forces the function to add the intervals or not. The level of the intervals can be also set, see ... for more details.
`graph`	A string used to select the graphical engine to use for plotting. Should (partial-)match the three options `"base"`, `"ggplot2"` or `"plotly"`. Default value is `"base"`.
`...`	If `graph="ggplot2"` and a named theme object is supplied, it will be added to the ggplot object. Additional arguments: • `alpha` can be used to set the CrI level when `plot.cri=TRUE`, with a default value of `alpha=0.05`. • `cri.quantile` controls the the method of calculation of the credible intervals. The default value `cri.quantile=TRUE` defines the CrI as the interval between the `alpha/2`-th and `1-alpha/2`-th quantiles of the IB distribution. Setting `cri.quantile=FALSE` will use a normal approximation on the IB distribution to calculate the intervals. • `line_colors`: specifies the line colour(s) - all graph types. • `line_types`: specifies the line type(s) as lty numeric values - all graph types. • `area_include`: include area under the EIB curve - plotly only. • `area_color`: specifies the AUC curve - plotly only.

Value

`eib`	If `graph="ggplot2"` a ggplot object, or if `graph="plotly"` a plotly object containing the requested plot. Nothing is returned when `graph="base"`, the default.

The function produces a plot of the Expected Incremental Benefit as a function of the discrete grid approximation of the willingness to pay parameter. The break even point (i.e. the point in which the EIB = 0, i.e. when the optimal decision changes from one intervention to another) is also showed by default. The value 'k*' is the discrete grid approximation of the ICER.

Author(s)

Gianluca Baio, Andrea Berardi

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`, `ib.plot`, `ceplane.plot`

Examples

```
data(Vaccine)

# Runs the health economic evaluation using BCEA
m <- bcea(
      e=e,
      c=c,                  # defines the variables of
                            #  effectiveness and cost
      ref=2,                # selects the 2nd row of (e, c)
                            #  as containing the reference intervention
      interventions=treats, # defines the labels to be associated
                            #  with each intervention
      Kmax=50000,           # maximum value possible for the willingness
                            #  to pay threshold; implies that k is chosen
                            #  in a grid from the interval (0, Kmax)
      plot=FALSE             # plots the results
)
eib.plot(m)
eib.plot(m, graph = "ggplot2") + ggplot2::theme_linedraw()

data(Smoking)
treats <- c("No intervention", "Self-help",
            "Individual counselling", "Group counselling")
m <- bcea(e, c, ref = 4, interventions = treats, Kmax = 500)
eib.plot(m)
```

`eib_params_base` *EIB parameters specific to base R plot*

Description

EIB parameters specific to base R plot

Usage

```
eib_params_base(he, graph_params, cri_params)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`graph_params`	Type of plot device
`cri_params`	Credible interval parameters

Value

list

`eib_plot_graph` *Expected Incremental Benefit Plot By Graph Device*

Description

Choice of base R, ggplot2 or plotly.

Usage

```
eib_plot_base(he, graph_params, ...)

eib_plot_ggplot(he, graph_params, ...)

eib_plot_plotly(he, graph_params, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`graph_params`	Graph parameters
`...`	Additional parameters

`estimate.hyperparams` *Estimate hyperparameters*

Description

Estimate hyperparameters

Usage

```
estimate.hyperparams(x, input.matrix, parameter, n.sim)
```

Arguments

`x`	x
`input.matrix`	Input matrix
`parameter`	Parameters
`n.sim`	Number of simulations

See Also

`evppi`

`evi.plot.bcea` *Expected Value of Information (EVI) Plot*

Description

Plots the Expected Value of Information (EVI) against the willingness to pay.

Usage

```
## S3 method for class 'bcea'
evi.plot(he, graph = c("base", "ggplot2", "plotly"), ...)

evi.plot(he, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`graph`	A string used to select the graphical engine to use for plotting. Should (partial-)match the three options `"base"`, `"ggplot2"` or `"plotly"`. Default value is `"base"`.
`...`	Additional graphical arguments: • `line_colors` to specify the EVPI line colour - all graph types. • `line_types` to specify the line type (lty) - all graph types. • `area_include` to specify whether to include the area under the EVPI curve - plotly only. • `area_color` to specify the area under the colour curve - plotly only.

Value

`eib`	If `graph="ggplot2"` a ggplot object, or if `graph="plotly"` a plotly object containing the requested plot. Nothing is returned when `graph="base"`, the default.

The function produces a plot of the Expected Value of Information as a function of the discrete grid approximation of the willingness to pay parameter. The break even point(s) (i.e. the point in which the EIB=0, ie when the optimal decision changes from one intervention to another) is(are) also showed.

Author(s)

Gianluca Baio, Andrea Berardi

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`, `ceac.plot`, `ceplane.plot`

Examples

```
data(Vaccine)
m <- bcea(
      e=e,
      c=c,                  # defines the variables of
                            #  effectiveness and cost
      ref=2,                # selects the 2nd row of (e, c)
                            #  as containing the reference intervention
      interventions=treats, # defines the labels to be associated
                            #  with each intervention
      Kmax=50000,           # maximum value possible for the willingness
                            #  to pay threshold; implies that k is chosen
                            #  in a grid from the interval (0, Kmax)
      plot=FALSE            # plots the results
)
evi.plot(m)

data(Smoking)
treats <- c("No intervention", "Self-help",
            "Individual counselling", "Group counselling")
m <- bcea(e, c, ref = 4, interventions = treats, Kmax = 500)
evi.plot(m)
```

`evi.plot.mixedAn` *EVI Plot of the Health Economic Analysis For Mixed Analysis*

Description

Compares the optimal scenario to the mixed case in terms of the EVPI.

Usage

```
## S3 method for class 'mixedAn'
evi.plot(he, y.limits = NULL, pos = c(0, 1), graph = c("base", "ggplot2"), ...)
```

Arguments

`he`	An object of class `mixedAn`, a subclass of `bcea`, given as output of the call to the function `mixedAn`.
`y.limits`	Range of the y-axis for the graph. The default value is `NULL`, in which case the maximum range between the optimal and the mixed analysis scenarios is considered.
`pos`	Parameter to set the position of the legend (only relevant for multiple interventions, ie more than 2 interventions being compared). Can be given in form of a string `(bottom|top)(right|left)` for base graphics and `bottom|top|left|right` for ggplot2. It can be a two-elements vector, which specifies the relative position on the x and y axis respectively, or alternatively it can be in form of a logical variable, with `FALSE` indicating to use the default position and `TRUE` to place it on the bottom of the plot.
`graph`	A string used to select the graphical engine to use for plotting. Should (partial-)match the two options `"base"` or `"ggplot2"`. Default value is `"base"`.
`...`	Arguments to be passed to methods, such as graphical parameters (see `par`).

Value

`evi`	A ggplot object containing the plot. Returned only if `graph="ggplot2"`.

The function produces a graph showing the difference between the ”optimal” version of the EVPI (when only the most cost-effective intervention is included in the market) and the mixed strategy one (when more than one intervention is considered in the market).

Author(s)

Gianluca Baio, Andrea Berardi

References

Baio, G. and Russo, P. (2009). A decision-theoretic framework for the application of cost-effectiveness analysis in regulatory processes. Pharmacoeconomics 27(8), 645-655 doi:10.2165/11310250

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London

See Also

`bcea`, `mixedAn`

Examples

```
# See Baio G., Dawid A.P. (2011) for a detailed description of the
# Bayesian model and economic problem
#
# Load the processed results of the MCMC simulation model
```

```
data(Vaccine)

# Runs the health economic evaluation using BCEA
m <- bcea(e=e,c=c,          # defines the variables of
                            #  effectiveness and cost
      ref=2,                # selects the 2nd row of (e,c)
                            #  as containing the reference intervention
      interventions=treats, # defines the labels to be associated
                            #  with each intervention
      Kmax=50000,           # maximum value possible for the willingness
                            #  to pay threshold; implies that k is chosen
                            #  in a grid from the interval (0,Kmax)
      plot=FALSE            # inhibits graphical output
)

mixedAn(m) <- NULL      # uses the results of the mixed strategy
                        #  analysis (a "mixedAn" object)
                        # the vector of market shares can be defined
                        #  externally. If NULL, then each of the T
                        #  interventions will have 1/T market share
                        # produces the plots
evi.plot(m)

evi.plot(m, graph="base")

# Or with ggplot2
if (require(ggplot2)) {
   evi.plot(m, graph="ggplot2")
}
```

`evi_plot_graph` *Expected Value of Information Plot By Graph Device*

Description

Choice of base R, ggplot2 or plotly.

Usage

```
evi_plot_base(he, data.psa, plot_aes, plot_annotations)

evi_plot_ggplot(he, data.psa, plot_aes, plot_annotations)

evi_plot_plotly(data.psa, plot_aes, plot_annotations)
```

Arguments

he	A bcea object containing the results of the Bayesian modelling and the economic evaluation.

`data.psa`	Data
`plot_aes`	Aesthetic parameters
`plot_annotations`	Plot parameters

`evppi` *Expected Value of Perfect Partial Information (EVPPI) for Selected Parameters*

Description

Calculates the Expected Value of Perfect Partial Information (EVPPI) for subsets of parameters. Uses GAM non-parametric regression for single parameter EVPPI and the SPDE-INLA method for larger parameter subsets.

Usage

```
evppi(he, param_idx, input, N = NULL, plot = FALSE, residuals = TRUE, ...)

## S3 method for class 'bcea'
evppi(he, param_idx, input, N = NULL, plot = FALSE, residuals = TRUE, ...)

## Default S3 method:
evppi(he, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`param_idx`	A vector of parameters for which the EVPPI should be calculated. This can be given as a string (or vector of strings) of names or a numeric vector, corresponding to the column numbers of important parameters.
`input`	A matrix containing the simulations for all the parameters monitored by the call to JAGS or BUGS. The matrix should have column names matching the names of the parameters and the values in the vector parameter should match at least one of those values.
`N`	The number of PSA simulations used to calculate the EVPPI. The default uses all the available samples.
`plot`	A logical value indicating whether the triangular mesh for SPDE-INLA should be plotted. Default set to ‘FALSE‘.
`residuals`	A logical value indicating whether the fitted values for the SPDE-INLA method should be outputted. Default set to ‘TRUE‘.

`...`	Additional arguments. The default methods to compute the EVPPI are: - For single-parameter: GAM regression. - For multi-parameter: INLA/SPDE. However, it is possible (mainly for backward compatibility) to use different methods. For single-parameter, the user can specify the method of Sadatsafavi et al or the method of Strong & Oakley. In order to do so, it is necessary to include the extra parameter `method` which takes as value a string `"sad"` in the former case and a string `"so"` in the latter. In case "sal" is selected, then it is possible to also specify the number of "separators" (e.g. `n.seps=3`). If none is specified, the default value `n.seps=1` is used. If `"so"` is used as method for the calculation of the EVPPI, then the user *needs* to also specify the number of "blocks" (e.g. `n.blocks=20`).

Details

The single parameter EVPPI has been calculated using the non-parametric GAM regression developed by Strong et al. (2014). The multi-parameter EVPPI is calculated using the SPDE-INLA regression method for Gaussian Process regression developed by Heath et al. (2015).

This function has been completely changed and restructured to make it possible to change regression method. The method argument can now be given as a list. The first element element in the list is a vector giving the regression method for the effects. The second gives the regression method for the costs. The 'method' argument can also be given as before which then uses the same regression method for all curves. All other 'extra_args' can be given as before. 'int.ord' can be updated using the list formulation above to give the interactions for each different curve. The formula argument for GAM can only be given once, either 'te()' or 's() + s()' as this is for computational reasons rather than to aid fit. You can still plot the INLA mesh elements but not output the meshes.

Value

Object of class 'evppi':

`evppi`	The computed values of evppi for all values of the parameter of willingness to pay.
`index`	A numerical vector with the index associated with the parameters for which the EVPPI was calculated.
`k`	The vector of values for the willingness to pay.
`evi`	The vector of values for the overall EVPPI.
`fitted.costs`	The fitted values for the costs.
`fitted.effects`	The fitted values for the effects.
`parameters`	A single string containing the names of the parameters for which the EVPPI was calculated, used for plotting the EVPPI.
`time`	Computational time (in seconds).
`fit.c`	The object produced by the model fit for the costs.
`fit.e`	The object produced by the model fit for the effects.
`formula`	The formula used to fit the model.
`method`	A string indicating the method used to estimate the EVPPI.

GAM regression

For multi-parameter, the user can select 3 possible methods. If `method = "GAM"` (BCEA will accept also `"gam"`, `"G"` or `"g"`), then the computations are based on GAM regression. The user can also specify the formula for the regression. The default option is to use a tensor product (e.g. if there are two main parameters, `p1` and `p2`, this amounts to setting `formula = "te(p1,p2)"`, which indicates that the two parameters interact). Alternatively, it is possible to specify a model in which the parameters are independent using the notation `formula = "s(p1) + s(p2)"`. This may lead to worse accuracy in the estimates.

Strong et al. GP regression

This is used if `method="GP"` (BCEA will also accept the specification `method="gp"`). In this case, the user can also specify the number of PSA runs that should be used to estimate the hyperparameters of the model (e.g. `n.sim=100`). This value is set by default to 500.

INLA-related options

These are all rather technical and are described in detail in Baio et al. (2017). The optional parameter vector `int.ord` can take integer values (c(1,1) is default) and will force the predictor to include interactions: if `int.ord = c(k,h)`, then all k-way interactions will be used for the effects and all h-way interactions will be used for the costs. Also, the user can specify the feature of the mesh for the "spatial" part of the model. The optional parameter `cutoff` (default 0.3) controls the density of the points inside the mesh. Acceptable values are typically in the interval (0.1, 0.5), with lower values implying more points (and thus better approximation and greater computational time). The construction of the boundaries for the mesh can be controlled by the optional inputs `convex.inner` (default = -0.4) and `convex.outer` (default = -0.7). These should be negative values and can be decreased (say to -0.7 and -1, respectively) to increase the distance between the points and the outer boundary, which also increases precision and computational time. The optional argument`robust` can be set to TRUE, in which case INLA will use a t prior distribution for the coefficients of the linear predictor. Finally, the user can control the accuracy of the INLA grid-search for the estimation of the hyperparameters. This is done by setting a value `h.value` (default = 0.00005). Lower values imply a more refined search (and hence better accuracy), at the expense of computational speed. The method argument can also be given as a list allowing different regression methods for the effects and costs, and the different incremental decisions. The first list element should contain a vector of methods for the incremental effects and the second for the costs, for example `method = list(c("GAM"),c("INLA"))`. The `int.ord` argument can also be given as a list to give different interaction levels for each regression curve.

By default, when no method is specified by the user, `evppi` will use GAM if the number of parameters is <5 and INLA otherwise.

Author(s)

Anna Heath, Gianluca Baio

References

Strong M., Oakley J. and Brennan A. (2014). Estimating multi-parameter partial Expected Value of Perfect Information from a probabilistic sensitivity analysis sample: a non-parametric regression approach. Medical Decision Making.

Sadatsafavi M., Bansback N., Zafari Z., Najafzadeh M., Marra C. (2013). Need for speed: an efficient algorithm for calculation of single-parameter expected value of partial perfect information. Value in Health.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

Baio, G, A Berardi, and A Heath. 2017. Bayesian Cost-Effectiveness Analysis with the R package BCEA. New York, NY: Springer. doi:10.1007/978-3-319-55718-2.

Heath A., Manolopoulou I., Baio G. (2016). Estimating the Expected Value of Partial Perfect Information in Health Economic Evaluations using Integrated Nested Laplace Approximation. Statistics in Medicine. http://onlinelibrary.wiley.com/doi/10.1002/sim.6983/full

See Also

`bcea`, `plot.evppi`,

Examples

```
# See Baio G., Dawid A.P. (2011) for a detailed description of the
# Bayesian model and economic problem

# Load the processed results of the MCMC simulation model
data(Vaccine, package = "BCEA")
treats <- c("Status quo", "Vaccination")

# Run the health economic evaluation using BCEA
m <- bcea(e.pts, c.pts, ref = 2, interventions = treats)

# Compute the EVPPI for a bunch of parameters
inp <- createInputs(vaccine)

EVPPI <- evppi(m, c("beta.1." , "beta.2."), inp$mat)

## Not run:
plot(EVPPI)

# deprecated (single parameter) methods
EVPPI.so <- evppi(m, c("beta.1.", "beta.2."), inp$mat, method = "so", n.blocks = 50)
EVPPI.sad <- evppi(m, c("beta.1.", "beta.2."), inp$mat, method = "sad", n.seps = 1)

plot(EVPPI.so)
plot(EVPPI.sad)

# Compute the EVPPI using INLA/SPDE
x_inla <- evppi(he = m, 39:40, input = inp$mat)

# using GAM regression
x_gam <- evppi(he = m, 39:40, input = inp$mat, method = "GAM")

# using Strong et al GP regression
x_gp <- evppi(he = m, 39:40, input = inp$mat, method = "GP")

# plot results
```

```
plot(x_inla)
points(x_inla$k, x_inla$evppi, type = "l", lwd = 2, lty = 2)
points(x_gam$k, x_gam$evppi, type = "l", col = "red")
points(x_gp$k, x_gp$evppi, type = "l", col = "blue")

plot(x_inla$k, x_inla$evppi, type = "l", lwd = 2, lty = 2)
points(x_gam$k, x_gam$evppi, type = "l", col = "red")
points(x_gp$k, x_gp$evppi, type = "l", col = "blue")

data(Smoking)
treats <- c("No intervention", "Self-help",
"Individual counselling", "Group counselling")
m <- bcea(e, c, ref = 4, interventions = treats, Kmax = 500)
inp <- createInputs(smoking_output)
EVPPI <- evppi(m, c(2,3), inp$mat, h.value = 0.0000005)
plot(EVPPI)

## End(Not run)
```

`evppi_plot_graph` *Plot Expected Value of Partial Information With Respect to a Set of Parameters*

Description

Base R and ggplot2 versions.

Usage

```
evppi_plot_base(evppi_obj, pos_legend, col = NULL, annot = FALSE)

evppi_plot_ggplot(evppi_obj, pos_legend = c(0, 0.8), col = c(1, 1), ...)
```

Arguments

`evppi_obj`	Object of class evppi
`pos_legend`	Position of legend
`col`	Colour
`annot`	Annotate EVPPI curve with parameter names
`...`	Additional arguments

`fit.gam` *Gaussian Additive Model Fitting*

Description

Gaussian Additive Model Fitting

Usage

```
fit.gam(parameter, inputs, x, form)
```

Arguments

`parameter`	Parameter
`inputs`	Inputs
`x`	Response variable
`form`	Formula

Value

List

See Also

`evppi`

`fit.gp` *Fit Gaussian Process*

Description

Fit Gaussian Process

Usage

```
fit.gp(parameter, inputs, x, n.sim)
```

Arguments

`parameter`	Parameters
`inputs`	Inputs
`x`	Response variable
`n.sim`	Number of simulations

Value

list

See Also

`evppi`

`fit.inla` *Fit INLA*

Description

Fit INLA

Usage

```
fit.inla(
  parameter,
  inputs,
  x,
  mesh,
  data.scale,
  int.ord,
  convex.inner,
  convex.outer,
  cutoff,
  max.edge,
  h.value,
  family
)
```

Arguments

`parameter`	Parameters
`inputs`	Inputs
`x`	Response variable
`mesh`	Mesh
`data.scale`	data.scale
`int.ord`	int.ord
`convex.inner`	convex.inner
`convex.outer`	convex.outer
`cutoff`	Cut-off
`max.edge`	Maximum edge
`h.value`	h.value
`family`	family

Value

list

See Also

evppi

`geom_cri` *Credible interval ggplot geom*

Description

Credible interval ggplot geom

Usage

```
geom_cri(plot.cri = TRUE, params = NA)
```

Arguments

`plot.cri`	Should we plot CrI? Logical
`params`	Plot parameters including data

`ib.plot.bcea` *Incremental Benefit (IB) Distribution Plot*

Description

Plots the distribution of the Incremental Benefit (IB) for a given value of the willingness to pay threshold.

Usage

```
## S3 method for class 'bcea'
ib.plot(
  he,
  comparison = NULL,
  wtp = 25000,
  bw = "bcv",
  n = 512,
  xlim = NULL,
  graph = c("base", "ggplot2"),
  ...
)

ib.plot(he, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`comparison`	In the case of multiple interventions, specifies the one to be used in comparison with the reference. Default value of `NULL` forces R to consider the first non-reference intervention as the comparator. Controls which comparator is used when more than 2 interventions are present
`wtp`	The value of the willingness to pay threshold. Default value at `25000`.
`bw`	Identifies the smoothing bandwidth used to construct the kernel estimation of the IB density.
`n`	The number of equally spaced points at which the density is to be estimated.
`xlim`	The limits of the plot on the x-axis.
`graph`	A string used to select the graphical engine to use for plotting. Should (partial-) match the two options `"base"` or `"ggplot2"`. Default value is `"base"`.
`...`	Additional arguments

Value

`ib`	A ggplot object containing the requested plot. Returned only if `graph="ggplot2"`.

The function produces a plot of the distribution of the Incremental Benefit for a given value of the willingness to pay parameter. The dashed area indicates the positive part of the distribution (i.e. when the reference is more cost-effective than the comparator).

Author(s)

Gianluca Baio, Andrea Berardi

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`, `ceplane.plot`

`ib_plot_base` *IB plot base R version*

Description

Choice of base R, ggplot2

Usage

```
ib_plot_base(he, comparison, wtp, bw, n, xlim)

ib_plot_ggplot(he, comparison, wtp, bw, n, xlim)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`comparison`	Comparison intervention
`wtp`	Willingness to pay
`bw`	band width
`n`	Number
`xlim`	x-axis limits

`info.rank.bcea` *Information-Rank Plot for bcea Class*

Description

Produces a plot similar to a tornado plot, but based on the analysis of the EVPPI. For each parameter and value of the willingness-to-pay threshold, a barchart is plotted to describe the ratio of EVPPI (specific to that parameter) to EVPI. This represents the relative ‘importance’ of each parameter in terms of the expected value of information.

Usage

```
## S3 method for class 'bcea'
info.rank(
  he,
  inp,
  wtp = he$k[min(which(he$k >= he$ICER))],
  howManyPars = NA,
  graph = c("base", "ggplot2", "plotly"),
  rel = TRUE,
  ...
)

info.rank(he, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`inp`	Named list from running `createInputs()` containing: • `parameter` = A vector of parameters for which the individual EVPPI should be calculated. This can be given as a string (or vector of strings) of names or a numeric vector, corresponding to the column numbers of important parameters. • `mat` = A matrix containing the simulations for all the parameters monitored by the call to JAGS or BUGS. The matrix should have column names matching the names of the parameters and the values in the vector parameter should match at least one of those values.
`wtp`	A value of the wtp for which the analysis should be performed. If not specified then the break-even point for the current model will be used.
`howManyPars`	Optional maximum number of parameters to be included in the bar plot. Includes all parameters by default.
`graph`	A string used to select the graphical enging to use for plotting. Should (partial-)match one of the two options "base" or "plotly". Default value is "base"
`rel`	Logical argument that specifies whether the ratio of EVPPI to EVPI (`rel = TRUE`, default) or the absolute value of the EVPPI should be used for the analysis.
`...`	Additional options. These include graphical parameters that the user can specify: • `xlim` = limits of the x-axis; ca = font size for the axis label (default = 0.7 of full size). • `cn` = font size for the parameter names vector (default = 0.7 of full size) - base graphics only. • `mai` = margins of the graph (default = c(1.36, 1.5, 1,1)) - base graphics only.

Value

With base graphics: A data.frame containing the ranking of the parameters with the value of the selected summary, for the chosen wtp; with plotly: a plotly object, incorporating in the $rank element the data.frame as above. The function produces a 'Info-rank' plot. This is an extension of standard 'Tornado plots' and presents a ranking of the model parameters in terms of their impact on the expected value of information. For each parameter, the specific individual EVPPI is computed and used to measure the impact of uncertainty in that parameter over the decision-making process, in terms of how large the expected value of gaining more information is.

Author(s)

Anna Heath, Gianluca Baio, Andrea Berardi

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

bcea, evppi

Examples

```
data("Vaccine")
m <- bcea(e,c)
inp <- createInputs(vaccine)
info.rank(m, inp)

## Not run:
info.rank(m, inp, graph = "base")
info.rank(m, inp, graph = "plotly")
info.rank(m, inp, graph = "ggplot2")

## End(Not run)
```

`inforank_params` *Prepare Info Rank plot parameters*

Description

Prepare Info Rank plot parameters

Usage

```
inforank_params(he, inp, wtp, rel, howManyPars, extra_args)
```

Arguments

`he`	A bcea object containing the results of the Bayesian modelling and the economic evaluation.
`inp`	Inputs
`wtp`	Willingness to pay
`rel`	Relative size
`howManyPars`	How mnay parameters to use?
`extra_args`	Additional arguments

`info_rank_graph` *Info Rank Plot By Graph Device*

Description

Choice of base R, ggplot2 and plotly.

Usage

```
info_rank_base(he, params)

info_rank_ggplot(he, params)

info_rank_plotly(params)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`params`	Graph Parameters including data

`is.bcea` *Check bcea Class*

Description

Check bcea Class

Usage

```
is.bcea(he)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.

Value

`is.bcea` returns TRUE or FALSE depending on whether its argument is a bcea class object.

`kstar_vlines` *Prepare K-star vertical lines*

Description

Prepare K-star vertical lines

Usage

```
kstar_vlines(he, plot_params)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`plot_params`	Plots parameters

`line_labels` *Create Labels for Plot*

Description

Create Labels for Plot

Usage

```
line_labels(he, ...)

## Default S3 method:
line_labels(he, ...)

## S3 method for class 'pairwise'
line_labels(he, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`...`	Additional arguments

`loo_rank` *Leave-one-out ranking*

Description

Leave-one-out ranking

Usage

```
loo_rank(params)
```

Arguments

`params`	Parameters

`make.mesh` *Make Mesh*

Description

Fit using INLA methods.

Usage

```
make.mesh(data, convex.inner, convex.outer, cutoff, max.edge)
```

Arguments

`data`	Data
`convex.inner`	convex.inner
`convex.outer`	convex.outer
`cutoff`	Cut-off value
`max.edge`	Maximum edge

Value

list

See Also

`evppi`

`make.proj` *INLA Fitting*

Description

INLA Fitting

Usage

```
make.proj(parameter, inputs, x, k, l)
```

Arguments

`parameter`	Parameter
`inputs`	Inputs
`x`	Response variable
`k`	k
`l`	l

Value

list

See Also

`evppi`

`make.report` *Make Report*

Description

Constructs the automated report from the output of the BCEA.

Usage

```
make.report(he, evppi = NULL, ext = "pdf", echo = FALSE, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`evppi`	An object obtained as output to a call to `evppi` (default is NULL, so not essential to producing the report).
`ext`	A string of text to indicate the extension of the resulting output file. Possible options are `"pdf"`, `"docx"`. This requires the use of pandoc, knitr and rmarkdown.
`echo`	A string (default to `FALSE`) to instruct whether the report should also include the `BCEA` commands used to produce the analyses. If the optional argument `echo` is set to `TRUE` (default = `FALSE`), then the commands are also printed.
`...`	Additional parameters. For example, the user can specify the value of the willingness to pay `wtp`, which is used in some of the resulting analyses (default at the break even point). Another additional parameter that the user can specify is the name of the file to which the report should be written. This can be done by simply passing the optional argument `filename="NAME"`. The user can also specify an object including the PSA simulations for all the relevant model parameters. If this is passed to the function (in the object `psa_sims`), then `make.report` will automatically construct an "Info-rank plot", which is a probabilistic form of tornado plot, based on the Expected Value of Partial Information. The user can also specify the optional argument `show.tab` (default=FALSE); if set to `TRUE`, then a table with the values of the Info-rank is also shown.

Author(s)

Gianluca Baio

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`

Examples

```
## Not run:
  data(Vaccine, package = "BCEA")
  m <- bcea(e, c, ref = 2)
  make.report(m)

## End(Not run)
```

`make_legend_plotly` *Legend Positioning*

Description

Legend Positioning

Usage

```
make_legend_plotly(pos_legend)
```

Arguments

`pos_legend`	Position of legend

Value

String

`mixedAn<-` *Cost-Effectiveness Analysis When Multiple (Possibly Non-Cost-Effective) Interventions are Present on the Market*

Description

Runs the cost-effectiveness analysis, but accounts for the fact that more than one intervention is present on the market.

Usage

```
mixedAn(he) <- value
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`value`	A vector of market shares associated with the interventions. Its size is the same as the number of possible comparators. By default, assumes uniform distribution for each intervention.

Value

Creates an object in the class `mixedAn`, a subclass of `bcea` which contains the results of the health economic evaluation in the mixed analysis case:

`Ubar`	An array with the simulations of the ”known-distribution” mixed utilities, for each value of the discrete grid approximation of the willingness to pay parameter
`OL.star`	An array with the simulations of the distribution of the Opportunity Loss for the mixed strategy, for each value of the discrete grid approximation of the willingness to pay parameter
`evi.star`	The Expected Value of Information for the mixed strategy, for each value of the discrete grid approximation of the willingness to pay parameter
`mkt.shares`	The vector of market shares associated with each available intervention

Author(s)

Gianluca Baio

References

Baio, G. and Russo, P. (2009). A decision-theoretic framework for the application of cost-effectiveness analysis in regulatory processes. Pharmacoeconomics 27(8), 645-655 doi:10.2165/11310250

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`

Examples

```
# See Baio G., Dawid A.P. (2011) for a detailed description of the
# Bayesian model and economic problem

# Load the processed results of the MCMC simulation model
data(Vaccine)

# Runs the health economic evaluation using BCEA
m <- bcea(e=e,c=c,          # defines the variables of
                            #  effectiveness and cost
      ref=2,                # selects the 2nd row of (e, c)
                            #  as containing the reference intervention
      interventions=treats, # defines the labels to be associated
                            #  with each intervention
      Kmax=50000,           # maximum value possible for the willingness
                            #  to pay threshold; implies that k is chosen
                            #  in a grid from the interval (0, Kmax)
      plot=FALSE)           # inhibits graphical output
```

```
mixedAn(m) <- NULL      # uses the results of the mixed strategy
                        #  analysis (a "mixedAn" object)
                        # the vector of market shares can be defined
                        #  externally. If NULL, then each of the T
                        #  interventions will have 1/T market share
                        # produces the plots
evi.plot(m)
```

`multi.ce` *Cost-effectiveness Analysis With Multiple Comparison*

Description

Computes and plots the probability that each of the `n_int` interventions being analysed is the most cost-effective and the cost-effectiveness acceptability frontier.

Usage

```
## S3 method for class 'bcea'
multi.ce(he)
```

Arguments

`he`	A bcea object containing the results of the Bayesian modelling and the economic evaluation.

Value

Original bcea object (list) of class "pairwise" with additional:

`p_best_interv`	A matrix including the probability that each intervention is the most cost-effective for all values of the willingness to pay parameter
`ceaf`	A vector containing the cost-effectiveness acceptability frontier

Author(s)

Gianluca Baio

See Also

`bcea`, `ceaf.plot`

Examples

```
# See Baio G., Dawid A.P. (2011) for a detailed description of the
# Bayesian model and economic problem

# Load the processed results of the MCMC simulation model
data(Vaccine)

# Runs the health economic evaluation using BCEA

m <- bcea(e=e,c=c,          # defines the variables of
                            #  effectiveness and cost
      ref=2,                # selects the 2nd row of (e,c)
                            #  as containing the reference intervention
      interventions=treats, # defines the labels to be associated
                            #  with each intervention
      Kmax=50000,           # maximum value possible for the willingness
                            #  to pay threshold; implies that k is chosen
                            #  in a grid from the interval (0,Kmax)
      plot=FALSE            # inhibits graphical output
)

mce <- multi.ce(m)          # uses the results of the economic analysis

ceac.plot(mce)
ceaf.plot(mce)
```

`multiplot` *Plot Multiple bcea Graphs*

Description

Arrange plots in grid. Sourced from R graphics cookbook.

Usage

```
multiplot(plotlist = NULL, cols = 1, layout_config = NULL)
```

Arguments

`plotlist`	List of ggplot objects
`cols`	Number of columns
`layout_config`	Matrix of plot configuration

Value

ggplot TableGrob object

---

new_bcea *Constructor for bcea*

Description

Constructor for bcea

Usage

```
new_bcea(df_ce, k)
```

Arguments

`df_ce`	Dataframe of all simulation eff and cost
`k`	Vector of willingness to pay values

Value

List object of class bcea.

See Also

`bcea`

---

num_lines *Get number of lines*

Description

Get number of lines

Usage

```
num_lines(dat)

## S3 method for class 'pairwise'
num_lines(dat)

## S3 method for class 'bcea'
num_lines(dat)

## S3 method for class 'evppi'
num_lines(dat)

## Default S3 method:
num_lines(dat)
```

Arguments

`dat`	Data

`openPDF` *Automatically open pdf output using default pdf viewer*

Description

Automatically open pdf output using default pdf viewer

Usage

```
openPDF(file_name)
```

Arguments

`file_name`	String file names for pdf

`plot.bcea` *Summary Plot of the Health Economic Analysis*

Description

Plots in a single graph the Cost-Effectiveness plane, the Expected Incremental Benefit, the CEAC and the EVPI.

Usage

```
## S3 method for class 'bcea'
plot(
  x,
  comparison = NULL,
  wtp = 25000,
  pos = FALSE,
  graph = c("base", "ggplot2"),
  ...
)
```

Arguments

`x`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`comparison`	Selects the comparator, in case of more than two interventions being analysed. Default as NULL plots all the comparisons together. Any subset of the possible comparisons can be selected (e.g., `comparison=c(1,3)` or `comparison=2`).
`wtp`	The value of the willingness to pay parameter. It is passed to `ceplane.plot`.
`pos`	Parameter to set the position of the legend (only relevant for multiple interventions, ie more than 2 interventions being compared). Can be given in form of a string `(bottom|top)(right|left)` for base graphics and `bottom|top|left|right` for ggplot2. It can be a two-elements vector, which specifies the relative position on the x and y axis respectively, or alternatively it can be in form of a logical variable, with `FALSE` indicating to use the default position and `TRUE` to place it on the bottom of the plot.
`graph`	A string used to select the graphical engine to use for plotting. Should (partial-)match the two options `"base"` or `"ggplot2"`. Default value is `"base"`.
`...`	Arguments to be passed to the methods `ceplane.plot` and `eib.plot`. Please see the manual pages for the individual functions. Arguments like `size`, `ICER.size` and `plot.cri` can be supplied to the functions in this way. In addition if `graph="ggplot2"` and the arguments are named theme objects they will be added to each plot.

Details

The default position of the legend for the cost-effectiveness plane (produced by `ceplane.plot`) is set to `c(1,1.025)` overriding its default for `pos=FALSE`, since multiple ggplot2 plots are rendered in a slightly different way than single plots.

Value

A plot with four graphical summaries of the health economic evaluation.

Author(s)

Gianluca Baio, Andrea Berardi

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`, `ceplane.plot`, `eib.plot`, `ceac.plot`, `evi.plot`

Examples

```
# See Baio G., Dawid A.P. (2011) for a detailed description of the
# Bayesian model and economic problem

# Load the processed results of the MCMC simulation model
data(Vaccine)

# Runs the health economic evaluation using BCEA
he <- bcea(
       e=e, c=c,             # defines the variables of
                             #  effectiveness and cost
       ref=2,                # selects the 2nd row of (e,c)
                             #  as containing the reference intervention
       interventions=treats, # defines the labels to be associated
                             #  with each intervention
       Kmax=50000,           # maximum value possible for the willingness
                             #  to pay threshold; implies that k is chosen
                             #  in a grid from the interval (0,Kmax)
       plot=FALSE            # does not produce graphical outputs
      )

# Plots the summary plots for the "bcea" object m using base graphics
plot(he, graph = "base")

# Plots the same summary plots using ggplot2
if(require(ggplot2)){
plot(he, graph = "ggplot2")

##### Example of a customized plot.bcea with ggplot2
plot(he,
  graph = "ggplot2",                                      # use ggplot2
 theme = theme(plot.title=element_text(size=rel(1.25))), # theme elements must have a name
  ICER_size = 1.5,                                       # hidden option in ceplane.plot
  size = rel(2.5)                                      # modifies the size of k = labels
  )                                                       # in ceplane.plot and eib.plot
}
```

`plot.CEriskav` *Plots EIB and EVPI for the Risk Aversion Case*

Description

Summary plot of the health economic analysis when risk aversion is included.

Usage

```
## S3 method for class 'CEriskav'
plot(x, pos = c(0, 1), graph = c("base", "ggplot2"), ...)
```

Arguments

`x`	An object of the class `CEriskav`, a subclass of `bcea`, containing the results of the economic analysis performed accounting for a risk aversion parameter (obtained as output of the function `CEriskav`).
`pos`	Parameter to set the position of the legend (only relevant for multiple interventions, ie more than 2 interventions being compared). Can be given in form of a string `(bottom|top)(right|left)` for base graphics and `bottom|top|left|right` for ggplot2. It can be a two-elements vector, which specifies the relative position on the x and y axis respectively, or alternatively it can be in form of a logical variable, with `FALSE` indicating to use the default position and `TRUE` to place it on the bottom of the plot.
`graph`	A string used to select the graphical engine to use for plotting. Should (partial-)match the two options `"base"` or `"ggplot2"`. Default value is `"base"`.
`...`	Arguments to be passed to methods, such as graphical parameters (see `par`).

Details

Plots the Expected Incremental Benefit and the Expected Value of Perfect Information when risk aversion is included in the utility function.

Value

`list(eib,evi)`	A two-elements named list of the ggplot objects containing the requested plots. Returned only if `graph="ggplot2"`.

The function produces two plots for the risk aversion analysis. The first one is the EIB as a function of the discrete grid approximation of the willingness parameter for each of the possible values of the risk aversion parameter, `r`. The second one is a similar plot for the EVPI.

Author(s)

Gianluca Baio, Andrea Berardi

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London

See Also

`bcea`, `CEriskav`

Examples

```
# See Baio G., Dawid A.P. (2011) for a detailed description of the
# Bayesian model and economic problem
#
```

```
# Load the processed results of the MCMC simulation model
data(Vaccine)
#
# Runs the health economic evaluation using BCEA
m <- bcea(e=e,c=c,          # defines the variables of
                            #  effectiveness and cost
      ref=2,                # selects the 2nd row of (e,c)
                            #  as containing the reference intervention
      interventions=treats, # defines the labels to be associated
                            #  with each intervention
      Kmax=50000,           # maximum value possible for the willingness
                            #  to pay threshold; implies that k is chosen
                            #  in a grid from the interval (0,Kmax)
      plot=FALSE            # inhibits graphical output
)
#
# Define the vector of values for the risk aversion parameter, r, eg:
r <- c(1e-10, 0.005, 0.020, 0.035)
#
# Run the cost-effectiveness analysis accounting for risk aversion

   CEriskav(m) <- r

#
# produce the plots

   plot(m)

## Alternative options, using ggplot2

   plot(m, graph = "ggplot2")
```

`plot.evppi` *Plot Expected Value of Partial Information With Respect to a Set of Parameters*

Description

Plot Expected Value of Partial Information With Respect to a Set of Parameters

Usage

```
## S3 method for class 'evppi'
plot(x, pos = c(0, 0.8), graph = c("base", "ggplot2"), col = c(1, 1), ...)
```

Arguments

`x`	An object in the class `evppi`, obtained by the call to the function `evppi`.
`pos`	Parameter to set the position of the legend (only relevant for multiple interventions, ie more than 2 interventions being compared). Can be given in form of a string `(bottom|top)(right|left)` for base graphics and `bottom|top|left|right` for ggplot2. It can be a two-elements vector, which specifies the relative position on the x and y axis respectively, or alternatively it can be in form of a logical variable, with `FALSE` indicating to use the default position and `TRUE` to place it on the bottom of the plot.
`graph`	A string used to select the graphical engine to use for plotting. Should (partial-) match the two options `"base"` or `"ggplot2"`. Default value is `"base"`.
`col`	Sets the colour for the lines depicted in the graph.
`...`	Arguments to be passed to methods, such as graphical parameters (see `par`).

Value

Plot with base R or ggplot2.

Author(s)

Gianluca Baio, Andrea Berardi

References

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`, `evppi`

Examples

```
data(Vaccine, package = "BCEA")
treats <- c("Status quo", "Vaccination")

# Run the health economic evaluation using BCEA
m <- bcea(e.pts, c.pts, ref = 2, interventions = treats)

# Compute the EVPPI for a bunch of parameters
inp <- createInputs(vaccine)

# Compute the EVPPI using INLA/SPDE
x0 <- evppi(m, c("beta.1." , "beta.2."), input = inp$mat)

plot(x0, pos = c(0,1))

## Not run:
x1 <- evppi(m, c(32,48,49), input = inp$mat)
```

```
plot(x1, pos = "topright")

plot(x0, col = c("black", "red"), pos = "topright")
plot(x0, col = c(2,3), pos = "bottomright")

plot(x0, pos = c(0,1), graph = "ggplot2")
plot(x1, pos = "top", graph = "ggplot2")

plot(x0, col = c("black", "red"), pos = "right", graph = "ggplot2")
plot(x0, col = c(2,3), size = c(1,2), pos = "bottom", graph = "ggplot2")

plot(x0, graph = "ggplot2", theme = ggplot2::theme_linedraw())

if (FALSE)
 plot(x0, col = 3, pos = "topright")
# The vector 'col' must have the number of elements for an EVPI
# colour and each of the EVPPI parameters. Forced to black

## End(Not run)
```

`plot.mesh` *Mesh Plot*

Description

Option of interactively saving the plot.

Usage

```
## S3 method for class 'mesh'
plot(mesh, data, plot)
```

Arguments

`mesh`	Mesh
`data`	Data
`plot`	Create plot? logical

See Also

evppi

`plot_eib_cri` *Plot Credible Intervals*

Description

Bayesian posterior credible intervals against willingness to pay.

Usage

```
plot_eib_cri(he, params)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`params`	Graph parameters

`post.density` *Gaussian Process Fitting*

Description

Gaussian Process Fitting

Usage

```
post.density(hyperparams, parameter, x, input.matrix)
```

Arguments

`hyperparams`	Hyperparameters
`parameter`	Parameters
`x`	Response variable
`input.matrix`	Input data matrix

See Also

`evppi`

`prep.x` *Prepare Delta arrays*

Description

Prepare Delta arrays

Usage

```
prep.x(he, seq_rows, k, l)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`seq_rows`	Rows of (e,c) to keep
`k`	e or c? 1 or 2.
`l`	Columns of (e,c) to keep

See Also

`evppi`

`prepare.output` *Prepare output*

Description

Prepare output

Usage

```
prepare.output(parameters, inputs)
```

Arguments

`parameters`	Parameters
`inputs`	Inputs

Value

name

See Also

`evppi`

`prep_ceplane_params` *Prepare CE-plane Parameters*

Description

In ggplot format, combine user-supplied parameters with defaults.

Usage

```
prep_ceplane_params(he, wtp, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`wtp`	Willingness-to-pay
`...`	Additional arguments

Value

List pf graph parameters

`prep_eib_params` *Prepare EIB plot parameters*

Description

Parameters general to all plotting devices.

Usage

```
prep_eib_params(he, plot.cri, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`plot.cri`	Make title including credible interval? Logical
`...`	Additional parameters

Value

List of graph parameters

`prep_frontier_data` *Prepare frontier data*

Description

Prepare frontier data

Usage

```
prep_frontier_data(he, threshold = NULL, start.origin = TRUE)
```

Arguments

`he`	A bcea object containing the results of the Bayesian modelling and the economic evaluation.
`threshold`	Cost-effectiveness threshold i.e angle of line. Must be >=0 or NULL.
`start.origin`	Where should the frontier start from?

Value

List with scatter.data, ceef.points, orig.avg

See Also

ceef.plot

`print.bcea` *bcea Print Method*

Description

bcea Print Method

Usage

```
## S3 method for class 'bcea'
print(x, digits = getOption("digits"), give.attr = FALSE, no.list = TRUE, ...)
```

Arguments

`x`	A bcea object containing the results of the Bayesian modelling and the economic evaluation.
`digits`	Minimal number of significant digits, see `print.default`.
`give.attr`	Logical; if TRUE (default), show attributes as sub structures.
`no.list`	Logical; if TRUE, no ‘list of ...’ nor the class are printed.
`...`	Potential further arguments.

Examples

```
data("Vaccine")
he <- BCEA::bcea(e, c)
```

`quiet` *Allow disabling of the cat messages*

Description

Allow disabling of the cat messages

Usage

```
quiet(x)
```

Arguments

`x`	Object to quietly return

`select_plot_type` *Choose Graphical Engine*

Description

From base R, ggplot2 or plotly.

Usage

```
select_plot_type(graph)
```

Arguments

`graph`	Type names; string

Value

Plot ID integer 1:base R; 2:ggplot2; 3:plotly

`setComparisons` *Set Comparisons Group*

Description

One of the alternative way to set (e,c) comparison group. Simply recompute all comparisons and drop unwanted.

Usage

```
setComparisons(he, comparison)
```

Arguments

`he`	A bcea object containing the results of the Bayesian modelling and the economic evaluation.
`comparison`	Selects the comparator, in case of more than two interventions being analysed. Default as NULL plots all the comparisons together. Any subset of the possible comparisons can be selected (e.g., `comparison=c(1,3)` or `comparison=2`).

See Also

`setComparisons<-`

`setComparisons_assign` *Set Comparison Group*

Description

One of the alternative way to set (e,c) comparison group.

Usage

```
setComparisons(he) <- value

## S3 replacement method for class 'bcea'
setComparisons(he) <- value

## Default S3 replacement method:
setComparisons(he) <- value
```

Arguments

`he`	A bcea object containing the results of the Bayesian modelling and the economic evaluation.
`value`	Comparison

Value

bcea-type object

See Also

`setComparisons`

`setKmax_assign` *Set Maximum Willingness to Pay*

Description

Alternative way to define ‘K‘ statistic.

Usage

```
setKmax(he) <- value

## S3 replacement method for class 'bcea'
setKmax(he) <- value

## Default S3 replacement method:
setKmax(he) <- value
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`value`	Maximum willingness to pay

Value

bcea-type object

`setReferenceGroup_assign` *Set Reference Group*

Description

Alternative way to define (e,c) reference group.

Usage

```
setReferenceGroup(he) <- value

## S3 replacement method for class 'bcea'
setReferenceGroup(he) <- value

## Default S3 replacement method:
setReferenceGroup(he) <- value
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`value`	Reference group number

Value

bcea-type object

`sim_table` *Table of Simulation Statistics for the Health Economic Model*

Description

Using the input in the form of MCMC simulations and after having run the health economic model, produces a summary table of the simulations from the cost-effectiveness analysis.

Usage

```
sim_table(he, ...)

## S3 method for class 'bcea'
sim_table(he, wtp = 25000, ...)
```

Arguments

`he`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`...`	Additional arguments
`wtp`	The value of the willingness to pay threshold to be used in the summary table.

Value

Produces the following elements:

`table`	A table with simulation statistics from the economic model
`names.cols`	A vector of labels to be associated with each column of the table
`wtp`	The selected value of the willingness to pay
`idx_wtp`	The index associated with the selected value of the willingness to pay threshold in the grid used to run the analysis

Author(s)

Gianluca Baio

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`

Examples

```
# See Baio G., Dawid A.P. (2011) for a detailed description of the
# Bayesian model and economic problem

# Load the processed results of the MCMC simulation model
data(Vaccine)

# Runs the health economic evaluation using BCEA
m <- bcea(e=e,                 # defines the variables of
          c=c,                 # effectiveness and cost
          ref=2,               # selects the 2nd row of (e, c)
                               # as containing the reference intervention
          interventions=treats, # defines the labels to be associated
                               # with each intervention
          Kmax=50000)          # maximum value possible for the willingness
                               # to pay threshold; implies that k is chosen
                               # in a grid from the interval (0, Kmax)

# Now can save the simulation exercise in an object using sim_table()
sim_table(m,         # uses the results of the economic evaluation
                     #  (a 'bcea' object)
          wtp=25000) # selects the particular value for k
```

`Smoking` *Data set for the Bayesian model for the cost-effectiveness of smoking cessation interventions*

Description

This data set contains the results of the Bayesian analysis used to model the clinical output and the costs associated with the health economic evaluation of four different smoking cessation interventions.

Format

A data list including the variables needed for the smoking cessation cost-effectiveness analysis. The variables are as follows:

list("c") a matrix of 500 simulations from the posterior distribution of the overall costs associated with the four strategies

list("data") a dataset containing the characteristics of the smokers in the UK population

list("e") a matrix of 500 simulations from the posterior distribution of the clinical benefits associated with the four strategies

list("life.years") a matrix of 500 simulations from the posterior distribution of the life years gained with each strategy

list("pi") a matrix of 500 simulations from the posterior distribution of the event of smoking cessation with each strategy

list("smoking") a data frame containing the inputs needed for the network meta-analysis model. The `data.frame` object contains: `nobs`: the record ID number, `s`: the study ID number, `i`: the intervention ID number, `r_i`: the number of patients who quit smoking, `n_i`: the total number of patients for the row-specific arm and `b_i`: the reference intervention for each study

list("smoking_output") a `rjags` object obtained by running the network meta-analysis model based on the data contained in the `smoking` object

list("smoking_mat") a matrix obtained by running the network meta-analysis model based on the data contained in the `smoking` object

list("treats") a vector of labels associated with the four strategies

Source

Effectiveness data adapted from Hasselblad V. (1998). Meta-analysis of Multitreatment Studies. Medical Decision Making 1998;18:37-43. Cost and population characteristics data adapted from various sources:

- Taylor, D.H. Jr, et al. (2002). Benefits of smoking cessation on longevity. American Journal of Public Health 2002;92(6)
- ASH: Action on Smoking and Health (2013). ASH fact sheet on smoking statistics, `http://ash.org.uk/files/documents/ASH_106.pdf`

- Flack, S., et al. (2007). Cost-effectiveness of interventions for smoking cessation. York Health Economics Consortium, January 2007
- McGhan, W.F.D., and Smith, M. (1996). Pharmacoeconomic analysis of smoking-cessation interventions. American Journal of Health-System Pharmacy 1996;53:45-52

References

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London

Examples

```

data(Smoking)


m <- bcea(e, c, ref = 4, interventions = treats, Kmax = 500)
```

`struct.psa` *Structural Probability Sensitivity Analysis*

Description

Computes the weights to be associated with a set of competing models in order to perform structural PSA.

Usage

```
struct.psa(
  models,
  effect,
  cost,
  ref = NULL,
  interventions = NULL,
  Kmax = 50000,
  plot = FALSE
)
```

Arguments

`models`	A list containing the output from either R2jags or R2WinBUGS for all the models that need to be combined in the model average
`effect`	A list containing the measure of effectiveness computed from the various models (one matrix with n.sim x n.ints simulations for each model)
`cost`	A list containing the measure of costs computed from the various models (one matrix with n.sim x n.ints simulations for each model)

`ref`	Which intervention is considered to be the reference strategy. The default value `ref=1` means that the intervention appearing first is the reference and the other(s) is(are) the comparator(s)
`interventions`	Defines the labels to be associated with each intervention. By default and if `NULL`, assigns labels in the form "Intervention1", ... , "InterventionT"
`Kmax`	Maximum value of the willingness to pay to be considered. Default value is `k=50000`. The willingness to pay is then approximated on a discrete grid in the interval `[0,Kmax]`. The grid is equal to `wtp` if the parameter is given, or composed of `501` elements if `wtp=NULL` (the default)
`plot`	A logical value indicating whether the function should produce the summary plot or not

Details

The model is a list containing the output from either R2jags or R2WinBUGS for all the models that need to be combined in the model average effect is a list containing the measure of effectiveness computed from the various models (one matrix with n_sim x n_ints simulations for each model) cost is a list containing the measure of costs computed from the various models (one matrix with n_sim x n_ints simulations for each model).

Value

List object of bcea object, model weights and DIC

Author(s)

Gianluca Baio

References

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`

`summary.bcea` *Summary Method for Objects of Class* `bcea`

Description

Produces a table printout with some summary results of the health economic evaluation.

Usage

```
## S3 method for class 'bcea'
summary(object, wtp = 25000, ...)
```

Arguments

`object`	A `bcea` object containing the results of the Bayesian modelling and the economic evaluation.
`wtp`	The value of the willingness to pay threshold used in the summary table.
`...`	Additional arguments affecting the summary produced.

Value

Prints a summary table with some information on the health economic output and synthetic information on the economic measures (EIB, CEAC, EVPI).

Author(s)

Gianluca Baio

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

bcea

Examples

```
data(Vaccine)

he <- bcea(e, c, interventions = treats, ref = 2)
summary(he)
```

`summary.mixedAn` *Summary Methods For Objects in the Class* `mixedAn` *(Mixed Analysis)*

Description

Prints a summary table for the results of the mixed analysis for the economic evaluation of a given model.

Usage

```
## S3 method for class 'mixedAn'
summary(object, wtp = 25000, ...)
```

Arguments

`object`	An object of the class `mixedAn`, which is the results of the function `mixedAn`, generating the economic evaluation of a set of interventions, considering given market shares for each option.
`wtp`	The value of the willingness to pay choosen to present the analysis.
`...`	Additional arguments affecting the summary produced.

Value

Produces a table with summary information on the loss in expected value of information generated by the inclusion of non cost-effective interventions in the market.

Author(s)

Gianluca Baio

References

Baio, G. and Russo, P. (2009). A decision-theoretic framework for the application of cost-effectiveness analysis in regulatory processes. Pharmacoeconomics 27(8), 645-655 doi:10.2165/11310250

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Baio G. (2012). Bayesian Methods in Health Economics. CRC/Chapman Hall, London.

See Also

`bcea`, `mixedAn`

Examples

```
# See Baio G., Dawid A.P. (2011) for a detailed description of the
# Bayesian model and economic problem

# Load the processed results of the MCMC simulation model
data(Vaccine)

# Runs the health economic evaluation using BCEA
m <- bcea(e=e,c=c,          # defines the variables of
                            #  effectiveness and cost
      ref=2,                # selects the 2nd row of (e,c)
                            #  as containing the reference intervention
      interventions=treats, # defines the labels to be associated
                            #  with each intervention
      Kmax=50000            # maximum value possible for the willingness
                            #  to pay threshold; implies that k is chosen
                            #  in a grid from the interval (0,Kmax)
)
```

```
mixedAn(m) <- NULL      # uses the results of the mixed strategy
                        #  analysis (a "mixedAn" object)
                        # the vector of market shares can be defined
                        #  externally. If NULL, then each of the T
                        #  interventions will have 1/T market share

# Prints a summary of the results
summary(m,         # uses the results of the mixed strategy analysis
        wtp=25000) #  (a "mixedAn" object)
                   # selects the relevant willingness to pay
                   #  (default: 25,000)
```

`tabulate_means` *Calculate Dataset For ICERs From bcea Object*

Description

Calculate Dataset For ICERs From bcea Object

Usage

```
tabulate_means(he, comp_label = NULL, ...)
```

Arguments

`he`	A bcea object containing the results of the Bayesian modelling and the economic evaluation.
`comp_label`	Optional vector of strings with comparison labels
`...`	Additional arguments

Value

A data.frame object including mean outcomes, comparison identifier, comparison label and associated ICER

`Vaccine` *Data set for the Bayesian model for the cost-effectiveness of influenza vaccination*

Description

This data set contains the results of the Bayesian analysis used to model the clinical output and the costs associated with an influenza vaccination.

Format

A data list including the variables needed for the influenza vaccination. The variables are as follows:

list("c") a matrix of simulations from the posterior distribution of the overall costs associated with the two treatments

list("c.pts")

list("cost.GP") a matrix of simulations from the posterior distribution of the costs for GP visits associated with the two treatments

list("cost.hosp") a matrix of simulations from the posterior distribution of the costs for hospitalisations associated with the two treatments

list("cost.otc") a matrix of simulations from the posterior distribution of the costs for over-the-counter medications associated with the two treatments

list("cost.time.off") a matrix of simulations from the posterior distribution of the costs for time off work associated with the two treatments

list("cost.time.vac") a matrix of simulations from the posterior distribution of the costs for time needed to get the vaccination associated with the two treatments

list("cost.travel") a matrix of simulations from the posterior distribution of the costs for travel to get vaccination associated with the two treatments

list("cost.trt1") a matrix of simulations from the posterior distribution of the overall costs for first line of treatment associated with the two interventions

list("cost.trt2") a matrix of simulations from the posterior distribution of the overall costs for second line of treatment associated with the two interventions

list("cost.vac") a matrix of simulations from the posterior distribution of the costs for vaccination

list("e") a matrix of simulations from the posterior distribution of the clinical benefits associated with the two treatments

list("e.pts")

list("N") the number of subjects in the reference population

list("N.outcomes") the number of clinical outcomes analysed

list("N.resources") the number of health-care resources under study

list("QALYs.adv") a vector from the posterior distribution of the QALYs associated with advert events

list("QALYs.death") a vector from the posterior distribution of the QALYs associated with death

list("QALYs.hosp") a vector from the posterior distribution of the QALYs associated with hospitalisation

list("QALYs.inf") a vector from the posterior distribution of the QALYs associated with influenza infection

list("QALYs.pne") a vector from the posterior distribution of the QALYs associated with pneumonia

list("treats") a vector of labels associated with the two treatments

list("vaccine") a `rjags` object containing the simulations for the parameters used in the original model

list("vaccine_mat") a matrix containing the simulations for the parameters used in the original model

Source

Adapted from Turner D, Wailoo A, Cooper N, Sutton A, Abrams K, Nicholson K. The cost-effectiveness of influenza vaccination of healthy adults 50-64 years of age. Vaccine. 2006;24:1035-1043.

References

Baio, G., Dawid, A. P. (2011). Probabilistic Sensitivity Analysis in Health Economics. Statistical Methods in Medical Research doi:10.1177/0962280211419832.

Examples

```
data(Vaccine)


m <- bcea(e, c, ref = 1, interventions = treats)
```

`validate_bcea` *Validate bcea*

Description

Validate bcea

Usage

```
validate_bcea(eff, cost, ref, interventions)
```

Arguments

`eff`	Effectiveness matrix
`cost`	Cost matrix
`ref`	Reference intervention
`interventions`	All interventions

`validate_eib_params` *Validate EIB parameters*

Description

Validate EIB parameters

Usage

```
validate_eib_params(params)
```

Arguments

`params`	Graph parameters

Value

List of graph parameters

See Also

`prep_eib_params`

Index