Introduction to R for Health Economics using BCEA

Gianluca Baio

Department of Statistical Science | University College London

g.baio@ucl.ac.uk
https://gianluca.statistica.it
https://egon.stats.ucl.ac.uk/research/statistics-health-economics
https://github.com/giabaio
https://github.com/StatisticsHealthEconomics
@gianlubaio

Follow our departmental social media accounts

Powered by Quarto & Alebaio

check out our departmental podcast "Random Talks" on Soundcloud!

Health Economic Modeling in R: A Hands-on Introduction
ISPOR Europe 2022, Vienna, Austria

6 November 2022

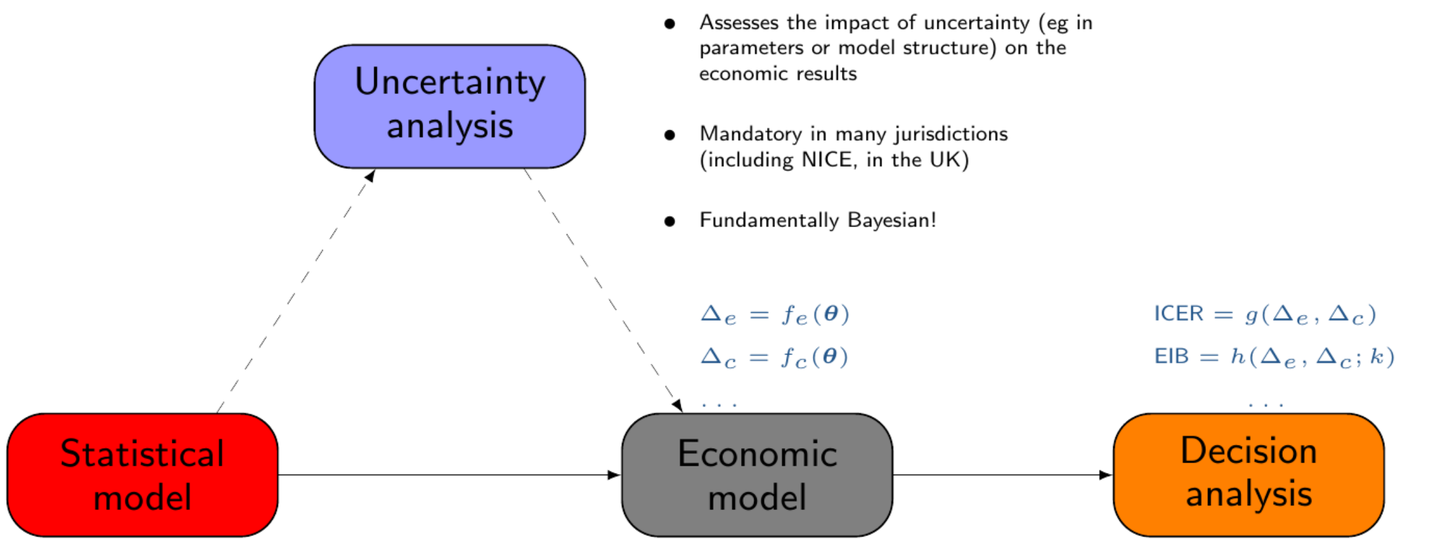
UCL Statistical Science

Disclaimer

…Just so you know what you’re about to get into… 😉

Health technology assessment (HTA)

Objective: Combine costs and benefits of a given intervention into a rational scheme for allocating resources



- Assesses the impact of uncertainty (eg in parameters or model structure) on the economic results
- Mandatory in many jurisdictions (including NICE, in the UK)
- Fundamentally Bayesian!

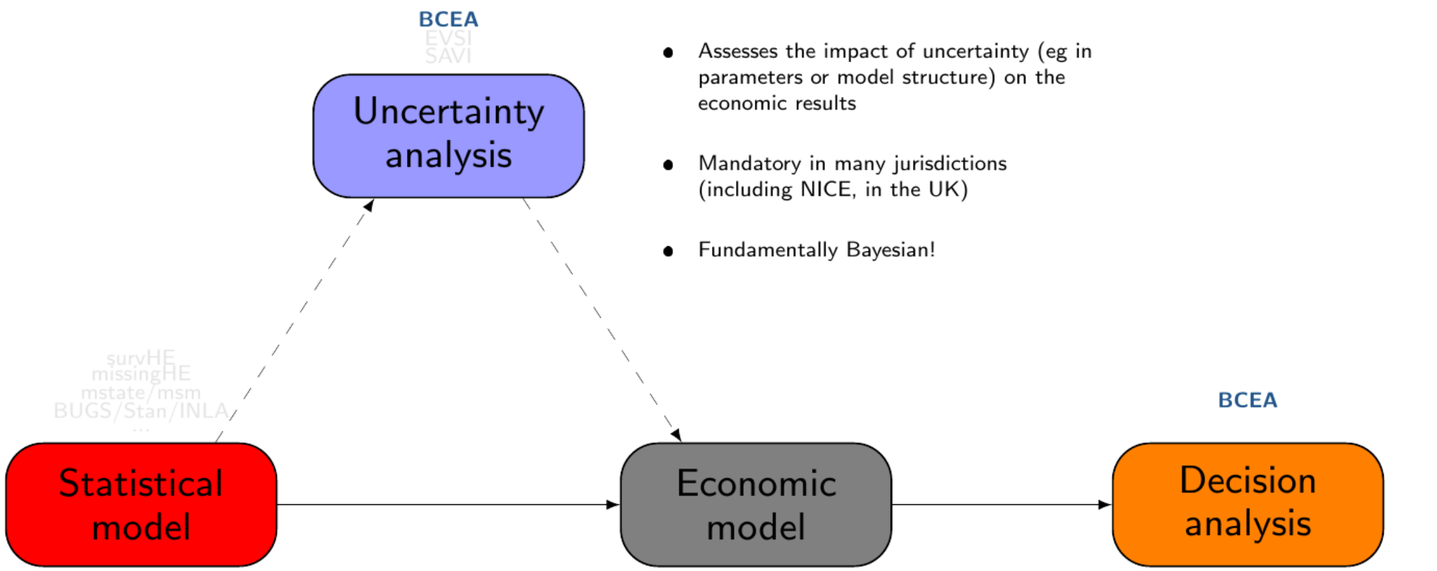
- Estimates relevant **population** parameters $\boldsymbol{\theta}$
- Varies with the type of available data (& statistical approach!)

- Combines the parameters to obtain a population average measure for costs and clinical benefits
- Varies with the type of available data & statistical model used

- Summarises the economic model by computing suitable measures of "cost-effectiveness"
- Dictates the best course of actions, given current evidence
- Standardised process

Health technology assessment (HTA)

For each module, we may need/use different/specific packages! (the "R-HTA-verse"?)



What is R?

- R is a very powerful **statistical software**
  - Specifically designed for statistical analysis
  - **Very** large community of contributors – basically you can find code/packages to do any statistical analysis you need
  - **Open source and free**

R & HTA

What is R?

- R is a very powerful **statistical software**
  - Specifically designed for statistical analysis
  - **Very** large community of contributors – basically you can find code/packages to do any statistical analysis you need
  - **Open source and free**

Why use R?

- Everything can be (and almost invariably is) scripted
- This helps with:
  - Reproducibility
  - Sharing your work with colleagues
  - Reusing templates for "similar" projects
  - "**Transparency**"!
- **Fantastic** graphical capability
  - Especially with new `tidyverse` packages (`ggplot2`)
- Generally **fit for purpose**
  - You **need** advanced tools for many (most??) of the models you do…

But…

> "*Transparency is in the eye of the beholder*"
>
> (Andy Briggs at the R-HTA workshop – October 2020)

- There **is** an entry cost
- And more importantly, the effort goes hand in hand with sophistication in the statistical modelling associated with the economic evaluation!


Totally worth it.

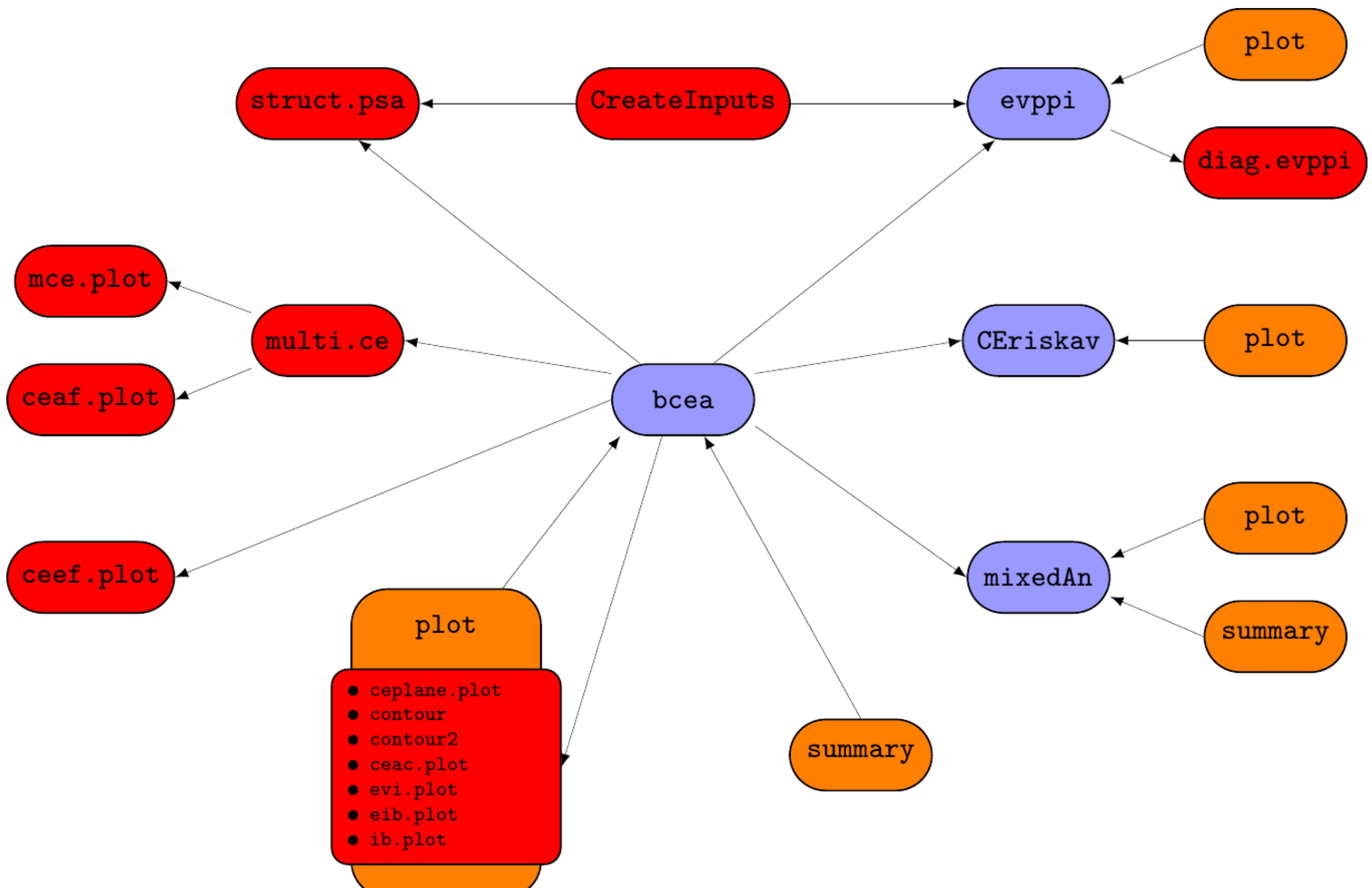
BCEA

A R package for (Bayesian) cost-effectiveness analysis

BCEA and its use directly in R are designed with these objectives in mind

1. Checking the model assumptions
   - Do we mean what we mean (eg in terms of PSA simulations)?…
   - Simulation error (especially, **but not only**, for a Bayesian approach)
2. Produce the base-case economic evaluation
   - What's the most cost-effective intervention, given current evidence?
   - Cost-effectiveness plane, Expected Incremental Benefit (as a function of k),…
3. Perform uncertainty analysis
   - Standard PSA (mandatory): Cost-effectiveness Plane, CEAC, …
   - Fairly easy (but not always used): CEAF
   - More advanced/"too difficult" (rarely used): EVP(P)I/EVSI
4. Standardised reporting
   - Graphical tools (use **excellent** R facilities)
   - Embed code in structured reports (docx/pdf)

An R package for (Bayesian) cost-effectiveness analysis



https://gianluca.statistica.it/software/bcea

https://github.com/giabaio/BCEA

Using BCEA to summarise outputs of an economic model

How does BCEA work?

- BCEA is available from CRAN
  - Current *stable* version: 2.4-2 (3 September 2022)
- But it is also under constant development in the GitHub repository
  - Current *stable* version: 2.4-2 (3 September 2022)
  - Current *development* version: 2.4-2 (ongoing development, until integrated in the next stable release)

```
# Install BCEA (only required once and needs an internet connection!).

# You can either get the "official" version from CRAN
install.packages("BCEA")

# Can also install the stable from GitHub
install.packages("remotes")                          # to install packages from GitHub
remotes::install_github("giabaio/BCEA")              # stable version (2.4.1)

# And the *development* version (from GitHub)
remotes::install_github("giabaio/BCEA",ref="devel")    # development version (2.4-1)
```

NB: The beauty of the GitHub version is that it can be updated on the fly and be immediately available for users!

How does BCEA work?

- At this point, we are ready to call the function `bcea` that runs the economic analysis, for example something like

```
treats = c("Status quo","Vaccination")
m = bcea(e=eff,c=cost,ref=2,interventions=treats,Kmax=50000)
```

- The inputs to the function are
  - `eff`: a **matrix** containing the simulations for the clinical benefits (that is $n_{\text{sim}} \times n_{\text{int}}$ values)
  - `cost`: a **matrix** containing the simulations for the costs (that is $n_{\text{sim}} \times n_{\text{int}}$ values)
  - `ref`: an indication of which intervention is to be taken as reference (default: the intervention in the first column of `eff` or `cost`)
  - `interventions`: a vector of labels for the interventions being compared
  - `Kmax`: the maximum value of k, the parameter of willingness to pay

- The output is an object `m` containing several elements

```
names(m)
```

```
 [1] "n_sim"          "n_comparators" "n_comparisons" "delta_e"       "delta_c"       "ICER"          "Kmax"          "k"             "ceac"
[10] "ib"             "eib"           "kstar"         "best"          "U"             "vi"            "Ustar"         "ol"            "evi"
[19] "ref"            "comp"          "step"          "interventions" "e"             "c"
```

How does BCEA work?

Can visualise the output in various formats (tables/graphs)

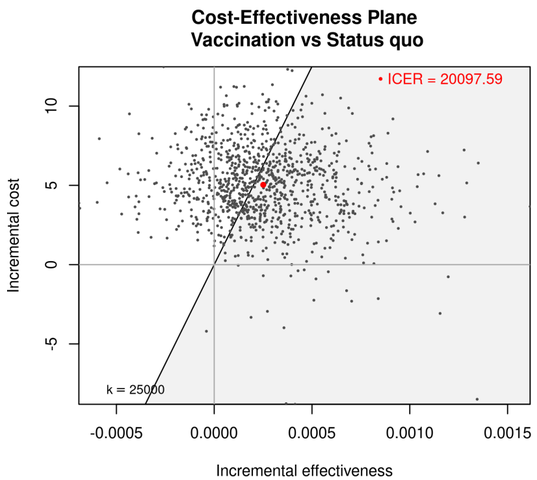
```
1 # The 'summary' "method" produces a tabular output
2 summary(m)
```

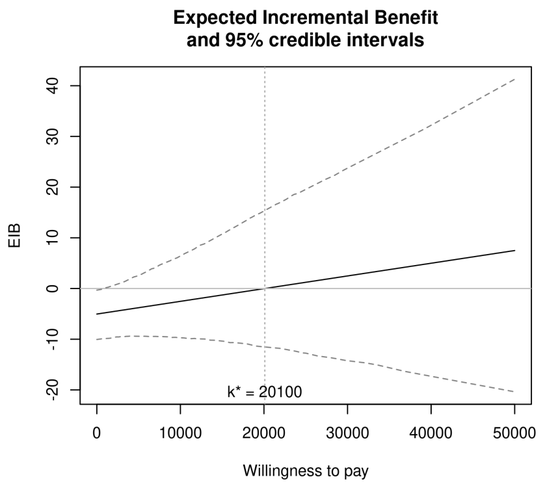
```
Cost-effectiveness analysis summary

Reference intervention:  Vaccination
Comparator intervention: Status quo

Optimal decision: choose Status quo for k < 20100 and Vaccination for k >= 20100


Analysis for willingness to pay parameter k = 25000

            Expected net benefit
Vaccination                -36.054
Status quo                 -34.826

                              EIB  CEAC  ICER
Vaccination vs Status quo 1.2284 0.529 20008
```

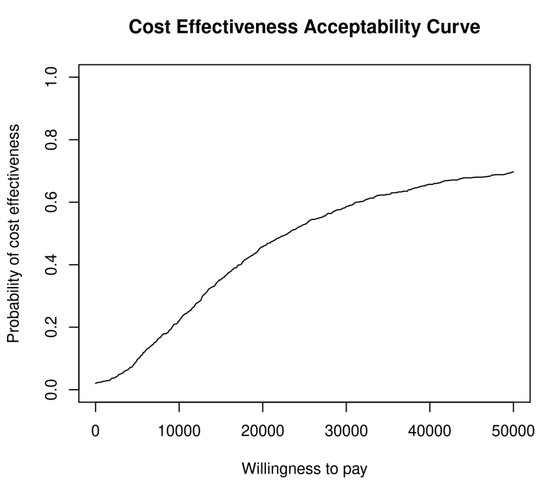
How does BCEA work?

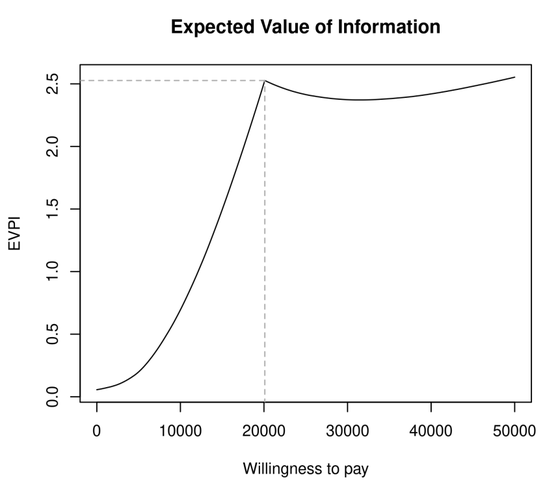
Can visualise the output in various formats (tables/graphs)

```
# The 'plot' "method" produces a *specific* version of graphical c
plot(m)
```

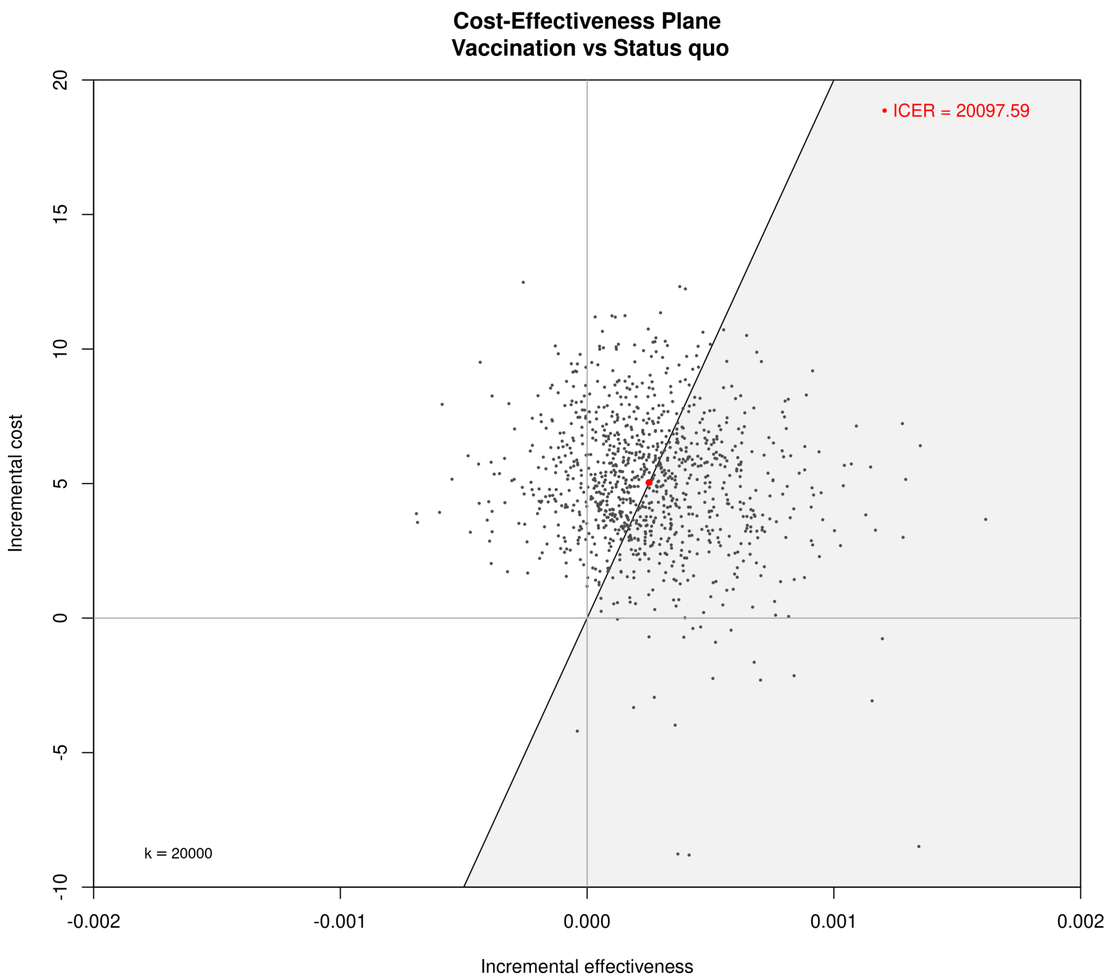






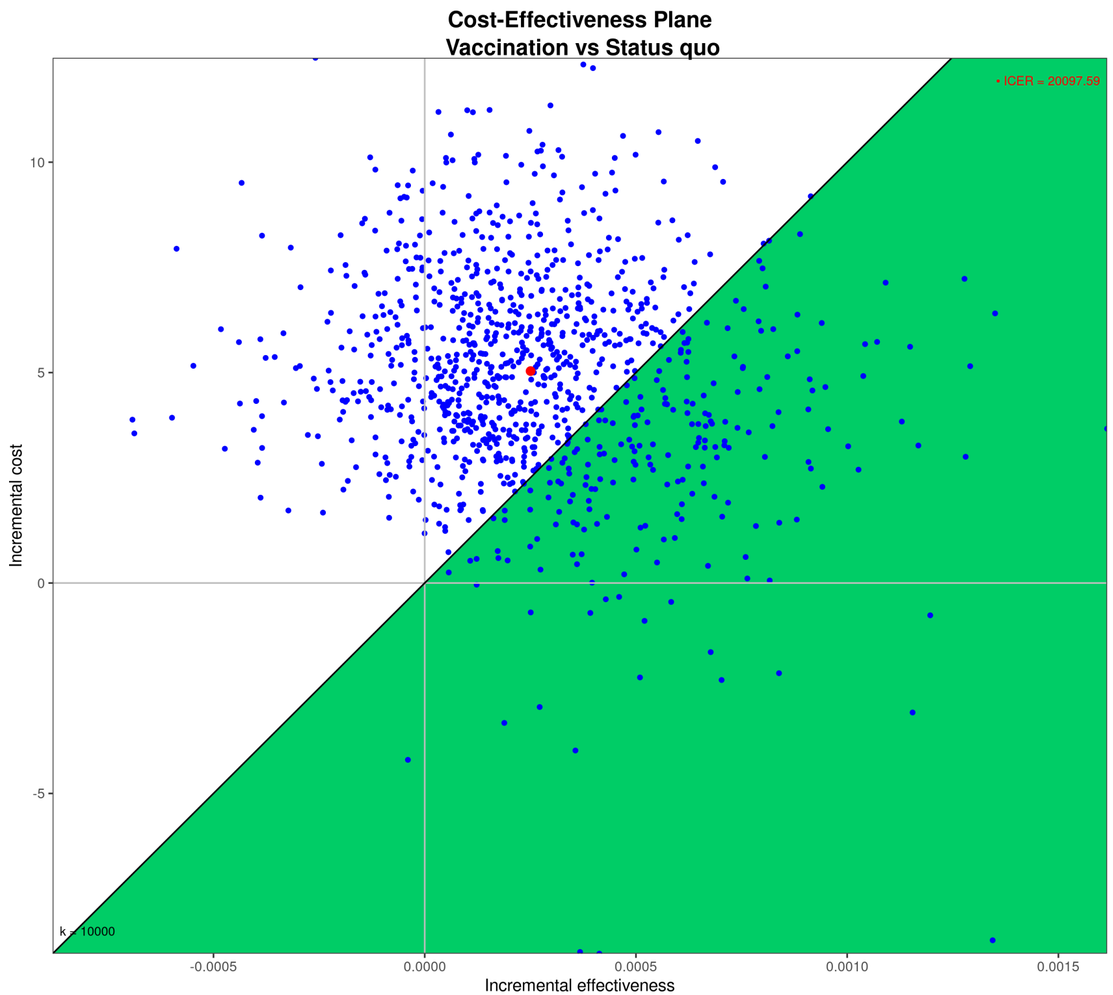



```
ceplane.plot(m,wtp=20000,xlim=c(-.002,.002),ylim=c(-10,20))
```



How does BCEA work?

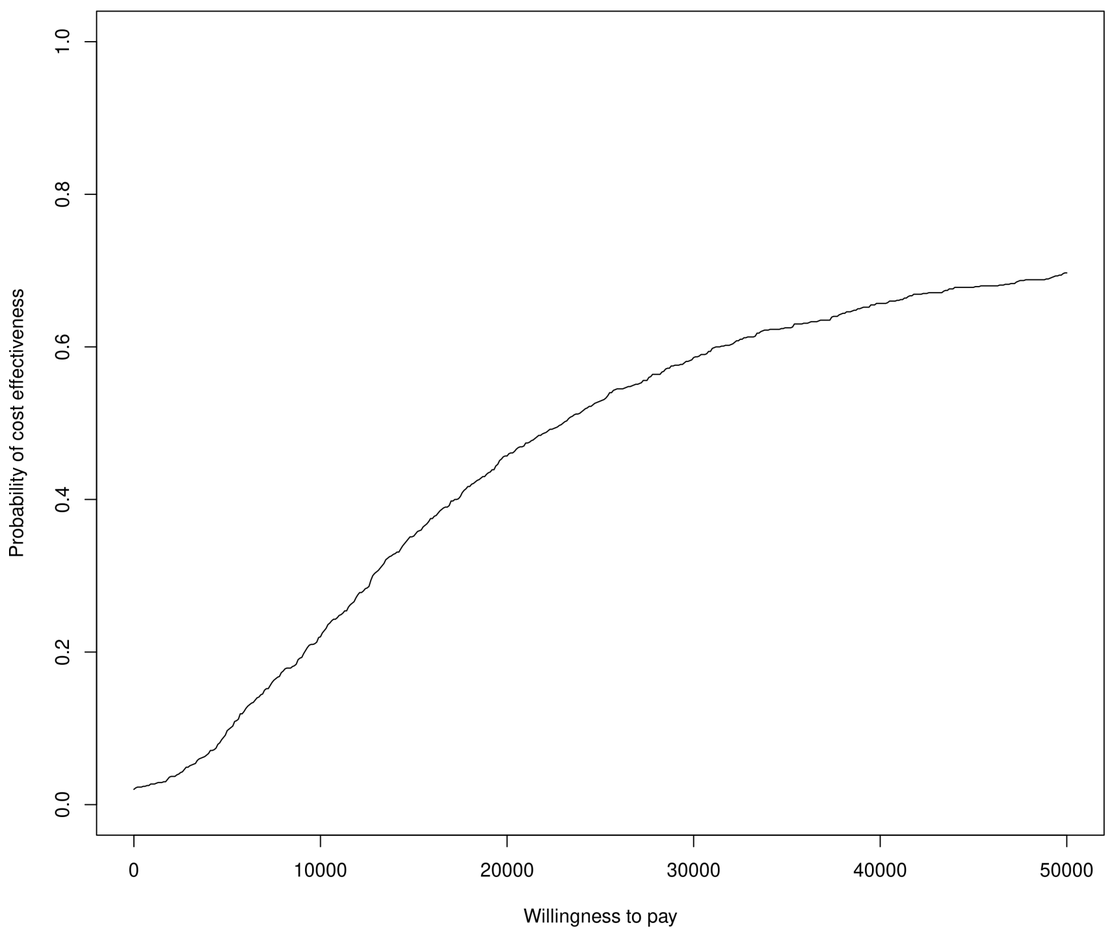
```
# Using 'ggplot', you can go crazy with customisation...
ceplane.plot(m,wtp=10000,graph="gg",point=list(color="blue",size=1.8),area=list(fill="springgreen3"))
```



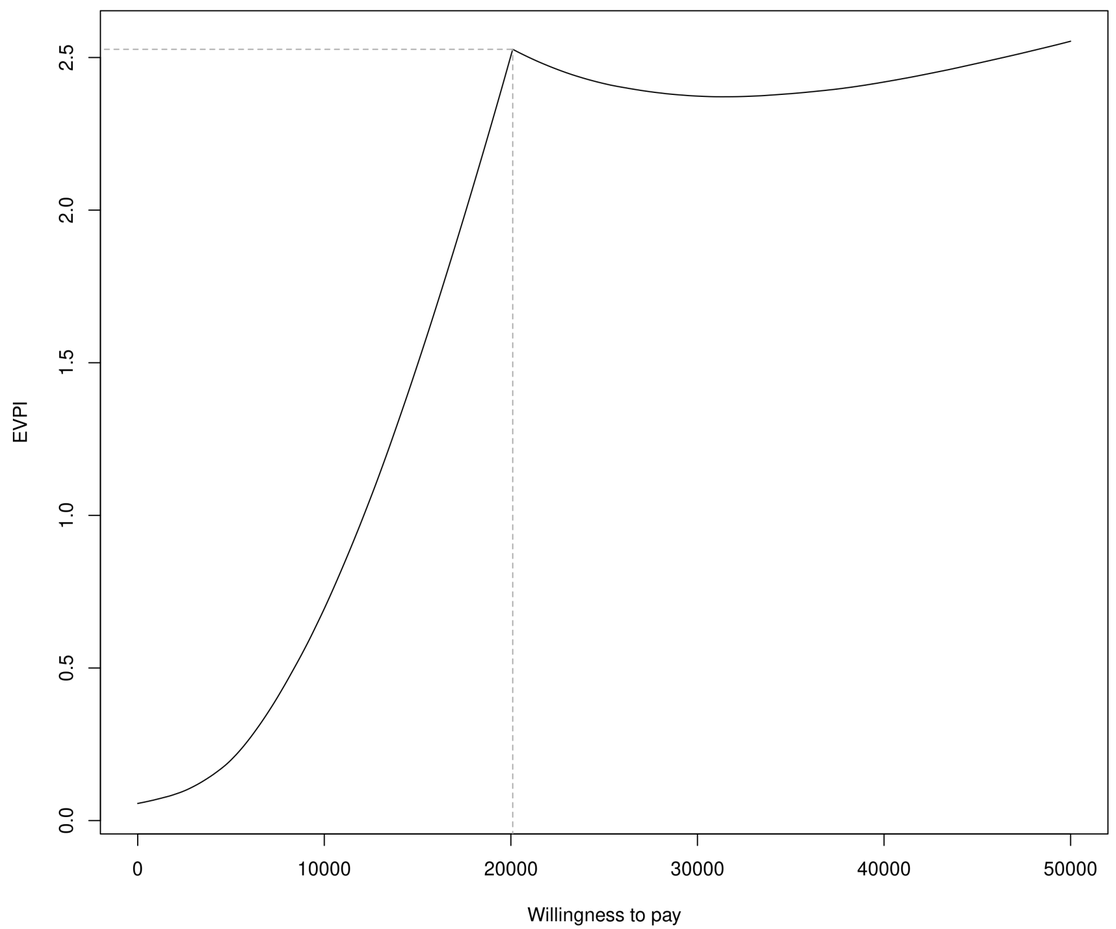
- https://ggplot2.tidyverse.org/
- https://n8thangreen.github.io/BCEA/

How does BCEA work?

```
# Plots the Cost-Effectiveness Acceptability Curve
ceac.plot(m)
```



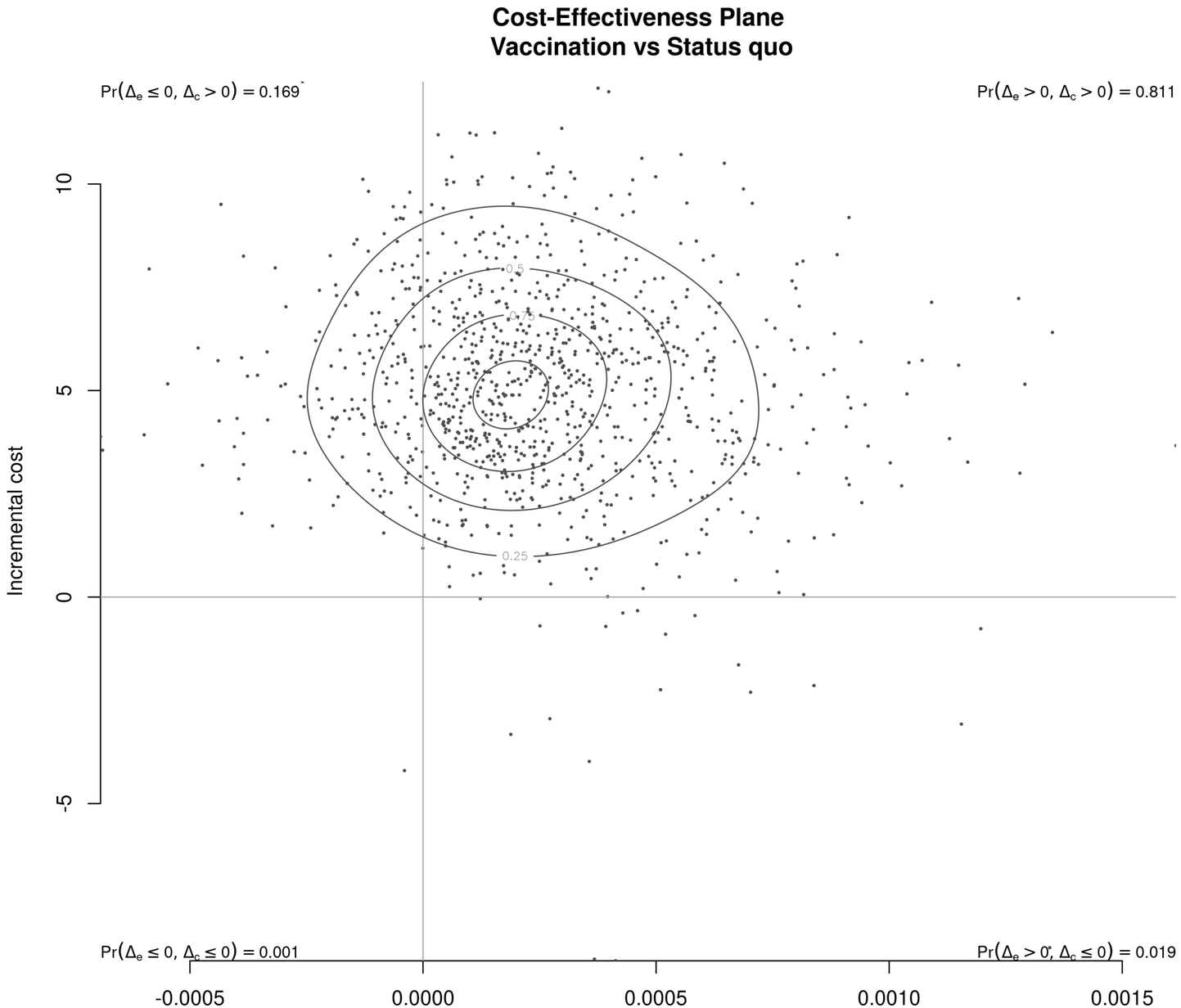

```
# Plots the Expected Value of Partial Information (EVPI)
evi.plot(m)
```



Specialised plots

- Can generate a *contour* plot of the cost-effectiveness plane and estimate the proportion of points in each quadrant

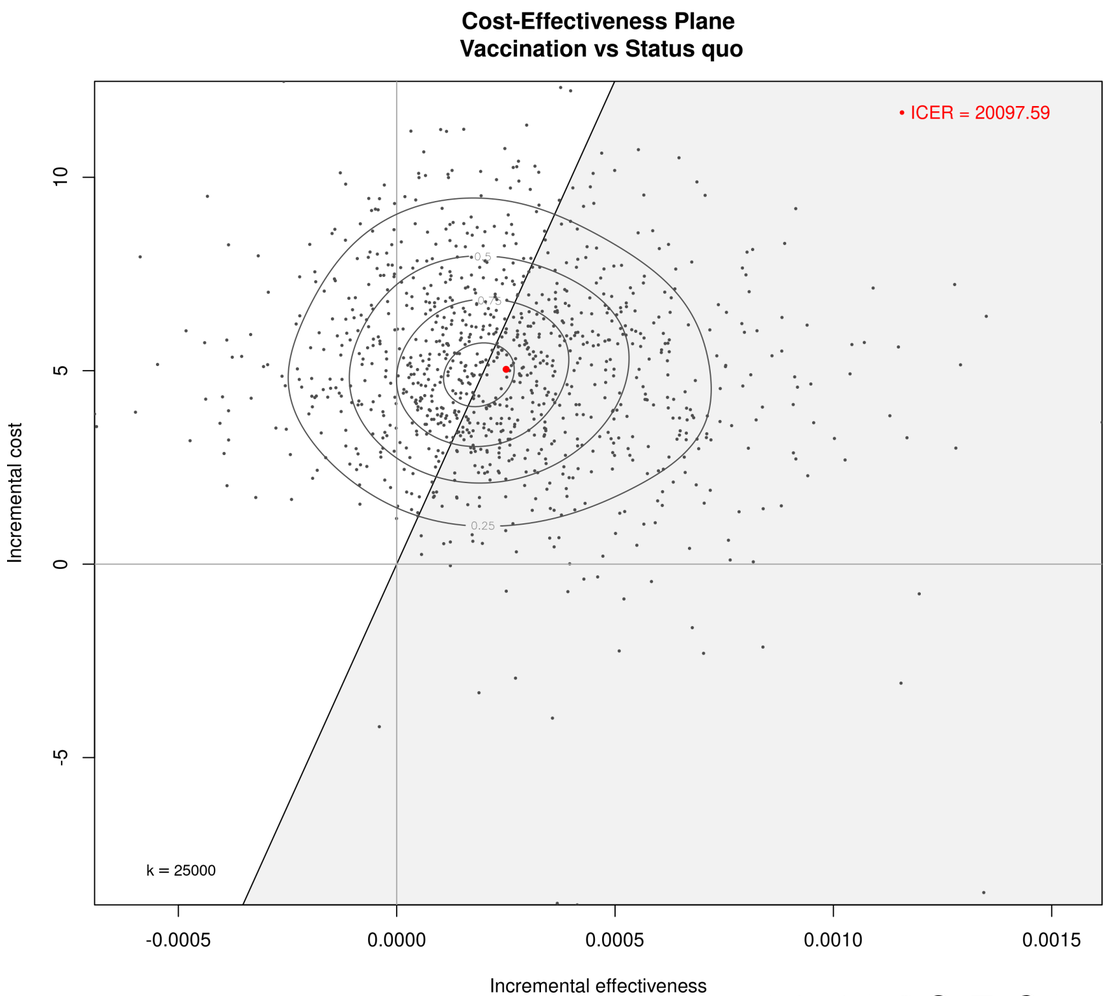
```
# "Basic" contourplot
contour(m)
```



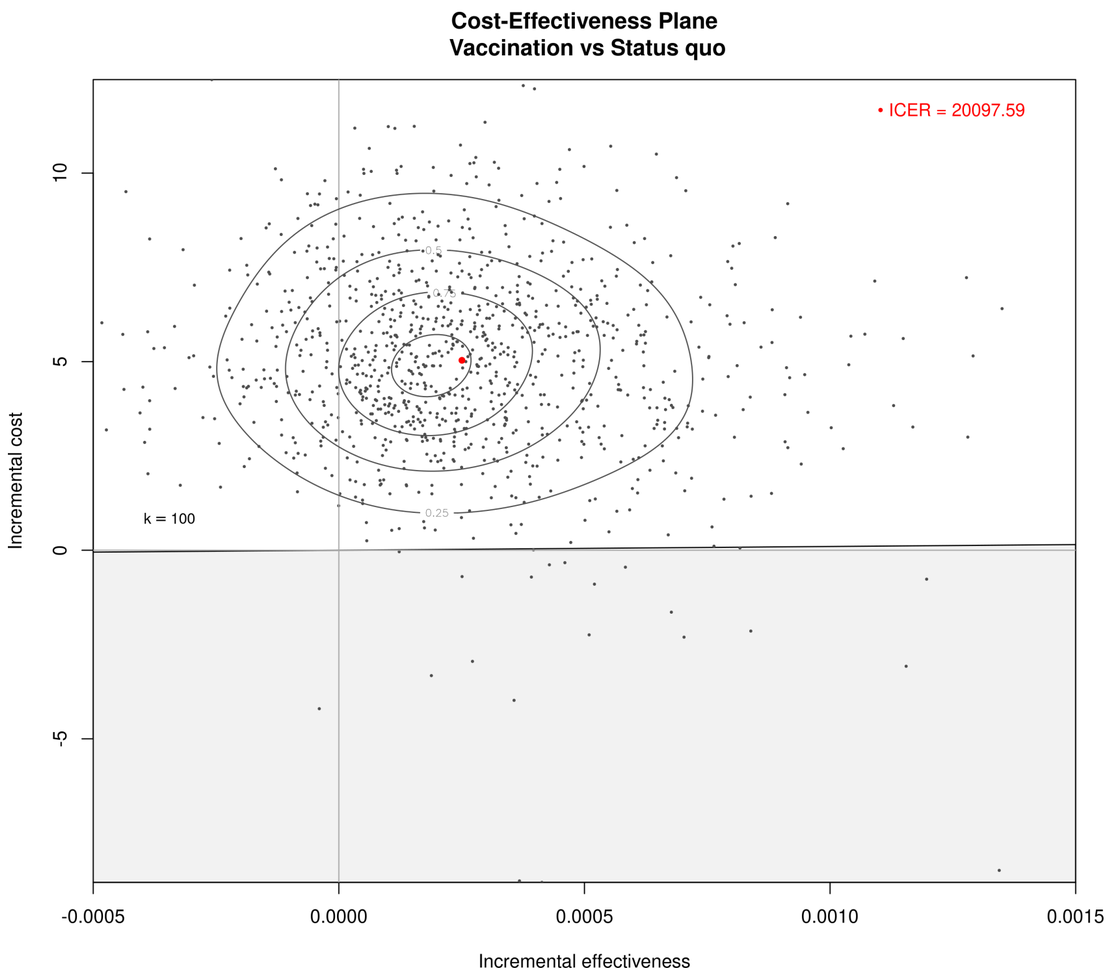
Specialised plots

The specialised function `contour2` also shows the **sustainability area**

```
contour2(m)
```




```
contour2(m,wtp=100,xlim=c(-.0005,0.0015))
```



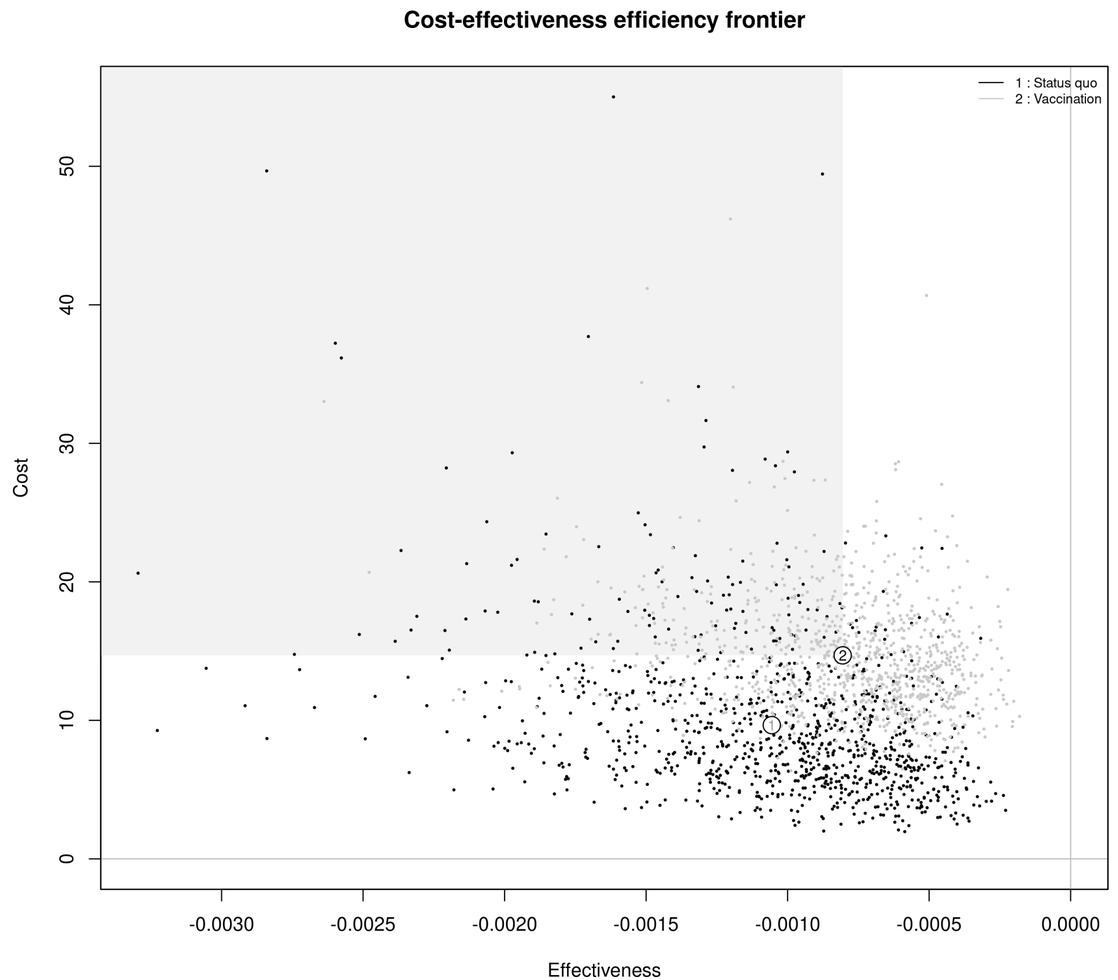
Specialised plots

Cost-effectiveness efficiency frontier

```
1 ceef.plot(m,print.plot=FALSE)
```

```
Cost-effectiveness efficiency frontier summary

Interventions on the efficiency frontier:
            Effectiveness  Costs Increase slope Increase angle
Vaccination   -0.00080537 14.691             NA             NA

Interventions not on the efficiency frontier:
           Effectiveness  Costs     Dominance type
Status quo    -0.0010559 9.6555 Extended dominance
```

```
1 ceef.plot(m,print.summary=FALSE)
```



Exporting graphical output

- R has excellent graphical facilities and the graphs produced by BCEA can be easily exported to many different formats

```
# "Opens" the graphical device
pdf("NAME_OF_THE_FILE",width=`8`,height=`8`)      `# for 'pdf', units are in inches`
# Makes the plot
ceplane.plot(`BCEA_OBJECT`)                      `# of course, specify whatever name you've chosen when creating the object...`
# "Closes" the graphical device
dev.off()


# "Open" the graphical device"
jpeg("NAME_OF_FILE.jpg",width=`480`,height=`480`)  `# for 'jpeg' units are in px`
# Makes the plot
ceplane.plot(BCEA_OBJECT)
# "Closes" the graphical device
dev.off()
```

NB: Rstudio and rmarkdown can do even more – that's for another time…

Advanced use of BCEA

Multiple treatment comparisons

Probabilistic “depression model”

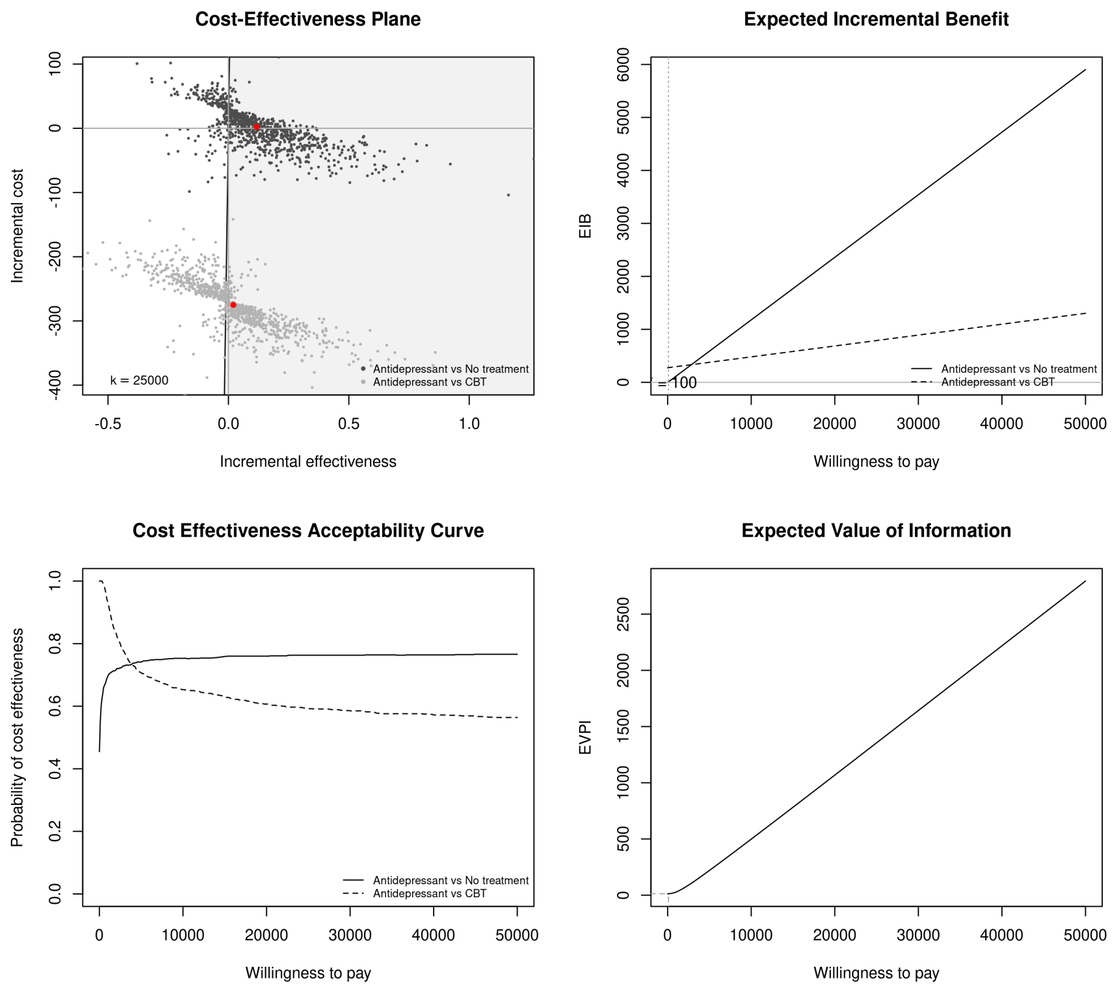
- Fictional model comparing antidepressants to cognitive behaviour therapy (CBT) and no treatment in people with depression
- Statistical modelling based on evidence synthesis
  - Benefits: based on QALYs
  - Costs: associated with treatments and various resources use
- Economic modelling: two matrices with relevant population summaries
  - `effects`
  - `costs`

- NB: The details of the actual modelling are *not* important for the purposes of demonstrating the example…

Multiple treatment comparisons

Probabilistic “depression model”

```
# Intervention labels
t.names<-c("No treatment","CBT","Antidepressant")

# "Standard" analysis: pairwise comparisons
depression.bcea = bcea(effects,costs,
                       interventions=t.names,ref=3)
# the third intervention is the reference

# Plots the results
plot(depression.bcea)
```

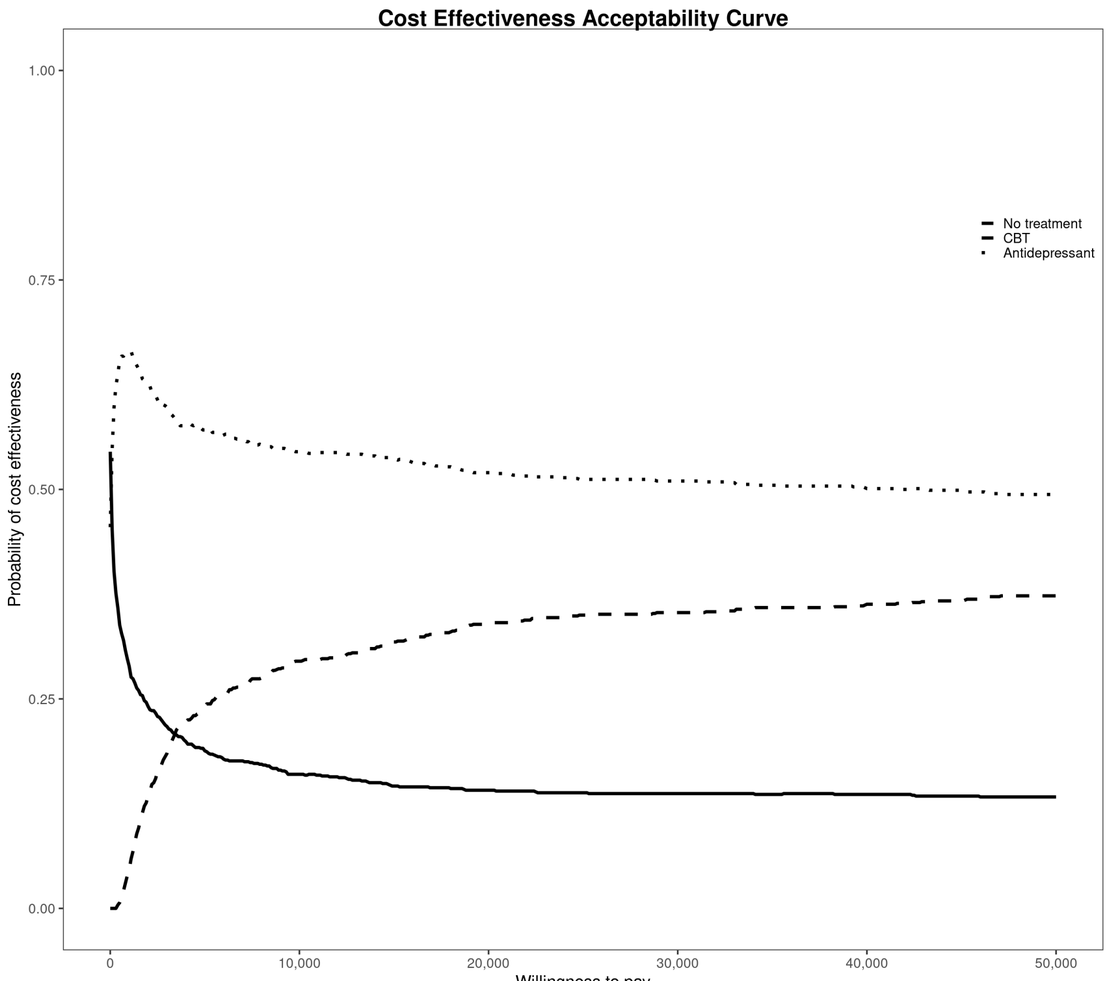


Multiple treatment comparisons

Probabilistic “depression model”

```
# For `multiple treatment comparison`
depression.multi.ce = multi.ce(depression.bcea)

# Specialised plot method
ceac.plot(depression.multi.ce,pos=c(1,0.8),graph="ggplot2")
```

NB: In older releases of BCEA, this graph was done using the **deprecated** function `mce.plot`

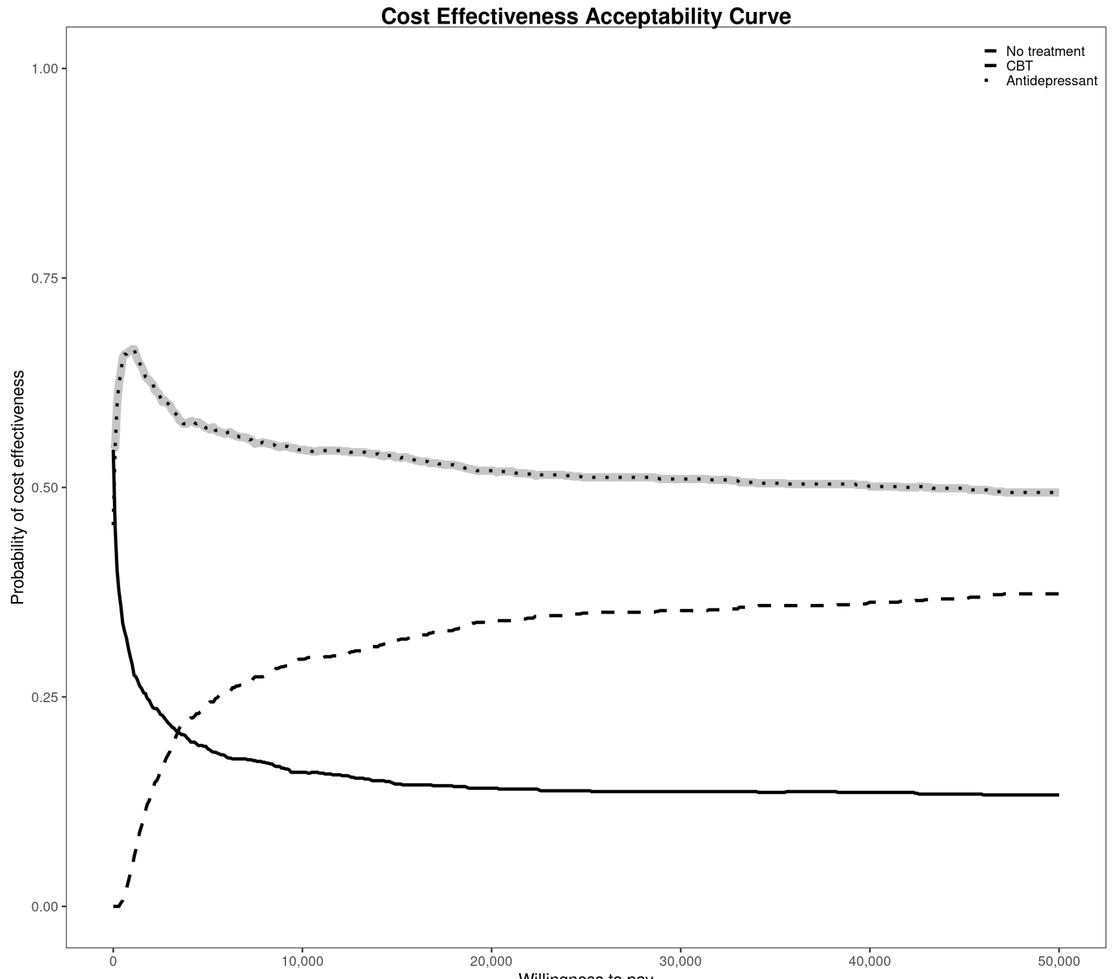


Multiple treatment comparisons

Probabilistic “depression model”

- Can use `ggplot` to customise the graph

```
ceac.plot(depression.multi.ce,pos=c(1,1),graph="ggplot2") +
  ggplot2::stat_summary(fun=max, geom="line",
                        colour="grey25", alpha=.3, lwd=2.5)
```

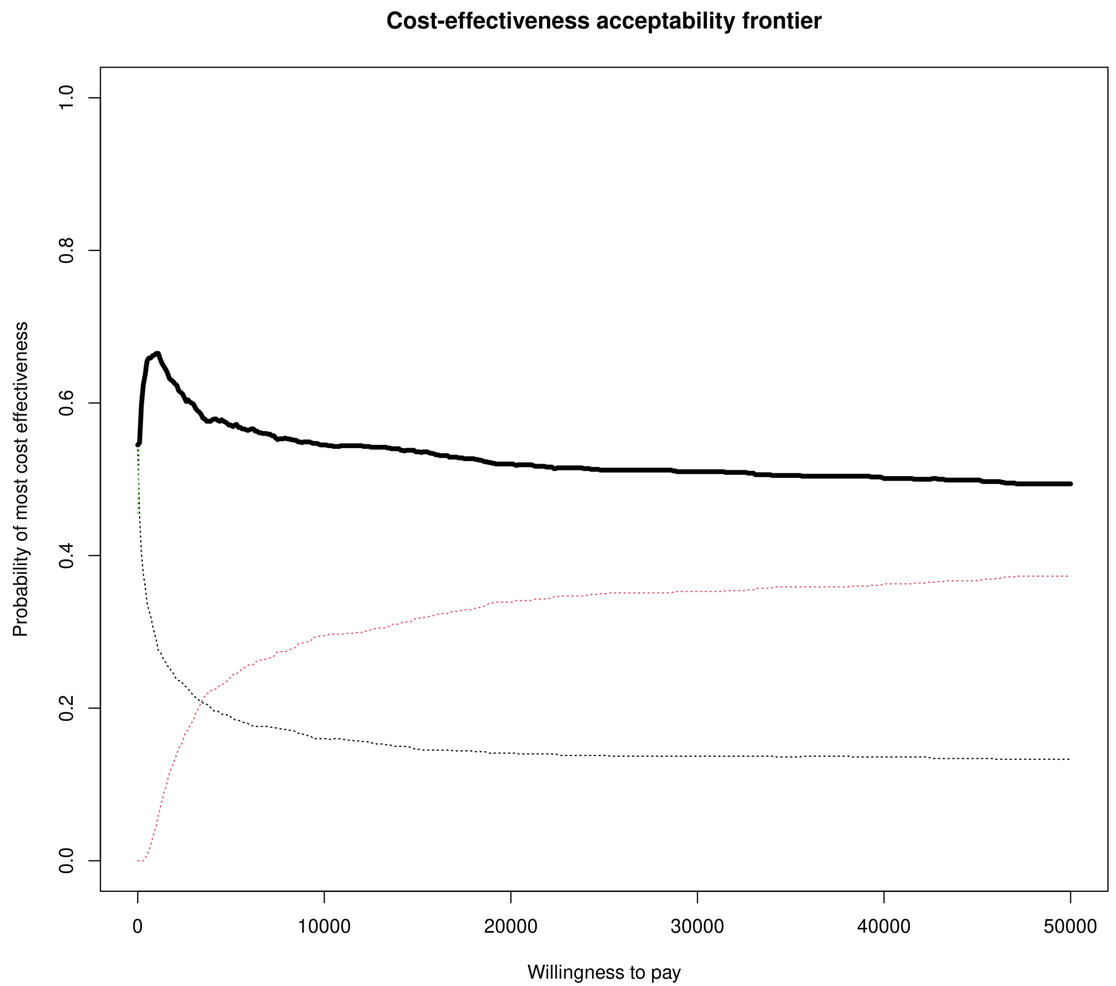


Multiple treatment comparisons

Probabilistic “depression model”

- Can also use the specialised function `ceaf.plot`

```
1  # Specialised plot
2  ceaf.plot(depression.multi.ce)
```



BCEAweb

- Inspired by similar projects – eg SAVI
- Create a web interface to use BCEA without even opening R (or even having it installed on your computer!)

- Typical work flow
  1. Design the economic model (eg Markov model, decision tree, …)
  2. Run the statistical analysis to estimate the quantities of interest (eg survival analysis, evidence synthesis, …)
  3. Run the economic model and obtain "PSA samples"
  4. Upload "PSA samples", including values for (e, c) to `BCEAweb`
  5. Use `BCEA` in the background to do **all** the economic analysis
  6. Create reports that can be used as the basis for papers, reimbursement files, …

BCEAweb

```r
# Creates a matrix with the underlying model simulations
inp = createInputs(vaccine_mat, print_is_linear_comb=FALSE)

# Runs BCEAweb
BCEAweb(e=e,                  # matrix of simulations for the effectiveness
        c=c,                  # matrix of simulations for the costs
        parameters=inp$mat    # matrix of simulations for all the model parameters
       )
```

- BCEAweb exists as a standalone webapp
  - Can access it here
- **Or**, you can launch your own "local" version from the BCEA package (as in the code above)!
  - This will launch a web page from which you can manipulate your output (Live Demo)


THANK YOU SO MUCH.

// Alebaio's script to place logo on all slides but the first page