

Package ‘BCEA’

November 26, 2025

Type Package

Title Bayesian Cost Effectiveness Analysis

Version 2.4.83

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, see Baio et al (2017) <[doi:10.1007/978-3-319-55718-2](https://doi.org/10.1007/978-3-319-55718-2)>.

Imports cli (>= 3.3.0), dplyr, ggplot2 (>= 3.5.0), graphics, gridExtra, MASS, Matrix, plotly, purrr, Rdpack, scales, tidyr, voi (>= 1.0.1)

Depends R (>= 4.1.0)

Suggests grid, knitr, mgcv, testthat (>= 2.1.0), withr

RdMacros Rdpack

VignetteBuilder knitr

License GPL-3

URL <https://gianluca.statistica.it/software/bcea/>,
<https://gianluca.statistica.it/>,
<https://n8thangreen.github.io/BCEA/>,
<https://github.com/giabaio/BCEA/>

NeedsCompilation no

RoxygenNote 7.3.2.9000

Encoding UTF-8

BugReports <https://github.com/giabaio/BCEA/issues/>

LazyData false

Author Gianluca Baio [aut, cre, cph] (ORCID: <<https://orcid.org/0000-0003-4314-2570>>),
Andrea Berardi [aut] (ORCID: <<https://orcid.org/0000-0002-2906-496X>>),
Anna Heath [aut] (ORCID: <<https://orcid.org/0000-0002-7263-4251>>),
Nathan Green [aut] (ORCID: <<https://orcid.org/0000-0003-2745-1736>>)

Maintainer Gianluca Baio <g.baio@ucl.ac.uk>

Repository CRAN

Date/Publication 2025-11-26 16:00:08 UTC

Contents

bcea	3
best_interv_given_k	8
ceac.plot.bcea	9
ceaf.plot.pairwise	11
ceef.plot.bcea	13
ceef_plot_graph	16
ceplane.plot.bcea	17
ceplane_plot_graph	19
CEriskav_assign	22
CEriskav_plot_graph	24
compute_CEAC	24
compute_ceaf	25
compute_EIB	25
compute_eib_cri	26
compute_EVI	27
compute_IB	27
compute_ICER	28
compute_kstar	28
compute_ol	29
compute_p_best_interv	30
compute_p_optimal_best	30
compute_U	31
compute_Ubar	31
compute_Ustar	32
compute_vi	32
contour.bcea	33
contour2.bcea	35
contour_graph	37
createInputs.default	38
eib.plot.bcea	39
eib_plot_graph	42
evi.plot.bcea	42
evi.plot.mixedAn	44
evi_plot_graph	46
evppi	46
evppi_plot_graph	51
ib.plot.bcea	52
ib_plot_base	53
info.rank.bcea	54
info_rank_graph	56
is.bcea	57

mixedAn<-	57
multi.ce	59
new_bcea	60
plot.bcea	61
plot.CEriskav	63
plot.evppi	65
print.bcea	67
setComparisons	67
setComparisons_assign	68
setKmax_assign	68
setReferenceGroup_assign	69
sim_table	70
Smoking	71
statins_base	72
statins_HC	73
struct.psa	73
summary.bcea	75
summary.mixedAn	76
summary.pairwise	78
tabulate_means	79
Vaccine	80

Index

82

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,
  k = NULL,
  plot = FALSE,
  ...
)
```

Default S3 method:

```

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

```

Arguments

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

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 analyzed.
- **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 (2013). *Bayesian Methods in Health Economics*. CRC.

Baio G, Dawid aP (2011). “Probabilistic sensitivity analysis in health economics.” *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Examples

```
# See Baio (2013), Baio (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=eff,
  c=cost,          # 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
)

# 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
    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	<i>Optimal intervention</i>
---------------------	-----------------------------

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	<i>Cost-Effectiveness Acceptability Curve (CEAC) Plot</i>
----------------	---

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 = "bottomright",
  graph = options("bcea.graph"),
  ...
)

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". This is set globally upon loading BCEA and can be modified for instance by using options("bcea.graph"="gg"), or options("bcea.graph"="plotly"). Partial matching still applies (so gg or pl also work). Not all plotting functions have a "plotly" implementation, yet – see help for the specifics.
...	If graph = "ggplot2" and a named theme object is supplied, it will be passed to the ggplot2 object. The usual ggplot2 syntax is used. Additional arguments: • line = list(color): specifies the line colour(s) - all graph types. • line = list(type): specifies the line type(s) as lty numeric values - all graph types.

- `line = list(size)`: specifies the line width(s) as numeric values - all graph types.
- `currency`: Currency prefix to willingness to pay values - ggplot2 only.
- `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:

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

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 aP (2011). "Probabilistic sensitivity analysis in health economics." *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

See Also

[bcea\(\)](#), [plot.bcea\(\)](#)

Examples

```

data("Vaccine")
he <- BCEA::bcea(eff, cost)
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(color = "green"),
          theme = ggplot2::theme_dark())

## more interventions
he2 <- BCEA::bcea(cbind(eff, eff - 0.0002), cbind(cost, cost + 5))
## could also construct a palette using RColorBrewer, e.g.
## mypalette <- RColorBrewer::brewer.pal(3, "Accent")
## and then call
## ceac.plot(he2, graph = "ggplot2",
##           title = "my title",
##           theme = ggplot2::theme_dark(),
##           pos = TRUE,
##           line = list(color = mypalette))
ceac.plot(he, graph = "base", title = "my title", line = list(color = "green"))
ceac.plot(he2, graph = "base")

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

```

ceaf.plot.pairwise	<i>Cost-Effectiveness Acceptability Frontier (CEAF) plot</i>
--------------------	--

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 = options("bcea.graph"), ...)

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 three options "base", "ggplot2" or "plotly". Default value is "base". This is set globally upon loading BCEA and can be modified for instance by using `options("bcea.graph"="gg")`, or `options("bcea.graph"="plotly")`. Partial matching still applies (so gg or pl also work). Not all plotting functions have a "plotly" implementation, yet – see help for the specifics.

... 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." *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

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=eff,
  c=cost,          # 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(eff, cost, ref = 4, intervention = treats, Kmax = 500, plot = FALSE)
mce <- multi.ce(m)
ceaf.plot(mce)

```

ceef.plot.bcea

Cost-Effectiveness Efficiency Frontier (CEEF) 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 = "topright",
  start.from.origins = TRUE,
  threshold = NULL,
  flip = FALSE,
  dominance = TRUE,
  print.summary = TRUE,

```

```

graph = options("bcea.graph"),
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?
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 three options "base", "ggplot2" or "plotly". Default value is "base". This is set globally upon loading BCEA and can be modified for instance by using options("bcea.graph"="gg"), or options("bcea.graph"="plotly"). Partial matching still applies (so gg or pl also work). Not all plotting functions have a "plotly" implementation, yet – see help for the specifics.
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.

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 (2013). *Bayesian Methods in Health Economics*. CRC.
 IQWiG (2009). "General Methods for the Assessment of the Relation of Benefits to Cost, Version 1.0." Institute for Quality and Efficiency in Health Care (IQWiG).

See Also

[bcea\(\)](#)

Examples

```
## create the bcea object m for the smoking cessation example
data(Smoking)
m <- bcea(eff, cost, 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_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_plotly(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 = "topleft",
  graph = options("bcea.graph"),
  ...
)

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. For ggplot2 and plotly can also provide a list of arguments for more options (see below).
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 three options "base", "ggplot2" or "plotly". Default value is "base". This is set globally upon loading BCEA and can be modified for instance by using options("bcea.graph"="gg"), or options("bcea.graph"="plotly"). Partial matching still applies (so gg or pl also work). Not all plotting functions have a "plotly" implementation, yet – see help for the specifics.

...

If `graph = "ggplot2"` and a named theme object is supplied, it will be passed to the **ggplot2** object. The usual `ggplot2` syntax is used. Additional graphical arguments:

- `label.pos = FALSE`: Places the willingness-to-pay label in a different position at the bottom of the graph. Applies to base and **ggplot2** only (no label in **plotly**).
- `line = list(color)`: A colour specifying the colour of the willingness-to-pay line (not available in the **plotly** version).
- `point = list(color)`: A vector of colours specifying the colour(s) associated with the cloud of points. Should be of length 1 or equal to the number of comparisons.
- `point = list(size)`: A vector specifying the size(s) of the points. Should be of length 1 or equal to the number of comparisons.
- `point = list(shape)`: A vector specifying the shape(s) of the points. Should be of length 1 or equal to the number of comparisons.
- `icer = list(color)`: A vector of colours specifying the colour(s) of the ICER points. Should be of length 1 or equal to the number of comparisons.
- `icer = list(size)`: A vector specifying the size(s) of the ICER points. Should be of length 1 or equal to the number of comparisons.
- `area_include`: Logical. Include or exclude the cost-effectiveness acceptability area (default is TRUE).
- `wtp = list(value)`: Equivalent to using `wtp = value` but for when multiple arguments are passed in list form.
- `area = list(color)`: A colour specifying the colour of the cost-effectiveness acceptability area.
- `wtp = list(color)`: A colour specifying the colour of the willingness-to-pay text.
- `wtp = list(size)`: A value specifying the size of the willingness-to-pay text.
- `wtp = list(x=..., y=...)`: Values specifying the x and y coordinates of the willingness-to-pay text (not displayed in the **plotly** version).
- `currency`: Currency prefix to cost differential values. Applies to **ggplot2** only.
- `icer_annot`: Annotates each ICER point with a text label. Applies to **ggplot2** only.

Details

In the **plotly** version, colors can also be specified as `rgba` colours using either the `[plotly]toRGB` function or a `rgba` colour string, e.g. `'rgb(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 aP (2011). “Probabilistic sensitivity analysis in health economics.” *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

See Also

[bcea\(\)](#), [ceplane_plot_graph\(\)](#)

Examples

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

m <- bcea(eff, cost, 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 = list(size = 2), graph = "ggplot2")
}

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

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, pos_legend, graph_params, ...)

ceplane_plot_base(he, ...)

## S3 method for class 'bcea'
ceplane_plot_ggplot(he, 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.
pos_legend	Legend position
graph_params	Graph parameters in ggplot2 format
...	Additional arguments
wtp	Willingness to pay threshold; default 25,000

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(eff, cost)
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)

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(he, cost, 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(he, cost)

ceplane.plot(he, graph = "ggplot2")
ceplane.plot(he, wtp=10000, graph = "ggplot2",
             point = list(colors = "blue", sizes = 2),
             area = list(col = "springgreen3"))

data(Smoking)
he <- bcea(he, cost, 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 ($r \rightarrow 0$) 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 parameter 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**: A vector containing all the possible values for the parameter of risk aversion r .

Author(s)

Gianluca Baio

References

Baio G, Dawid aP (2011). “Probabilistic sensitivity analysis in health economics.” *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

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=eff,c=cost,      # defines the variables of
          ref=2,            # effectiveness and cost
          interventions=treats, # selects the 2nd row of (e, c)
                                # as containing the reference intervention
                                # 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

# Can plot the resulting output using
plot(m)

```

CEriskav_plot_graph	<i>Cost-effectiveness Plot Including a Parameter of Risk Aversion</i>
---------------------	---

Description

Choice of base R, **ggplot2** or **plotly**.

Usage

```

CEriskav_plot_base(he, pos_legend)

CEriskav_plot_ggplot(he, pos_legend)

CEriskav_plot_plotly(he, pos_legend)

```

Arguments

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

compute_CEAC	<i>Compute Cost-Effectiveness Acceptability Curve</i>
--------------	---

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	<i>Compute Cost-Effectiveness Acceptability Frontier</i>
--------------	--

Description

Compute Cost-Effectiveness Acceptability Frontier

Usage

compute_ceaf(p_best_interv)

Arguments

p_best_interv Probability of being best intervention

compute_EIB	<i>Compute Expected Incremental Benefit</i>
-------------	---

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:

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

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	<i>Calculate Credible Intervals</i>
-----------------	-------------------------------------

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	<i>Compute Expected Value of Information</i>
-------------	--

Description

Compute Expected Value of Information

Usage

```
compute_EVI(ol)
```

Arguments

ol	Opportunity loss
----	------------------

Value

EVI

compute_IB	<i>Compute Incremental Benefit</i>
------------	------------------------------------

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	<i>Compute Incremental Cost-Effectiveness Ratio</i>
--------------	---

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	<i>Compute k^*</i>
---------------	---------------------------------

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	<i>Compute Opportunity Loss</i>
------------	---------------------------------

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,

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

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 $OL(\theta) = 0$ 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_p_optimal_best
Compute Probability Optimal Intervention Best

Description

Compute Probability Optimal Intervention Best

Usage

```
compute_p_optimal_best(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

<code>df_ce</code>	Cost-effectiveness dataframe
<code>k</code>	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

<code>he</code>	A bcea object containing the results of the Bayesian modelling and the economic evaluation.
<code>value</code>	Mixture weights

compute_Ustar	<i>Compute Ustar Statistic</i>
---------------	--------------------------------

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	<i>Compute Value of Information</i>
------------	-------------------------------------

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:

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

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\(\)](#)

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,
  pos = "topleft",
  graph = options("bcea.graph"),
  comparison = NULL,
  ...
)

contour(he, ...)
```

Arguments

he	A bcea object containing the results of the Bayesian modelling and the economic evaluation.
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". This is set globally upon loading BCEA and can be modified for instance by using options("bcea.graph"="gg"), or options("bcea.graph"="plotly"). Partial matching still applies (so gg or pl also work). Not all plotting functions have a "plotly" implementation, yet – see help for the specifics.

- 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`).
- ... Additional graphical arguments. The usual `ggplot2` syntax is used regardless of graph type.
- `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`
 - `scale`: Scales the plot as a function of the observed standard deviation.
 - `levels`: Numeric vector of levels at which to draw contour lines. Quantiles $0 < p < 1$.
 - `nlevels`: Number of levels to be plotted in the contour.

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 aP (2011). "Probabilistic sensitivity analysis in health economics." *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

See Also

[bcea\(\)](#), [ceplane.plot\(\)](#), [contour2\(\)](#)

Examples

```
data(Vaccine)

# run the health economic evaluation using BCEA
m <- bcea(e=eff,
          c=cost,          # 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
                                # maximum value possible for the willingness
                                # to pay threshold; implies that k is chosen
                                # in a grid from the interval (0,Kmax)
                                # plots the results
                                )

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

contour(m,                        # uses the results of the economic evaluation
                                # (a "bcea" object)
      comparison=1, # if more than 2 interventions, selects the
                                # pairwise comparison
      nlevels=10,  # 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=1,     # 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,
  comparison = NULL,
  wtp = 25000,
  graph = options("bcea.graph"),
  pos = "topleft",
  ...
)

contour2(he, ...)

```

Arguments

he	A bcea object containing the results of the Bayesian modelling and the economic evaluation.
comparison	The comparison being plotted. Default to NULL. 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)).
wtp	The selected value of the willingness-to-pay. Default is 25000.
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". This is set globally upon loading BCEA and can be modified for instance by using options("bcea.graph"="gg"), or options("bcea.graph"="plotly"). Partial matching still applies (so gg or pl also work). Not all plotting functions have a "plotly" implementation, yet – see help for the specifics.
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.
...	Arguments to be passed to <code>ceplane.plot()</code> . 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 aP (2011). "Probabilistic sensitivity analysis in health economics." *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.
- Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

See Also

`bcea()`, `ceplane.plot()`, `contour()`

Examples

```
## create the bcea object m for the smoking cessation example
data(Smoking)
m <- bcea(eff, cost, 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")

## vaccination example
data(Vaccine)
treats = c("Status quo", "Vaccination")
m <- bcea(eff, cost, ref = 2, interventions = treats, Kmax = 50000)
contour2(m)
contour2(m, wtp = 100)
```

contour_graph

*Contour Cost-Effectiveness Plane***Description**

Choice of base R, **ggplot2**.

Usage

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

contour_ggplot(he, pos_legend, graph_params, ...)

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

Arguments

he	A bcea object containing the results of the Bayesian modelling and the economic evaluation.
pos_legend	Legend position
graph_params	Plot parameters; list
...	Additional arguments

Value

For **plotly** returns a plot in the Viewer

See Also

[contour\(\)](#)

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	An object containing suitable simulations for the model parameters. It can be a <code>rjags</code> or <code>bugs</code> object (obtained directly by running <code>R2jags</code> , <code>rjags</code> , <code>R2openBUGS</code> or <code>R2winBUGS</code>). For those objects, which are in the class <code>rjags</code> or <code>bugs</code> , the user
--------	---

can simply pass them to `createInputs()`, which will now know how to process them. If the user has run their model using `rstan`, then they should pre-process the output using the `rstan::extract()` function; so if `rstan` is used to run a Bayesian model and save the results to, say, the object `fit`, then the user should first create another object `tmp=rstan::extract(fit) |> as.data.frame()` and then pass this onto `createInputs()`. Finally, this function handles inputs from data-frames of vectors directly.

`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

<code>mat</code>	Data.frame containing all the simulations for all the monitored parameters
<code>parameters</code>	Character vectors of the names of all the monitored parameters

Author(s)

Gianluca Baio, Anna Heath and Mark Strong

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 = "bottomright",
  size = NULL,
  plot.cri = NULL,
  graph = options("bcea.graph"),
  ...
)

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., <code>comparison=c(1,3)</code> or <code>comparison=2</code>).
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 (<code>bottom top</code>)(<code>right left</code>) for base graphics and <code>bottom top left right</code> 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 <code>graph="ggplot2"</code> , 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% include them for multiple comparisons. Setting <code>plot.cri=TRUE</code> or <code>plot.cri=FALSE</code> 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". This is set globally upon loading BCEA and can be modified for instance by using <code>options("bcea.graph"="gg")</code> , or <code>options("bcea.graph"="plotly")</code> . Partial matching still applies (so gg or pl also work). Not all plotting functions have a "plotly" implementation, yet – see help for the specifics.
...	If <code>graph="ggplot2"</code> and a named theme object is supplied, it will be added to the ggplot object. Additional arguments: • <code>alpha</code> can be used to set the CrI level when <code>plot.cri=TRUE</code>, with a default value of <code>alpha=0.05</code>. • <code>cri.quantile</code> controls the the method of calculation of the credible intervals. The default value <code>cri.quantile=TRUE</code> defines the CrI as the interval between the $\alpha/2$-th and $1-\alpha/2$-th quantiles of the IB distribution. Setting <code>cri.quantile=FALSE</code> will use a normal approximation on the IB distribution to calculate the intervals. • <code>line = list(color)</code>: specifies the line colour(s) - all graph types. • <code>line = list(type)</code>: specifies the line type(s) as lty numeric values - all graph types. • <code>area_include</code>: include area under the EIB curve - plotly only. • <code>area_color</code>: 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 aP (2011). "Probabilistic sensitivity analysis in health economics." *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

See Also

[bcea\(\)](#), [ib.plot\(\)](#), [ceplane.plot\(\)](#)

Examples

```
data(Vaccine)

# Runs the health economic evaluation using BCEA
m <- bcea(
  e=eff,
  c=cost,
  ref=2,
  interventions=treats,
  Kmax=50000,
  plot=FALSE
)
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(eff, cost, ref = 4, interventions = treats, Kmax = 500)
eib.plot(m)
```

eib_plot_graph	<i>Expected Incremental Benefit Plot By Graph Device</i>
----------------	--

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

evi.plot.bcea	<i>Expected Value of Information (EVI) Plot</i>
---------------	---

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 aP (2011). "Probabilistic sensitivity analysis in health economics." *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

See Also

[bcea\(\)](#), [ceac.plot\(\)](#), [ceplane.plot\(\)](#)

Examples

```
data(Vaccine)
m <- bcea(
  e=eff,
  c=cost,
  ref=2,
  interventions=treats,
  Kmax=50000,
  plot=FALSE
)
evi.plot(m)

data(Smoking)
treats <- c("No intervention", "Self-help",
```

```

      "Individual counselling", "Group counselling")
m <- bcea(eff, cost, 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 <code>mixedAn</code> , a subclass of <code>bcea</code> , given as output of the call to the function <code>mixedAn()</code> .
y.limits	Range of the y-axis for the graph. The default value is <code>NULL</code> , 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 (<code>bottom top</code>)(<code>right left</code>) for base graphics and <code>bottom top left right</code> for <code>ggplot2</code> . 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 <code>FALSE</code> indicating to use the default position and <code>TRUE</code> 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 <code>"base"</code> or <code>"ggplot2"</code> . Default value is <code>"base"</code> . The <code>"plotly"</code> option is not implemented for this particular graph.
...	Arguments to be passed to methods, such as graphical parameters (see <code>par()</code>).

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, Russo P (2009). “A decision-theoretic framework for the application of cost-effectiveness analysis in regulatory processes.” *Pharmacoeconomics*, **27**(8), 5–16. ISSN 20356137, doi:10.1007/bf03320526.

Baio G, Dawid aP (2011). “Probabilistic sensitivity analysis in health economics.” *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

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=eff, c=cost,      # 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	<i>Expected Value of Information Plot By Graph Device</i>
----------------	---

Description

Choice of base R, **ggplot2** or **plotly**.

Usage

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

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

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

Arguments

he	A bcea object containing the results of the Bayesian modelling and the economic evaluation.
data.psa	Data
plot_aes	Aesthetic parameters
plot_annotatations	Plot parameters

evppi	<i>Expected Value of Perfect Partial Information (EVPPI) for Selected Parameters</i>
-------	--

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, ...)

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

## S3 method for class 'bcea'
evppi(
  he,
  param_idx = NULL,
```

```

    input,
    N = NULL,
    plot = FALSE,
    residuals = TRUE,
    method = NULL,
    ...
)

```

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. Details of the methods to compute the EVPPI and their additional arguments are: • For single-parameter: – Generalized additive model (GAM) (default). – The method of Strong & Oakley use method as string so. The user <i>needs</i> to also specify the number of "blocks" (e.g. n.blocks=20). Note that the multi-parameter version for this method has been deprecated. – The method of Sadatsafavi <i>et al.</i> where method takes as value a string of either sad or sal. It is then 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. Note that the multi-parameter version for this method has been deprecated. • For multi-parameter: – INLA/SPDE (default). – Gaussian process regression with method of gp.
method	Character string to select which method to use. The default methods are recommended. However, it is possible (mainly for backward compatibility) to use different methods.

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 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

An object of class `evppi`, containing the following components:

- **evppi**: The computed values of `evppi` for all values of the parameter of willingness to pay.
- **index**: A numerical vector with the indices of the parameters for which the EVPPI was calculated.
- **k**: A vector of values for the willingness to pay.
- **evi**: A 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).
- **time**: The 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 JE, Brennan A (2014). "Estimating Multiparameter Partial Expected Value of Perfect Information from a Probabilistic Sensitivity Analysis Sample : A Nonparametric Regression Approach." *Medical Decision Making*, 311–326. doi:10.1177/0272989X13505910.
- 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 Heal.*, 16(2), 438–448. ISSN 10983015, doi:10.1016/j.jval.2012.10.018, <http://dx.doi.org/10.1016/j.jval.2012.10.018>.
- Baio G (2013). *Bayesian Methods in Health Economics*. CRC.
- Baio, Gianluca, Berardi, Andrea, Heath A (2017). *Bayesian Cost-Effectiveness Analysis with the R package BCEA*. Springer International Publishing. <https://link.springer.com/book/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." *Stat. Med.*, 35(23), 4264–4280. ISSN 0277-6715, doi:10.1002/sim.6983, 1504.05436, <https://pubmed.ncbi.nlm.nih.gov/27189534/>.

See Also

`bcea()`, `plot.evppi()`

Examples

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

## Not run:
# Load the post-processed results of the MCMC simulation model
# original JAGS output is can be downloaded from here
# https://gianluca.statistica.it/books/bcea/code/vaccine.RData

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_mat)

# explicitly use BCEA package namespace to avoid voi package conflict
EVPPI <- BCEA::evppi(m, c("beta.1.", "beta.2."), inp$mat)

plot(EVPPI)

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

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

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

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

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

# plot results
if (require("INLA")) 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")

if (require("INLA")) {
  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(eff, cost, ref = 4, interventions = treats, Kmax = 500)
  inp <- createInputs(smoking_output)
  EVPPI <- BCEA::evppi(m, c(2,3), inp$mat, h.value = 0.0000005)
  plot(EVPPI)

  ## End(Not run)

```

evppi_plot_graph	<i>Plot Expected Value of Partial Information With Respect to a Set of Parameters</i>
------------------	---

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

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 = options("bcea.graph"),
  ...
)

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 three options "base", "ggplot2" or "plotly". Default value is "base". This is set globally upon loading BCEA and can be modified for instance by using options("bcea.graph"="gg"), or options("bcea.graph"="plotly"). Partial matching still applies (so gg or pl also work). Not all plotting functions have a "plotly" implementation, yet – see help for the specifics.
...	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 aP (2011). "Probabilistic sensitivity analysis in health economics." *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

See Also

[bcea\(\)](#), [ceplane.plot\(\)](#)

Examples

```
data("Vaccine")
he <- BCEA::bcea(eff, cost)
ib.plot(he)
```

ib_plot_base

IB plot base R version

Description

Choice of base R, ggplot2, plotly

Usage

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

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

ib_plot_plotly(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	<i>Information-Rank Plot for bcea Class</i>
----------------	---

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 = NULL,
  howManyPars = NA,
  graph = options("bcea.graph"),
  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.

	• <code>mat</code> = 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.
<code>wtp</code>	A value of the <code>wtp</code> for which the analysis should be performed. If not specified then the break-even point for the current model will be used.
<code>howManyPars</code>	Optional maximum number of parameters to be included in the bar plot. Includes all parameters by default.
<code>graph</code>	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". This is set globally upon loading BCEA and can be modified for instance by using <code>options("bcea.graph"="gg")</code> , or <code>options("bcea.graph"="plotly")</code> . Partial matching still applies (so <code>gg</code> or <code>pl</code> also work). Not all plotting functions have a "plotly" implementation, yet – see help for the specifics.
<code>rel</code>	Logical argument that specifies whether the ratio of EVPPI to EVPI (<code>rel</code> = TRUE, default) or the absolute value of the EVPPI should be used for the analysis.
<code>...</code>	Additional options. These include graphical parameters that the user can specify: • <code>xlim</code> = limits of the x-axis; <code>ca</code> = font size for the axis label (default = 0.7 of full size). • <code>cn</code> = font size for the parameter names vector (default = 0.7 of full size) - base graphics only. • <code>mai</code> = margins of the graph (default = <code>c(1.36, 1.5, 1,1)</code>) - 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 aP (2011). "Probabilistic sensitivity analysis in health economics." *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.
- Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

See Also

[bcea\(\)](#), [evppi\(\)](#)

Examples

```
## Not run:
# Load the post-processed results of the MCMC simulation model
# original JAGS output is can be downloaded from here
# https://gianluca.statistica.it/books/bcea/code/vaccine.RData

data("Vaccine")
m <- bcea(eff, cost)
inp <- createInputs(vaccine_mat)
info.rank(m, inp)

info.rank(m, inp, graph = "base")
info.rank(m, inp, graph = "plotly")
info.rank(m, inp, graph = "ggplot2")

## End(Not run)
```

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.

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:

<code>Ubar</code>	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
<code>OL.star</code>	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
<code>evi.star</code>	The Expected Value of Information for the mixed strategy, for each value of the discrete grid approximation of the willingness to pay parameter
<code>mkt.shares</code>	The vector of market shares associated with each available intervention

Author(s)

Gianluca Baio

References

Baio G, Russo P (2009). "A decision-theoretic framework for the application of cost-effectiveness analysis in regulatory processes." *Pharmacoeconomics*, **27**(8), 5–16. ISSN 20356137, doi:10.1007/bf03320526.

Baio G, Dawid aP (2011). "Probabilistic sensitivity analysis in health economics." *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

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=eff, c=cost,      # 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=eff, c=cost,      # 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)

```

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\(\)](#)

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 = options("bcea.graph"),
  ...
)
```

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 three options "base", "ggplot2" or "plotly". Default value is "base". This is set globally upon loading BCEA and can be modified for instance by using options("bcea.graph"="gg"), or options("bcea.graph"="plotly"). Partial matching still applies (so gg or pl also work). Not all plotting functions have a "plotly" implementation, yet – see help for the specifics.
...	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 aP (2011). “Probabilistic sensitivity analysis in health economics.” *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

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=eff, c=cost,      # 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
      ICER_size = 1.5,           # extra options modifies the mean point size
      text = list(size=rel(1.25)) # modifies the text size
    )
}

```

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 = "topright", graph = options("bcea.graph"), ...)

```

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 three options "base", "ggplot2" or "plotly". Default value is "base". This is set globally upon loading BCEA and can be modified for instance by using options("bcea.graph"="gg"), or options("bcea.graph"="plotly"). Partial matching still applies (so gg or pl also work). Not all plotting functions have a "plotly" implementation, yet – see help for the specifics.
...	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 aP (2011). "Probabilistic sensitivity analysis in health economics." *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

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=eff, c=cost,      # 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, e.g.:
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	<i>Plot Expected Value of Partial Information With Respect to a Set of Parameters</i>
------------	---

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 <code>evppi()</code> .
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". The "plotly" option is not implemented for this particular graph.
col	Sets the colour for the lines depicted in the graph.
...	Arguments to be passed to methods, such as graphical parameters (see <code>par()</code>).

Value

Plot with base R or **ggplot2**.

Author(s)

Gianluca Baio, Andrea Berardi

References

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

See Also

[bcea\(\)](#), [evppi\(\)](#)

Examples

```
## Not run:
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_mat)

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

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

  x1 <- BCEA::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)
```

print.bcea	<i>bcea Print Method</i>
------------	--------------------------

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(eff, cost)
```

setComparisons	<i>Set Comparisons Group</i>
----------------	------------------------------

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 aP (2011). “Probabilistic sensitivity analysis in health economics.” *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

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=eff,          # defines the variables of
          c=cost,         # 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
          wtp=25000)      # selects the particular value for k
```

Smoking

Smoking Cessation Cost-Effectiveness Data

Description

This data set contains the results of a Bayesian analysis modeling the clinical outputs and costs for an economic evaluation of four different smoking cessation interventions.

Format

A list containing the variables for the cost-effectiveness analysis:

cost A matrix of 500 simulations from the posterior distribution of the overall costs for the four strategies.

data A dataset with characteristics of smokers in the UK population.

eff A matrix of 500 simulations from the posterior distribution of the clinical benefits for the four strategies.

life.years A matrix of 500 simulations from the posterior distribution of the life years gained with each strategy.

pi_post A matrix of 500 simulations from the posterior distribution of the probability of smoking cessation with each strategy.

smoking A data frame with inputs for the network meta-analysis, containing: nobs (record ID), s (study ID), i (intervention ID), r_i (number of patients who quit), n_i (total patients in arm), and b_i (reference intervention for the study).

smoking_output A matrix of results from the network meta-analysis model run on the smoking object.

treats A character vector of labels for the four strategies.

Source

Effectiveness data adapted from Hasselblad V. (1998). "Meta-analysis of Multitreatment Studies". *Medical Decision Making*, 18:37-43.

Cost and population data adapted from various sources:

- Taylor, D.H. Jr, et al. (2002). "Benefits of smoking cessation on longevity". *American Journal of Public Health*, 92(6).
- Action on Smoking and Health (ASH) (2013). "ASH fact sheet on smoking statistics". <https://ash.org/wp-content/uploads/2014/05/ASH-Annual-Report-2014.pdf>.
- Flack, S., et al. (2007). "Cost-effectiveness of interventions for smoking cessation". *York Health Economics Consortium*.
- McGhan, W.F.D., and Smith, M. (1996). "Pharmacoeconomic analysis of smoking-cessation interventions". *American Journal of Health-System Pharmacy*, 53:45-52.

References

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

statins_base

Statins Evidence-Synthesis Data (Base Model)

Description

This data set contains the results of a Bayesian analysis used to model the effectiveness of various statins. The analysis is based on the simplest model, using vague priors.

Format

A BUGS object containing the simulations for the evidence synthesis model.

Source

A systematic review and economic evaluation of statins for the prevention of coronary events. Ward 2007.

References

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

Examples

```
data(statins_base)
lapply(statins_base$sims.list, summary)
```

statins_HC

*Statins Evidence-Synthesis Data (Robust Model)***Description**

This data set contains the results of a Bayesian analysis used to model the effectiveness of various statins. The analysis uses robust Half-Cauchy priors for the structured effects standard deviations.

Format

A BUGS object containing the simulations for the evidence synthesis model.

Source

A systematic review and economic evaluation of statins for the prevention of coronary events. Ward 2007.

References

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

Examples

```
data(statins_HC)
lapply(statins_HC$sims.list, summary)
```

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,
  w = NULL
)
```

Arguments

models	A (possibly named) 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 $n_{sim} \times n_{ints}$ simulations for each model)
cost	A list containing the measure of costs computed from the various models (one matrix with $n_{sim} \times n_{ints}$ simulations for each model)
ref	Which intervention is considered to be the reference strategy. The default value <code>ref=1</code> 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 50000. The willingness to pay is then approximated on a discrete grid in the interval $[0, Kmax]$. The grid is equal to k if the parameter is given, or composed of 501 elements if $k=NULL$ (the default)
plot	A logical value indicating whether the function should produce the summary plot or not
w	A vector of weights. By default it's NULL to indicate that the function will calculate the model weights based on DIC and the individual model fit. This behaviour can be overridden by passing a vector w , for instance based on expert opinion

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 $n_{sim} \times n_{ints}$ simulations for each model) cost is a list containing the measure of costs computed from the various models (one matrix with $n_{sim} \times n_{ints}$ simulations for each model).

Value

List object of bcea object, model weights and DIC

Author(s)

Gianluca Baio

References

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

See Also

[bcea\(\)](#)

Examples

```
# load sample jags output
data("statins_base")
data("statins_HC")

interventions <- c("Atorvastatin", "Fluvastatin",
                  "Lovastatin", "Pravastatin",
                  "Rosuvastatin", "Simvastatin")

m1 <- bcea(eff = statins_base$sims.list$effect,
          cost = statins_base$sims.list$cost.tot,
          ref = 1, interventions = interventions)

m2 <- bcea(eff = statins_HC$sims.list$effect,
          cost = statins_HC$sims.list$cost.tot,
          ref = 1, interventions = interventions)

models <- list("Base"=statins_base, "Half Cauchy"=statins_HC)

effects <- list(statins_base$sims.list$effect,
               statins_HC$sims.list$effect)
costs <- list(statins_base$sims.list$cost.tot,
              statins_HC$sims.list$cost.tot)

## Not run:
m3 <- struct.psa(models, effects, costs,
                 ref = 1, interventions = interventions)

## End(Not run)
```

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 aP (2011). “Probabilistic sensitivity analysis in health economics.” *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

See Also

[bcea\(\)](#)

Examples

```
data(Vaccine)

he <- bcea(eff, cost, 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 chosen 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, Russo P (2009). “A decision-theoretic framework for the application of cost-effectiveness analysis in regulatory processes.” *Pharmacoeconomics*, **27**(8), 5–16. ISSN 20356137, doi:10.1007/bf03320526.

Baio G, Dawid aP (2011). “Probabilistic sensitivity analysis in health economics.” *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

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=eff, c=cost,      # 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)

```

summary.pairwise	<i>Summary Method for Objects of Class pairwise</i>
------------------	---

Description

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

Usage

```

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

```

Arguments

object	A pairwise 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 aP (2011). "Probabilistic sensitivity analysis in health economics." *Stat. Methods Med. Res.*, 1–20. ISSN 1477-0334, doi:10.1177/0962280211419832, <https://pubmed.ncbi.nlm.nih.gov/21930515/>.

Baio G (2013). *Bayesian Methods in Health Economics*. CRC.

See Also

[bcea\(\) multi.ce\(\)](#)

Examples

```
data(Vaccine)
he <- bcea(eff, cost, interventions = treats, ref = 2)
he_multi <- multi.ce(he)
summary(he_multi)
```

tabulate_means	<i>Calculate Dataset For ICERs From bcea Object</i>
----------------	---

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

Examples

```
data("Smoking")
he <- BCEA::bcea(eff, cost)
tabulate_means(he)
```

Vaccine

*Influenza Vaccination Cost-Effectiveness Data***Description**

This data set contains the results of a Bayesian analysis modeling the clinical outputs and costs associated with an influenza vaccination program.

Format

A list containing variables for the influenza vaccination model:

cost A matrix of simulations of the overall costs for the two treatments.

c.pts Coordinates for plotting cost distributions.

cost.GP A matrix of simulations of the costs for GP visits.

cost.hosp A matrix of simulations of the costs for hospitalisations.

cost.otc A matrix of simulations of the costs for over-the-counter medications.

cost.time.off A matrix of simulations of the costs for time off work.

cost.time.vac A matrix of simulations of the costs for time to get vaccinated.

cost.travel A matrix of simulations of the costs for travel to get vaccinated.

cost.trt1 A matrix of simulations of the overall costs for first-line treatment.

cost.trt2 A matrix of simulations of the overall costs for second-line treatment.

cost.vac A matrix of simulations of the costs for vaccination.

eff A matrix of simulations of the clinical benefits.

e.pts Coordinates for plotting effectiveness distributions.

N The number of subjects in the reference population.

N.outcomes The number of clinical outcomes analysed.

N.resources The number of health-care resources under study.

QALYs.adv A vector of QALYs associated with adverse events.

QALYs.death A vector of QALYs associated with death.

QALYs.hosp A vector of QALYs associated with hospitalisation.

QALYs.inf A vector of QALYs associated with influenza infection.

QALYs.pne A vector of QALYs associated with pneumonia.

treats A character vector of labels for the two treatments.

vaccine_mat A matrix of simulations for the parameters in the original model.

Source

Adapted from Turner D, et al. (2006). "The cost-effectiveness of influenza vaccination of healthy adults 50-64 years of age". *Vaccine*, 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.

Index

- * **datasets**
 - Smoking, [71](#)
 - statins_base, [72](#)
 - statins_HC, [73](#)
 - Vaccine, [80](#)
- * **dplot**
 - info.rank.bcea, [54](#)
 - multi.ce, [59](#)
- * **hplot**
 - ceac.plot.bcea, [9](#)
 - ceaf.plot.pairwise, [11](#)
 - ceplane.plot.bcea, [17](#)
 - ceplane_plot_graph, [19](#)
 - contour.bcea, [33](#)
 - contour2.bcea, [35](#)
 - eib.plot.bcea, [39](#)
 - evi.plot.bcea, [42](#)
 - ib.plot.bcea, [52](#)
 - multi.ce, [59](#)
 - plot.bcea, [61](#)
 - plot.CEriskav, [63](#)
 - plot.evppi, [65](#)
- * **manip**
 - bcea, [3](#)
- * **models**
 - info.rank.bcea, [54](#)
- * **print**
 - print.bcea, [67](#)
 - sim_table, [70](#)
 - summary.bcea, [75](#)
 - summary.mixedAn, [76](#)
 - summary.pairwise, [78](#)
- bcea, [3](#)
- bcea(), [10](#), [12](#), [15](#), [19](#), [23](#), [34](#), [36](#), [39](#), [41](#), [43](#),
[45](#), [50](#), [53](#), [55](#), [58–60](#), [62](#), [64](#), [66](#), [70](#),
[74](#), [76–78](#)
- best_interv_given_k, [8](#)
- c.pts (Vaccine), [80](#)
- ceac.plot (ceac.plot.bcea), [9](#)
- ceac.plot(), [25](#), [26](#), [29](#), [43](#), [62](#)
- ceac.plot.bcea, [9](#)
- ceaf.plot (ceaf.plot.pairwise), [11](#)
- ceaf.plot(), [59](#)
- ceaf.plot.pairwise, [11](#)
- ceef.plot (ceef.plot.bcea), [13](#)
- ceef.plot.bcea, [13](#)
- ceef_plot_base (ceef_plot_graph), [16](#)
- ceef_plot_ggplot (ceef_plot_graph), [16](#)
- ceef_plot_graph, [16](#)
- ceef_plot_plotly (ceef_plot_graph), [16](#)
- ceplane.plot (ceplane.plot.bcea), [17](#)
- ceplane.plot(), [34](#), [36](#), [41](#), [43](#), [53](#), [61](#), [62](#)
- ceplane.plot.bcea, [17](#)
- ceplane_plot_base (ceplane_plot_graph),
[19](#)
- ceplane_plot_ggplot
(ceplane_plot_graph), [19](#)
- ceplane_plot_graph, [19](#)
- ceplane_plot_graph(), [19](#)
- ceplane_plot_plotly
(ceplane_plot_graph), [19](#)
- CEriskav (CEriskav_assign), [22](#)
- CEriskav(), [63](#), [64](#)
- CEriskav<- (CEriskav_assign), [22](#)
- CEriskav_assign, [22](#)
- CEriskav_plot_base
(CEriskav_plot_graph), [24](#)
- CEriskav_plot_ggplot
(CEriskav_plot_graph), [24](#)
- CEriskav_plot_graph, [24](#)
- CEriskav_plot_plotly
(CEriskav_plot_graph), [24](#)
- compute_CEAC, [24](#)
- compute_CEAC(), [26](#)
- compute_ceaf, [25](#)
- compute_EIB, [25](#)
- compute_EIB(), [28](#)

compute_eib_cri, 26
 compute_EVI, 27
 compute_IB, 27
 compute_IB(), 26
 compute_ICER, 28
 compute_kstar, 28
 compute_ol, 29
 compute_ol(), 33
 compute_p_best_interv, 30
 compute_p_optimal_best, 30
 compute_U, 31
 compute_Ubar, 31
 compute_Ustar, 32
 compute_vi, 32
 compute_vi(), 30
 contour(contour.bcea), 33
 contour(), 36, 38
 contour.bcea, 33
 contour2(contour2.bcea), 35
 contour2(), 34
 contour2.bcea, 35
 contour_base(contour_graph), 37
 contour_ggplot(contour_graph), 37
 contour_graph, 37
 contour_plotly(contour_graph), 37
 cost(Smoking), 71
 cost.GP(Vaccine), 80
 cost.hosp(Vaccine), 80
 cost.otc(Vaccine), 80
 cost.time.off(Vaccine), 80
 cost.time.vac(Vaccine), 80
 cost.travel(Vaccine), 80
 cost.trt1(Vaccine), 80
 cost.trt2(Vaccine), 80
 cost.vac(Vaccine), 80
 createInputs(createInputs.default), 38
 createInputs.default, 38

 data(Smoking), 71

 e.pts(Vaccine), 80
 eff(Smoking), 71
 eib.plot(eib.plot.bcea), 39
 eib.plot(), 61, 62
 eib.plot.bcea, 39
 eib_plot_base(eib_plot_graph), 42
 eib_plot_ggplot(eib_plot_graph), 42
 eib_plot_graph, 42
 eib_plot_plotly(eib_plot_graph), 42

 evi.plot(evi.plot.bcea), 42
 evi.plot(), 62
 evi.plot.bcea, 42
 evi.plot.mixedAn, 44
 evi_plot_base(evi_plot_graph), 46
 evi_plot_ggplot(evi_plot_graph), 46
 evi_plot_graph, 46
 evi_plot_plotly(evi_plot_graph), 46
 evppi, 46
 evppi(), 39, 55, 65, 66
 evppi_plot_base(evppi_plot_graph), 51
 evppi_plot_ggplot(evppi_plot_graph), 51
 evppi_plot_graph, 51

 ib.plot(ib.plot.bcea), 52
 ib.plot(), 41
 ib.plot.bcea, 52
 ib_plot_base, 53
 ib_plot_ggplot(ib_plot_base), 53
 ib_plot_graph(ib_plot_base), 53
 ib_plot_plotly(ib_plot_base), 53
 info.rank(info.rank.bcea), 54
 info.rank.bcea, 54
 info_rank_base(info_rank_graph), 56
 info_rank_ggplot(info_rank_graph), 56
 info_rank_graph, 56
 info_rank_plotly(info_rank_graph), 56
 is.bcea, 57

 life.years(Smoking), 71

 mixedAn(mixedAn<-), 57
 mixedAn(), 44, 45, 76, 77
 mixedAn<-, 57
 multi.ce, 59
 multi.ce(), 11, 12, 78

 N(Vaccine), 80
 new_bcea, 60

 par(), 44, 63, 65
 pi_post(Smoking), 71
 plot.bcea, 61
 plot.bcea(), 10
 plot.CEriskav, 63
 plot.evppi, 65
 plot.evppi(), 50
 print.bcea, 67
 print.default(), 67

QALYs.adv (Vaccine), 80
QALYs.death (Vaccine), 80
QALYs.hosp (Vaccine), 80
QALYs.inf (Vaccine), 80
QALYs.pne (Vaccine), 80

setComparisons, 67
setComparisons(), 68
setComparisons<-
 (setComparisons_assign), 68
setComparisons_assign, 68
setKmax<- (setKmax_assign), 68
setKmax_assign, 68
setReferenceGroup<-
 (setReferenceGroup_assign), 69
setReferenceGroup_assign, 69
sim_table, 70
Smoking, 71
smoking (Smoking), 71
smoking_output (Smoking), 71
statins_base, 72
statins_HC, 73
struct.psa, 73
summary.bcea, 75
summary.mixedAn, 76
summary.pairwise, 78

tabulate_means, 79
treats (Smoking), 71

Vaccine, 80
vaccine_mat (Vaccine), 80