

# Introduction to R for Health Economics using BCEA

## Gianluca Baio

Department of Statistical Science | University College London

Follow our  
departmental  
social media  
accounts   

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

- ✉ [g.baio@ucl.ac.uk](mailto: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](https://twitter.com/gianlubaio)

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

6 November 2022

# Disclaimer...



Manuela Joore  
@ManuelaJoore · [Follow](#) 

Best opening sentence [#ISPOREurope](#) from Gianluca Baio: “statisticians should rule the world and Bayesian statisticians should rule all statisticians”



Gianluca Baio @gianlubaio  
Ready for our session on open source models & methods!

4:52 PM · Nov 4, 2019 

 16  Reply  Copy link

[Read 2 replies](#)

...Just so you know what you're about to get into... 😊

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

## Statistical model

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

# Health technology assessment (HTA)

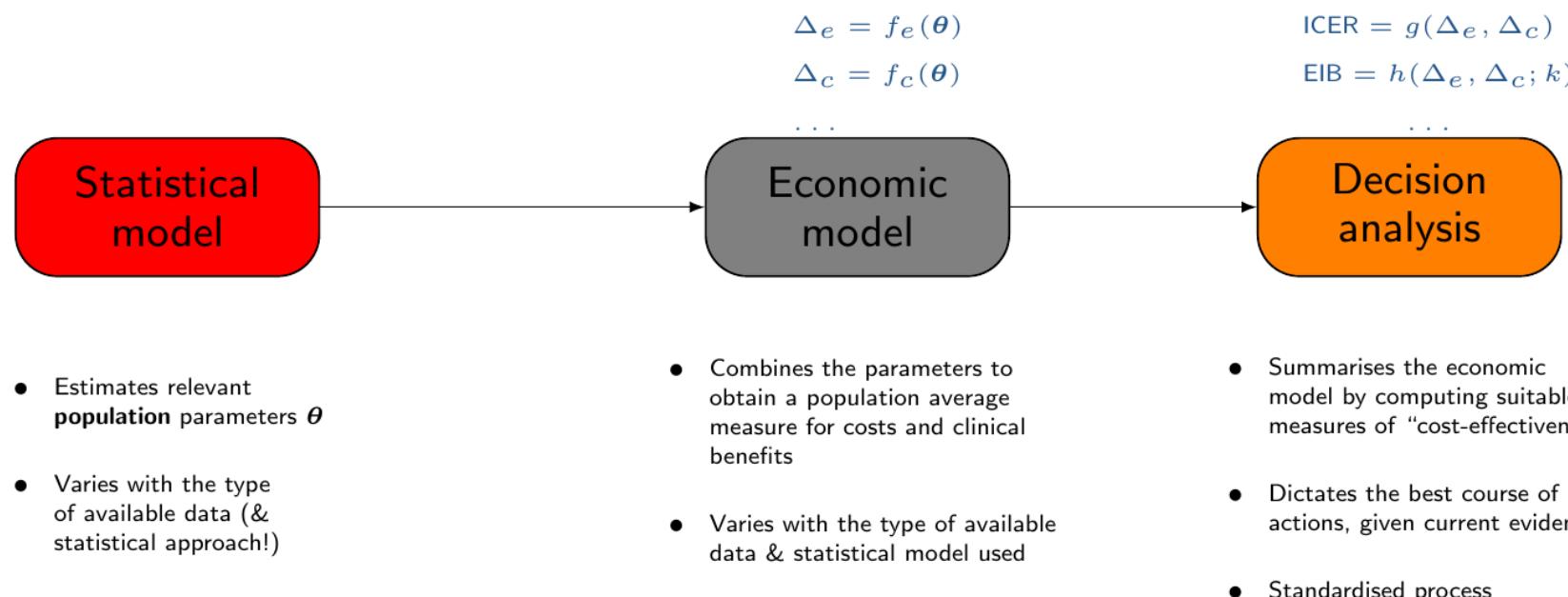
**Objective:** Combine **costs** and **benefits** of a given intervention into a rational scheme for allocating resources



- Estimates relevant **population** parameters  $\theta$
- 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

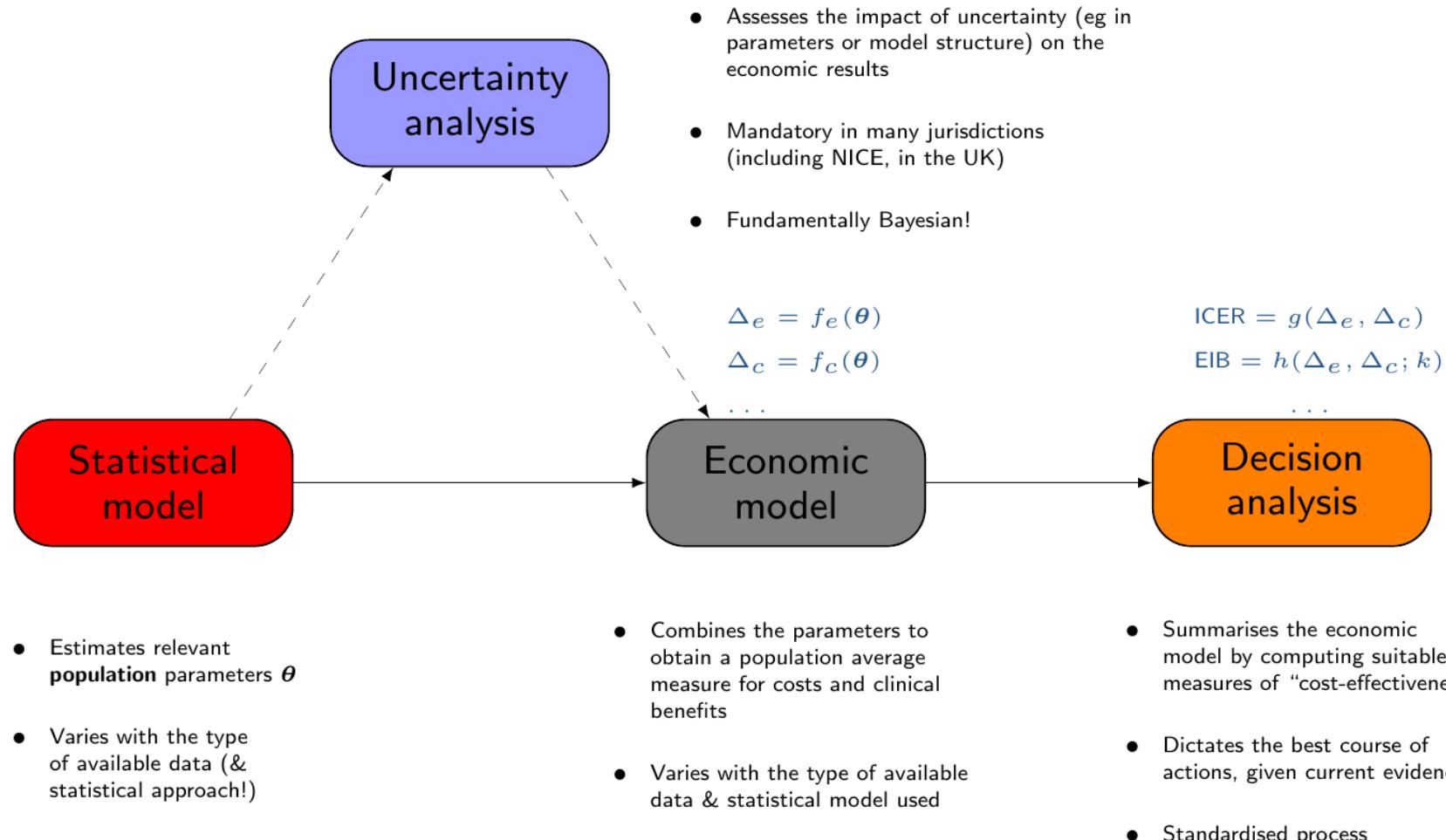
# Health technology assessment (HTA)

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



# Health technology assessment (HTA)

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



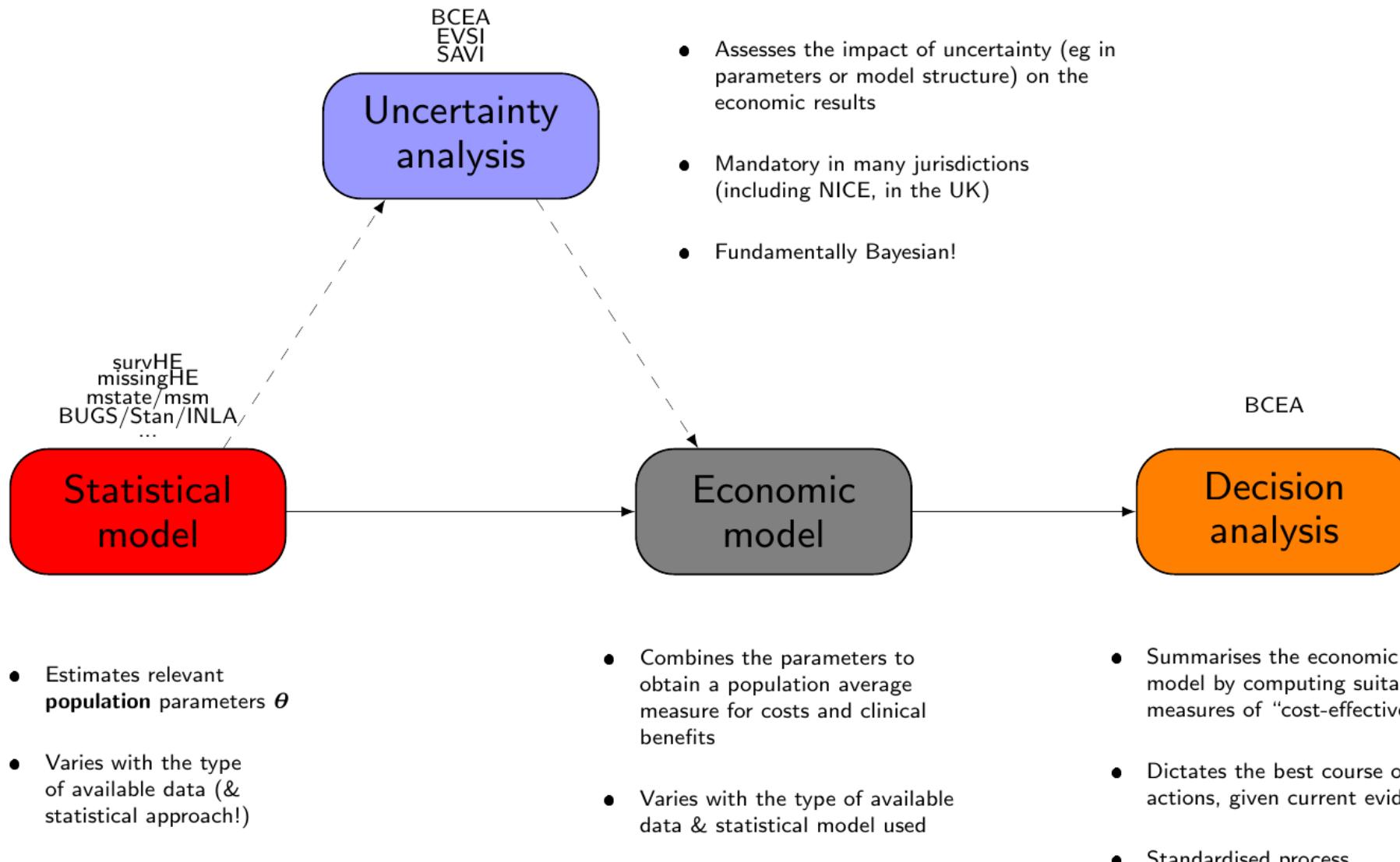
# Health technology assessment (HTA)



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

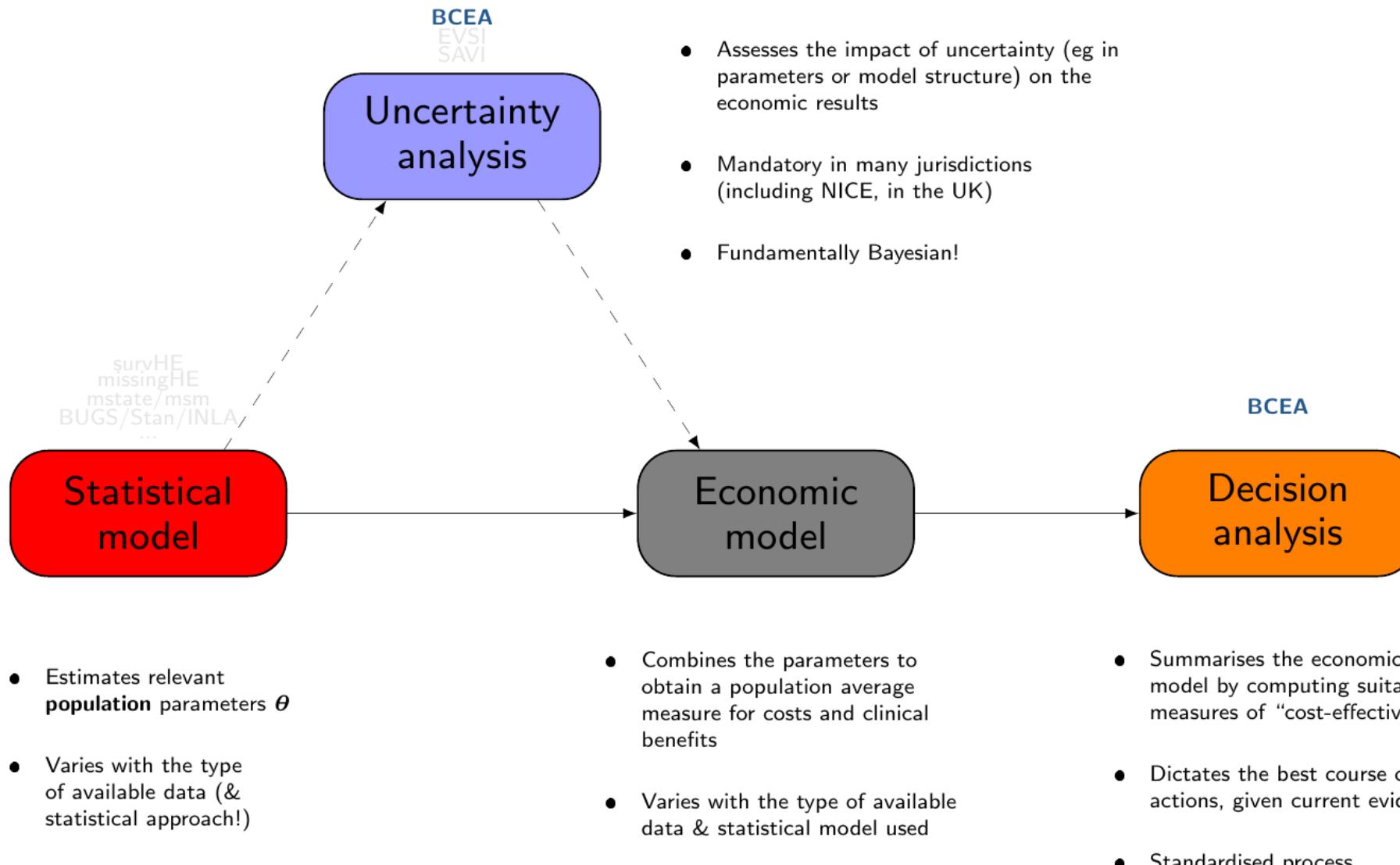
# Health technology assessment (HTA)

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



# 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**

## 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...

## Section added to the site with Tutorials!



We have added a new section to the website, with a view of collating tutorials on tools for economic evaluation based on R. The first one is an online tutorial on creating a Shiny interface for Markov models by [Robert Smith](#) and [Paul Schneider](#) from University of Sheffield. Shiny allows you to create user friendly and web-accessible graphical user interfaces for your R models. More details [here](#)"



Tweets from

*"Transparency is in the eye of the beholder"*

(Andy Briggs at the **R-HTA workshop** – October 2020)



*"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

## I 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)

# 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  $k$ ),...

# 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

# 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  $k$ ),...

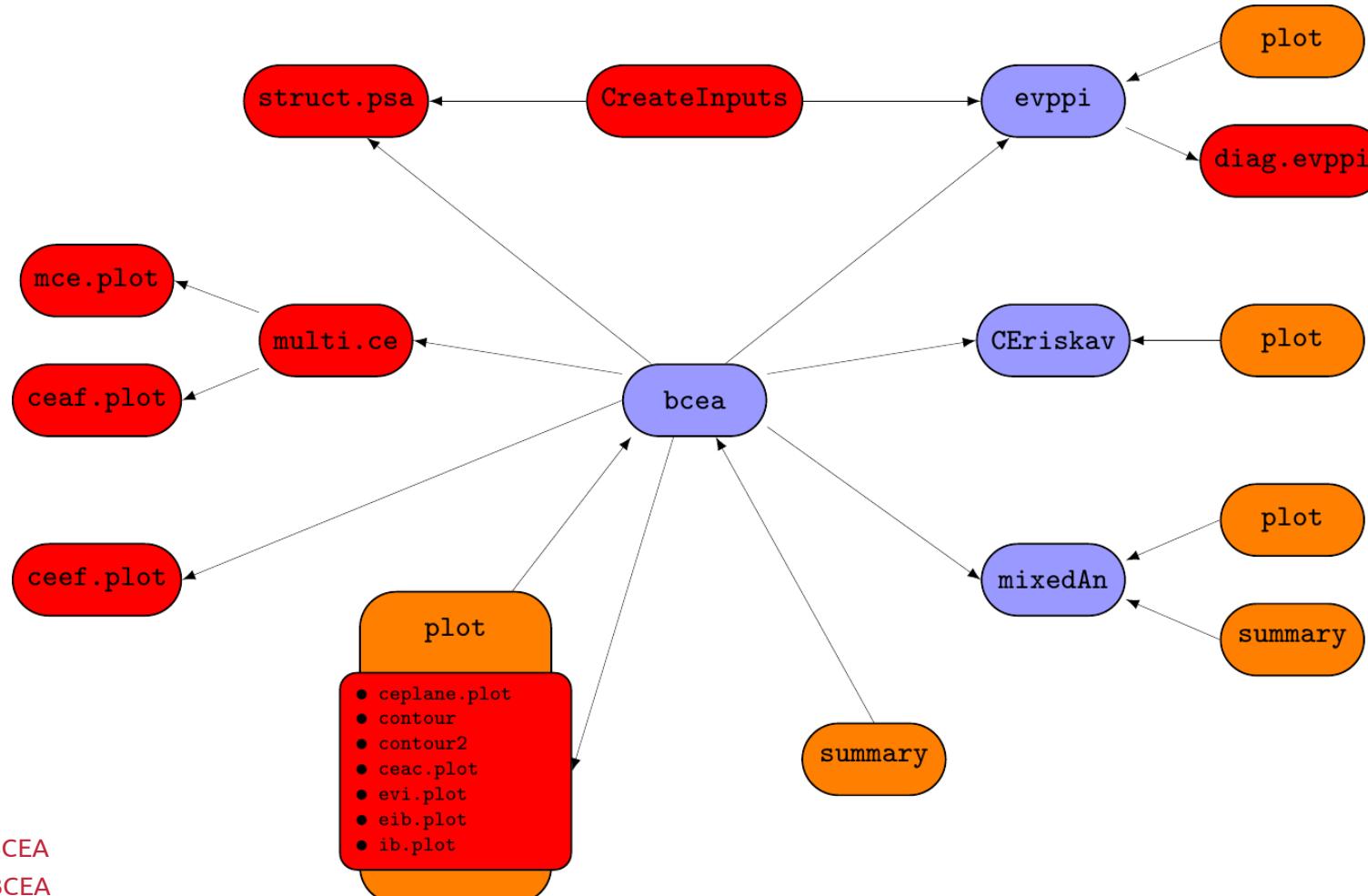
## 3 Perform uncertainty analysis

- 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)

# A R package for (Bayesian) cost-effectiveness analysis



# Using BCEA to summarise outputs of an economic model

# How does BCEA work?

Installation

Using BCEA

Show. Me. The. Data!

...

Economic model

Cost & effects

- 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")  
>  
> # Or the development version from the GitHub repository  
> remotes::install_github("giabaio/BCEA")           # stable version (2.4-2)  
>  
> remotes::install_github("giabaio/BCEA", ref="dev") # development version (2.4-2)
```

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?

[Installation](#)[Using BCEA](#)[Show. Me. The. Data!](#)[...](#)[Economic model](#)[Cost & effects](#)

```
> library(dplyr) # (Not necessary - helpful for data manipulation!)
>
> library(BCEA) # Then loads the package (so you can access its functions)
> data(Vaccine) # Loads an example dataset
```

- The "Vaccine" example is a fictional cost-effectiveness model for an influenza vaccine, based on evidence synthesis (and a real case)
- 2 treatment options ("Standard of care" vs "Vaccination") and overall 63 parameters
- Discussed in details in  [Baio et al, 2017](#) and  [Baio and Dawid, 2011](#)
- In this case, PSA simulations obtained from a full Bayesian model, but could be done in a spreadsheet and imported into R

# How does BCEA work?

[Installation](#)[Using BCEA](#)[Show. Me. The. Data!](#)[...](#)[Economic model](#)[Cost & effects](#)

```
> # The object 'Vaccine' contains a matrix 'vaccine_mat', with all the simulated values for the many model  
> # parameters BCEA can create a matrix with the underlying model simulations starting from various formats  
> # (BUGS/R/Excel) and can get rid of "redundant" columns (those that are linear combination of each other...)  
> inp = createInputs(vaccine_mat, print_is_linear_comb = FALSE)  
>  
> # Visualise the output: "piping" ('%>%') + nicer visualisation  
> inp$mat %>% as_tibble()
```

**NB:** There are various R packages/tools to work with tables

- **kable** and **kableExtra**

# How does BCEA work?

Installation	Using BCEA	Show. Me. The. Data!	...	Economic model	Cost & effects										
# A tibble: 1,000 × 56															
1	1466	1	0	0	1664	958	230	0	1	0	5992	3401	876	691	405
2	5329	1	1	0	1414	748	276	0	0	1	7471	4024	1536	570	308
3	5203	1	1	0	809	489	80	0	0	0	6718	4300	788	332	214
4	2351	2	0	0	1761	1157	261	1	0	0	4837	3269	702	739	479
5	8303	1	2	0	2472	964	432	1	1	0	4749	1894	846	1049	425
6	3607	1	1	0	2224	1342	260	1	0	0	4938	2976	596	915	560
7	6304	4	1	1	3478	1107	591	2	1	0	11080	3547	2045	1505	482
8	4337	1	1	1	1483	799	189	0	0	0	3867	2164	525	622	333
9	5482	0	0	0	1587	798	279	0	0	0	5163	2532	910	675	332
10	3125	2	2	0	2578	1681	243	0	0	0	7265	4766	700	1063	711
# ... with 990 more rows, 38 more variables: Pneumonia.2.2. <dbl>, Trt.1.1.1. <dbl>, Trt.2.1.1. <dbl>, Trt.1.2.1. <dbl>, Trt.2.2.1. <dbl>, Trt.1.2.2. <dbl>, Trt.2.2.2. <dbl>, beta.1. <dbl>, beta.2. <dbl>, beta.3. <dbl>, beta.4. <dbl>, beta.5. <dbl>, beta.6. <dbl>, delta <dbl>, eta <dbl>, gamma.1. <dbl>, gamma.2. <dbl>, lambda <dbl>, n.1.2. <dbl>, n.2.2. <dbl>, phi <dbl>, pi.1.2. <dbl>, pi.2. <dbl>, pi.3. <dbl>, pi.4. <dbl>, pi.5. <dbl>, pi.6. <dbl>, pi.7. <dbl>, pi.8. <dbl>, q.1. <dbl>, q.4. <dbl>, q.5. <dbl>, q.7. <dbl>, rho.2. <dbl>, xi <dbl>, and abbreviated variable names `¹Adverse.events, `²Death.1.1., `³Death.2.1., `⁴Death.2.2., `⁵H															
# `⁶Hospital.2.1., `⁷Hospital.2.2., `⁸Infected.1.1., `⁹Infected.2.1., `*Infected.2.2., `*Mild.Compl.1.1., `*Mild.Compl.2.1., `*Mild.Compl.3.1., `*Pneumonia.1.1., `*Pneumonia.2.1.															

# How does BCEA work?

[Installation](#)[Using BCEA](#)[Show. Me. The. Data!](#)[...](#)[Economic model](#)[Cost & effects](#)

```
> # Defines the number of simulations considered
> n.sims=inp$mat %>% nrow()
> # applies the function 'nrow' (number of rows) to the object 'inp$mat'
> # NB: Since R 4.1.0, can also use 'native' pipe ('|>')
> # (probably a bit quicker, but in most cases, may be immaterial...)
>
> # Aggregates the model inputs to compute (e,c)
> QALYs.inf = QALYs.pne <- QALYs.hosp <- QALYs.adv <- QALYs.death <- matrix(0,n.sims,2)
> for (t in 1:2) {
+   QALYs.inf[,t] = ((Infected[,t,1] + Infected[,t,2])*omega[,1]/365)/N
+   QALYs.pne[,t] = ((Pneumonia[,t,1] + Pneumonia[,t,2])*omega[,4]/365)/N
+   QALYs.hosp[,t] = ((Hospital[,t,1] + Hospital[,t,2])*omega[,5]/365)/N
+   QALYs.death[,t] = ((Death[,t,1] + Death[,t,2])*omega[,6])/N
+ }
> QALYs.adv[,2] = (Adverse.events*omega[,7]/365)/N
> e = -(QALYs.inf + QALYs.pne + QALYs.adv + QALYs.hosp + QALYs.death) + ...
```

- NB: The data stored in the Vaccine object (built-in in BCEA) already contains the objects e, c that can be used to run the decision analysis...
- So, this step is actually not needed (but documented in  Baio et al, 2017)

# How does BCEA work?

[Installation](#)[Using BCEA](#)[Show. Me. The. Data!](#)

...

[Economic model](#)[Cost & effects](#)

```
> cbind(eff,cost) %>% as_tibble(.name_repair="universal")      # ensures that the columns are named
```

```
# A tibble: 1,000 × 4
  Status.Quo...1 Vaccination...2 Status.Quo...3 Vaccination...4
    <dbl>          <dbl>          <dbl>          <dbl>
1 -0.00105       -0.000899      10.4          16.3 
2 -0.000884      -0.000732      5.83         9.37  
3 -0.000890      -0.000698      5.78        15.9  
4 -0.00164       -0.00114       12.2         18.7  
5 -0.00135       -0.000957      9.79        16.5  
6 -0.00143       -0.000936      6.56        9.69  
7 -0.000960      -0.00105       8.45        11.3  
8 -0.00181       -0.00139       6.76        9.99  
9 -0.000842      -0.000556      3.60        10.1  
10 -0.00168      -0.00105       4.09        11.0 
# ... with 990 more rows
```

- These calculations *can* be done also in a spreadsheet (nothing more than algebra, **once you have the simulations**)

# 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 `e` or `c`)
  - `interventions`: a vector of labels for the interventions being compared
  - `Kmax`: the maximum value of  $k$ , the parameter of willingness to pay

# 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 `e` or `c`)
  - `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"
[10] "ib"              "eib"             "kstar"          "best"           "U"              "vi"            "Ustar"         "ol"
[19] "ref"             "comp"            "step"           "interventions" "e"              "c"
```

# How does BCEA work?

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

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

Cost-effectiveness analysis summary

Reference intervention: Vaccination  
Comparator intervention: Status quo

Optimal decision: choose Status quo for  $k < 20100$  and Vaccination for  $k \geq 20100$

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	20098

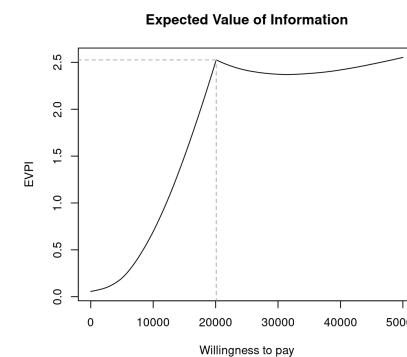
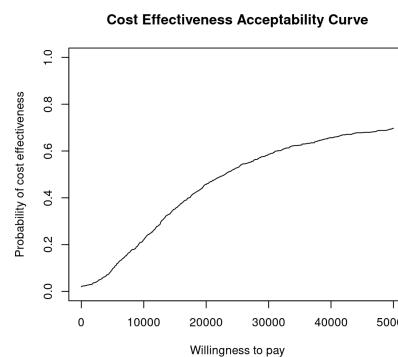
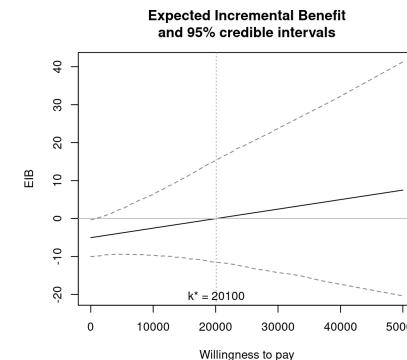
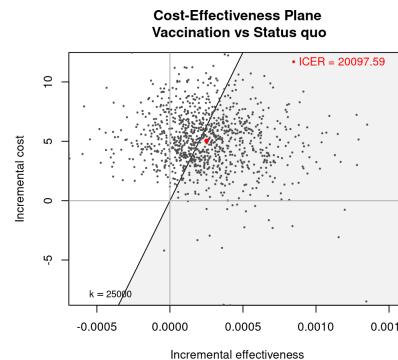
Optimal intervention (max expected net benefit) for  $k = 25000$ : Vaccination

EVPI 2.4145

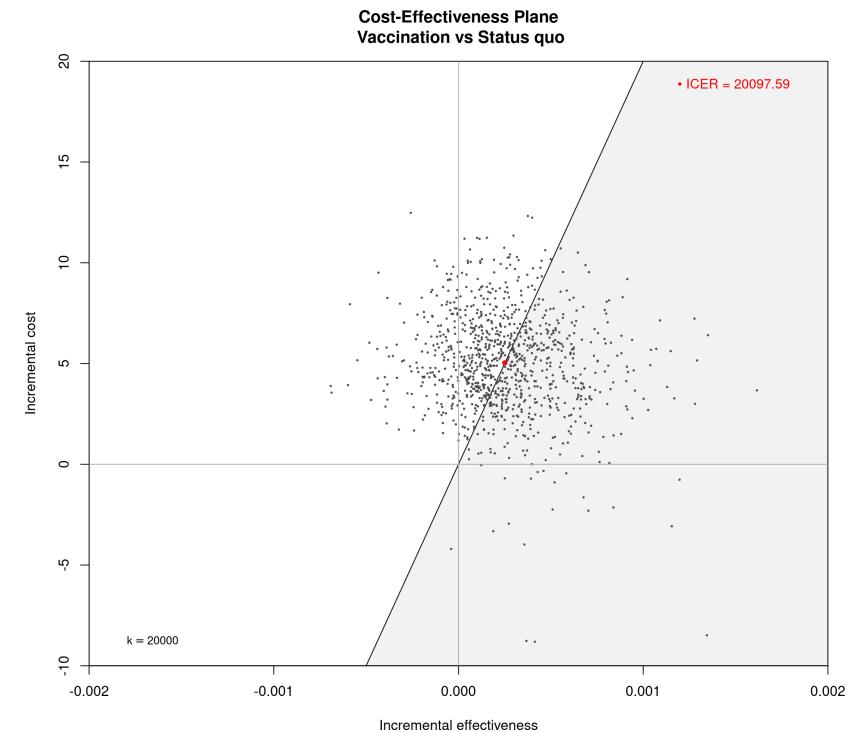
# How does BCEA work?

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

```
> # The 'plot' "method" produces a *specific*
> # version of graphical output
> 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/>



# BCEA

Perform Bayesian Cost-Effectiveness Analysis in R.

🚀 **Version 2.4.1 out now!** [Check out the release notes here.](#)



## Links

[View on CRAN](#)

[Browse source code](#)

[Report a bug](#)

## License

[Full license](#)

[GPL-3](#)

## Community

[Contributing\\_guide](#)

## Citation

[Citing\\_BCEA](#)

## Developers

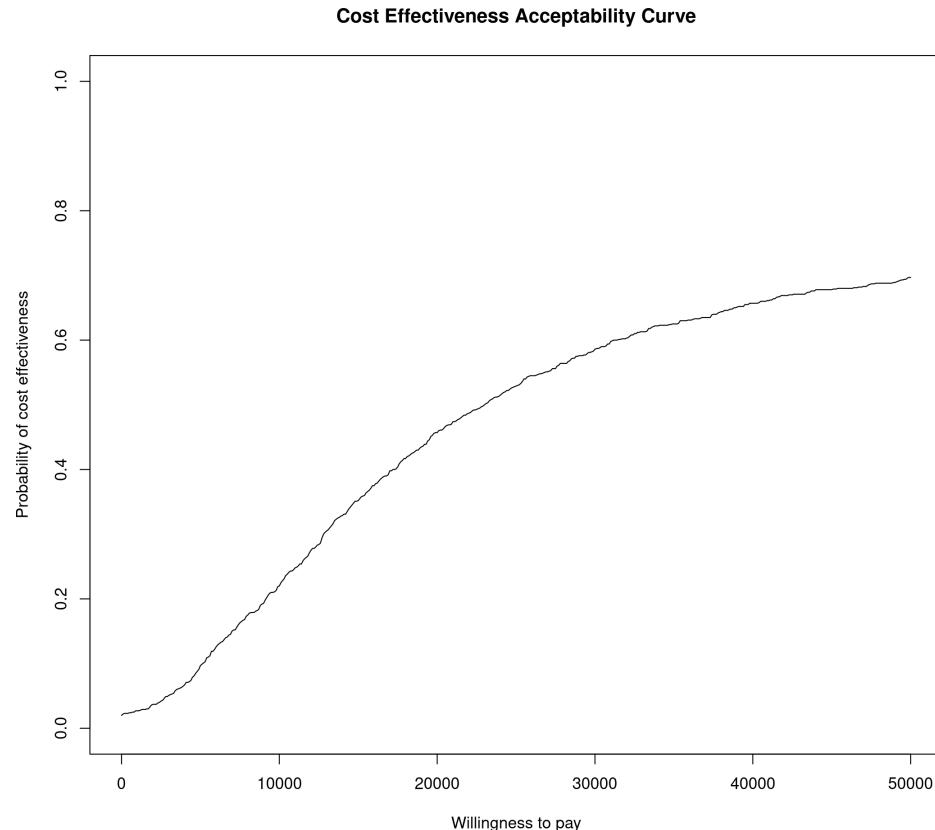
## Contents

- [Overview](#)
- [Features](#)
- [Installation](#)
- [Articles](#)
- [Further details](#)

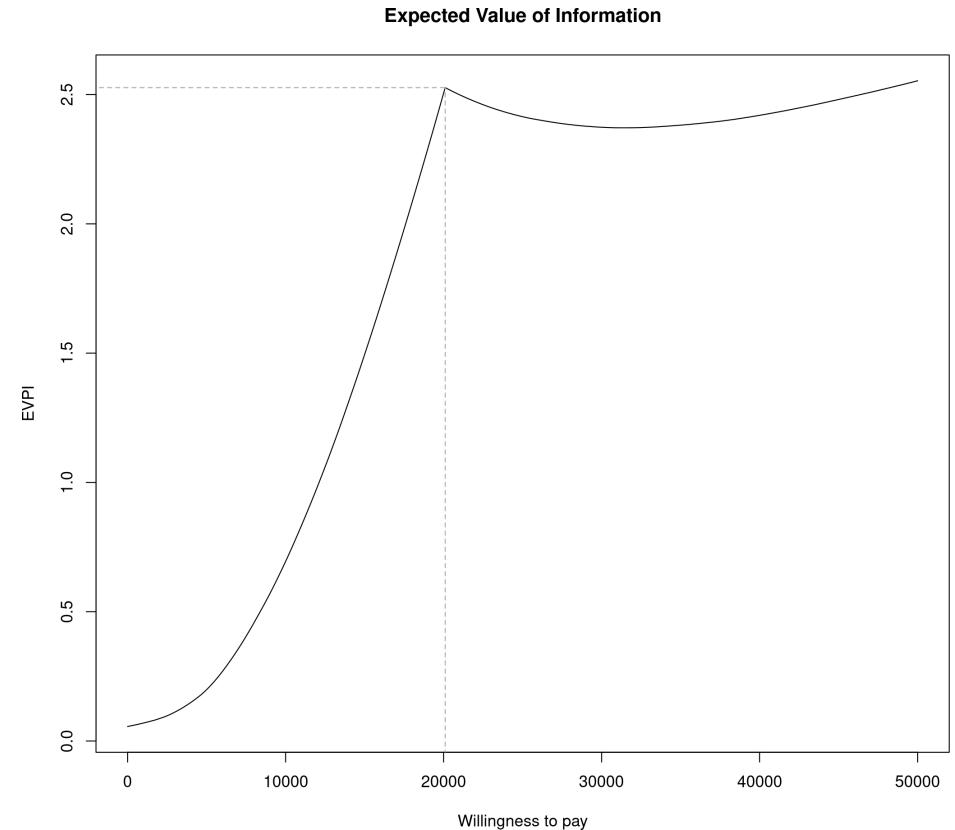
## Overview

# How does BCEA work?

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



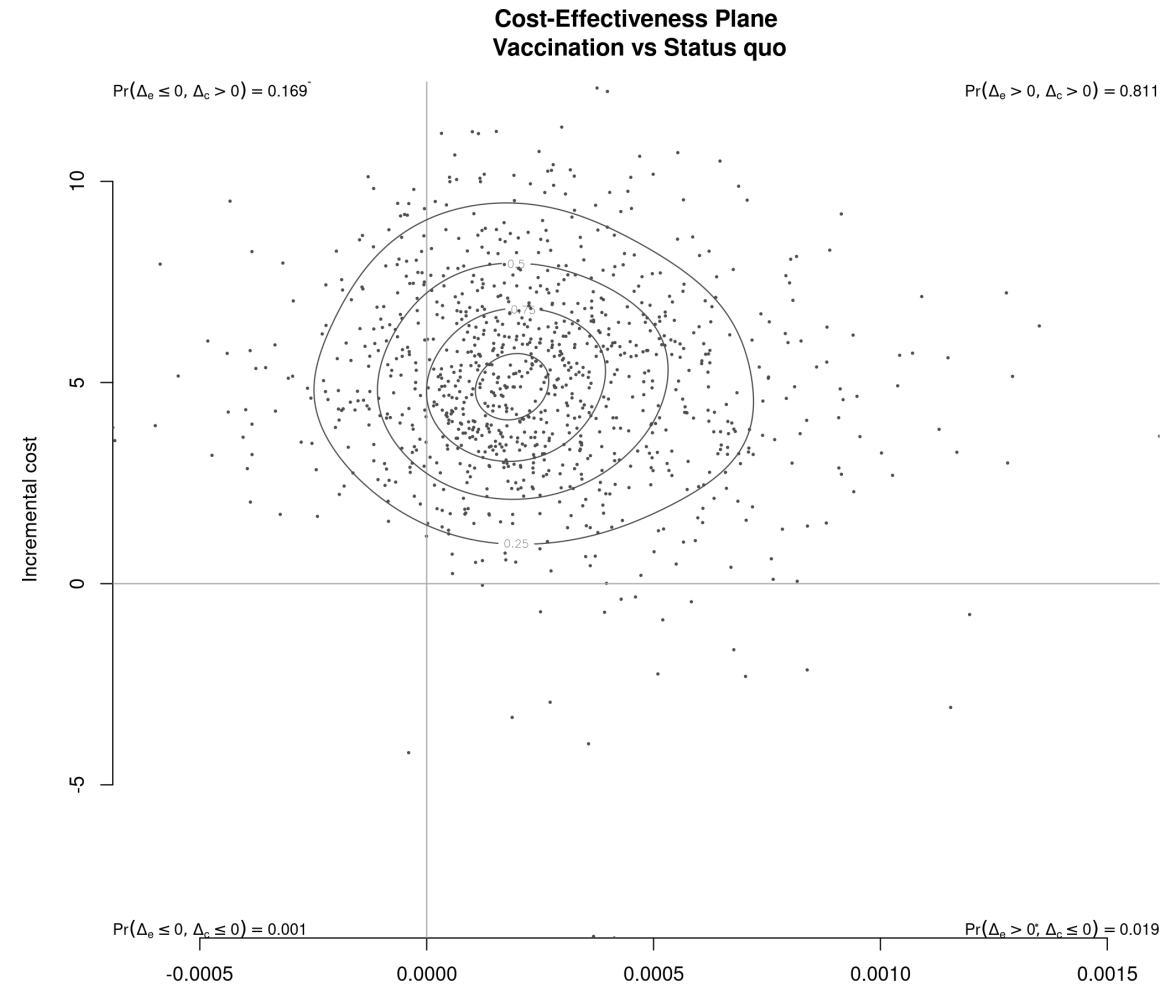
```
> # Plots the Expected Value of Partial Information (EVPI)  
> evi.plot(m)
```



# Specialised plots

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

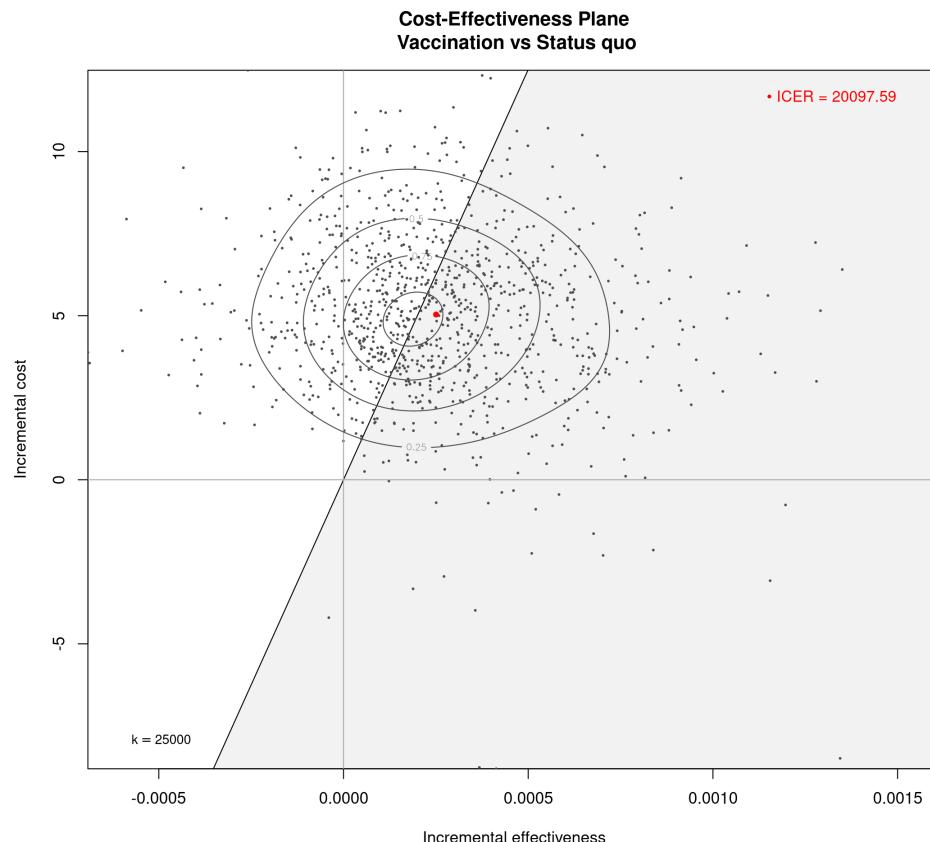
```
> # "Basic" contourplot  
> contour(m)
```



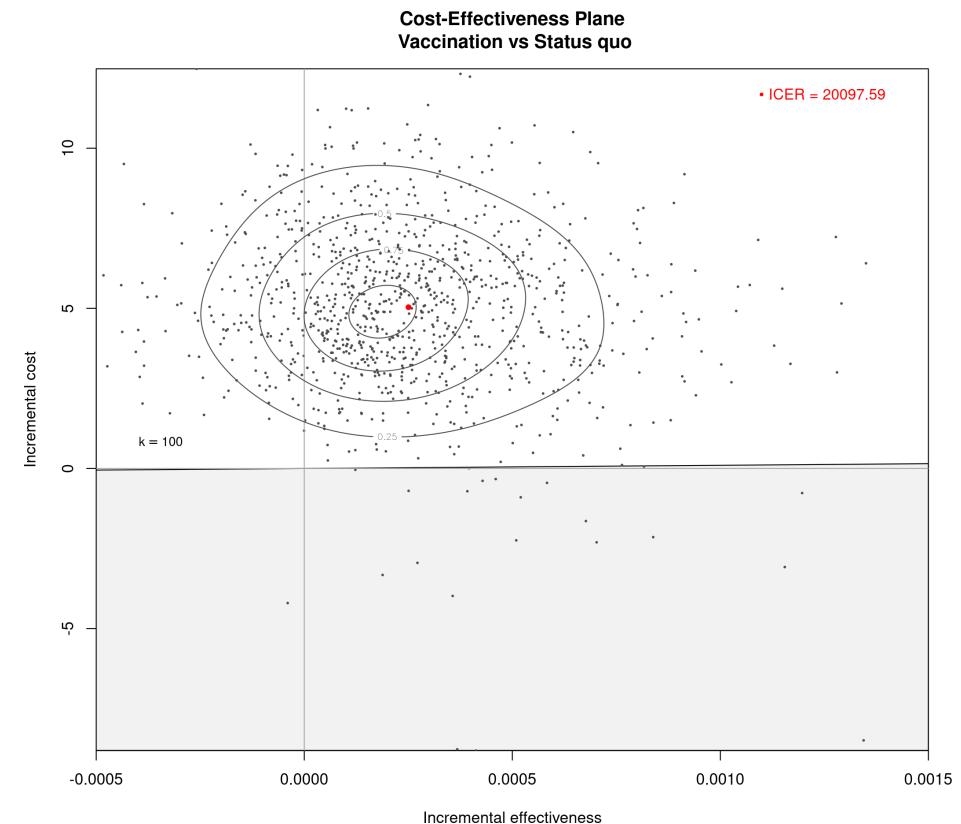
# Specialised plots

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

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

Cost-effectiveness efficiency frontier summary

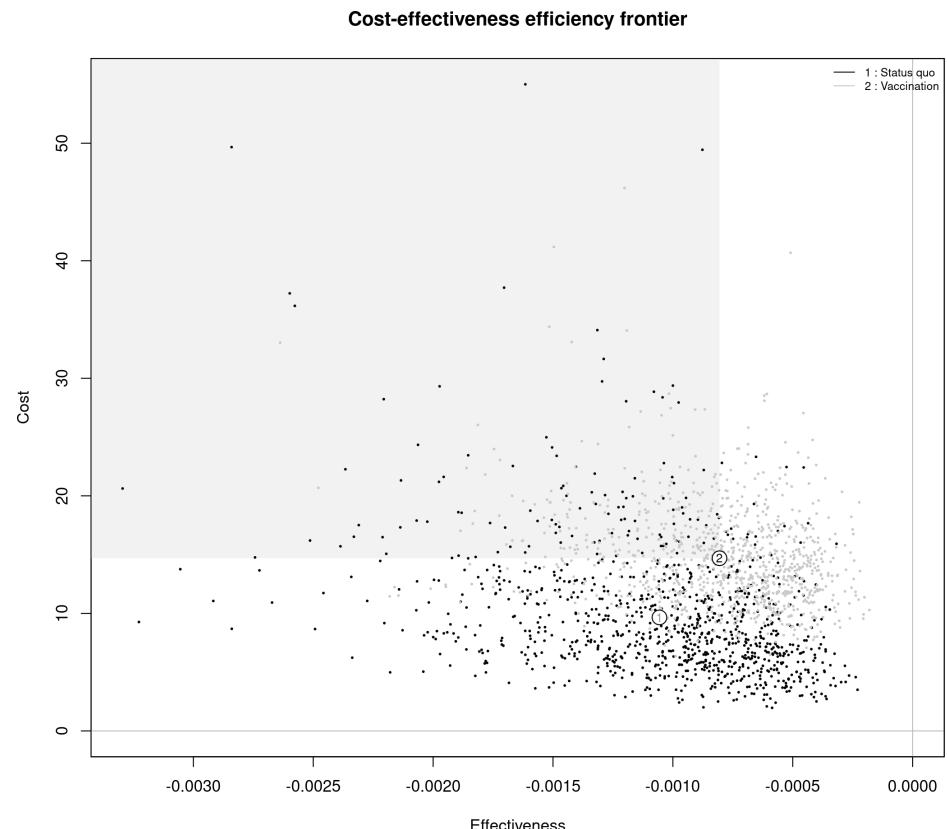
Interventions on the efficiency frontier:

	Effectiveness	Costs	Increase slope	Increase angle
Vaccination	-0.00080537	14.691	NA	↑

Interventions not on the efficiency frontier:

	Effectiveness	Costs	Dominance type
Status quo	-0.0010559	9.6555	Extended dominance

```
> 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...

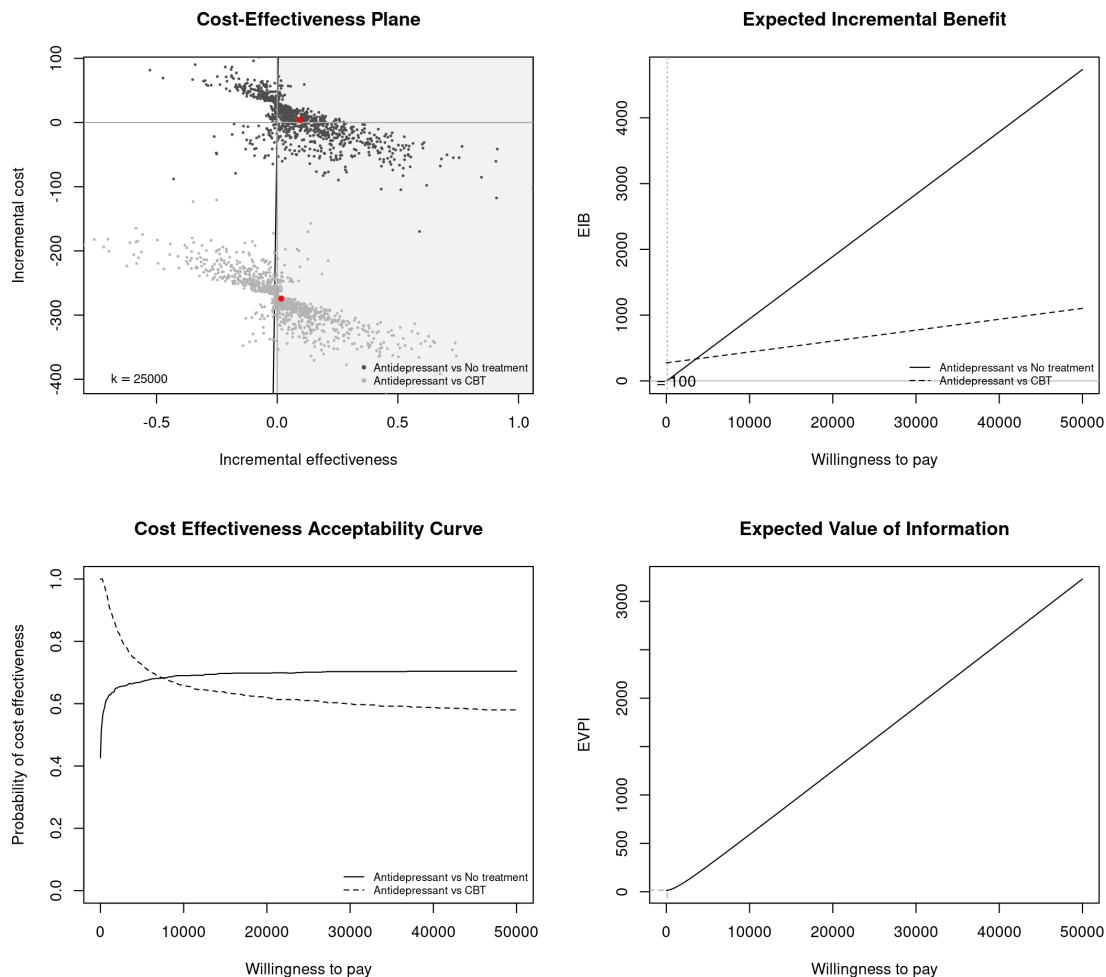
# Advanced use of BCEA

## 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...

## Probabilistic "depression model"

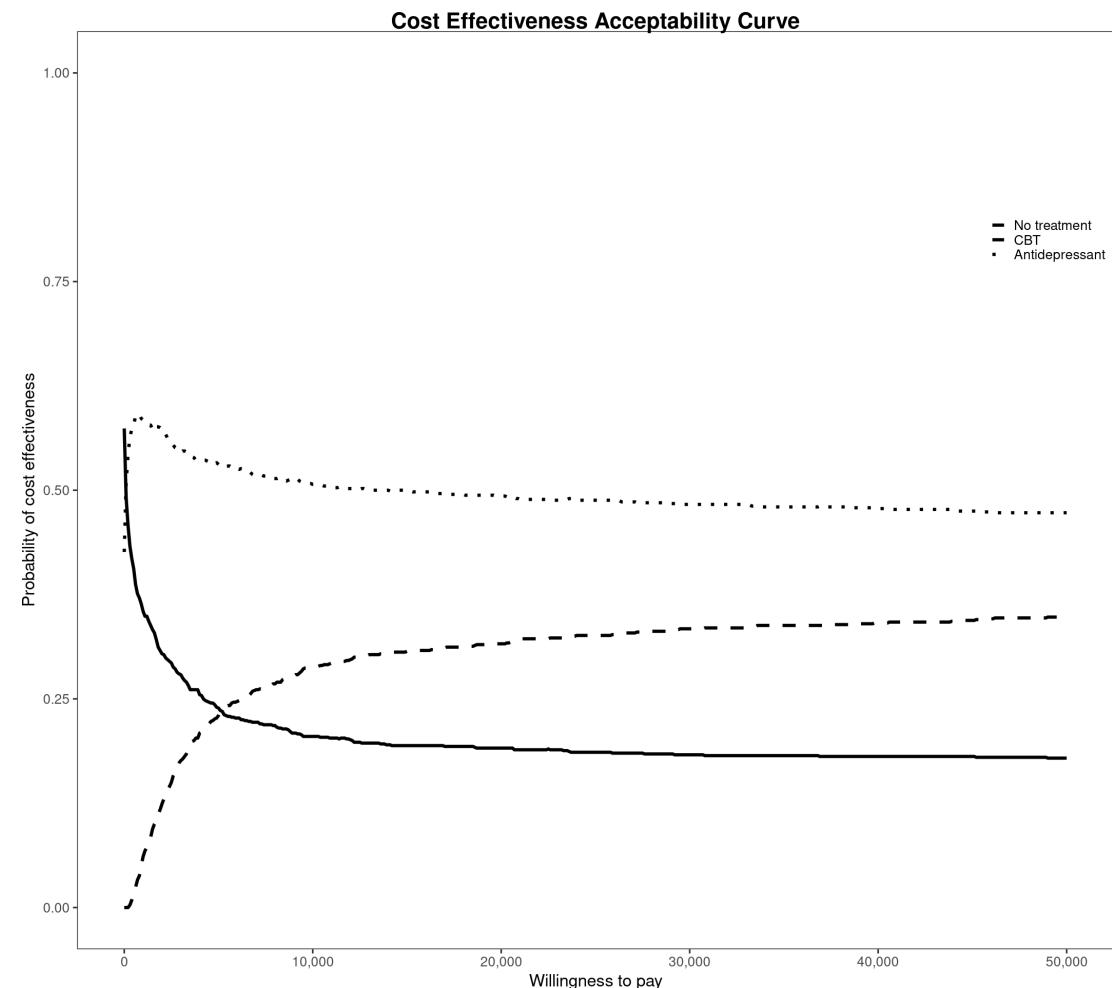
```
> # Intervention labels  
> t.names<-c("No treatment", "CBT", "Antidepressant")  
>  
> # "Standard" analysis: pairwise comparisons  
> depression.bcea = bcea(effects, costs,  
+                           interventions=t.names, ref=3)  
> # the third intervention is the reference  
>  
> # Plots the results  
> plot(depression.bcea)
```



## 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"  
+ )
```

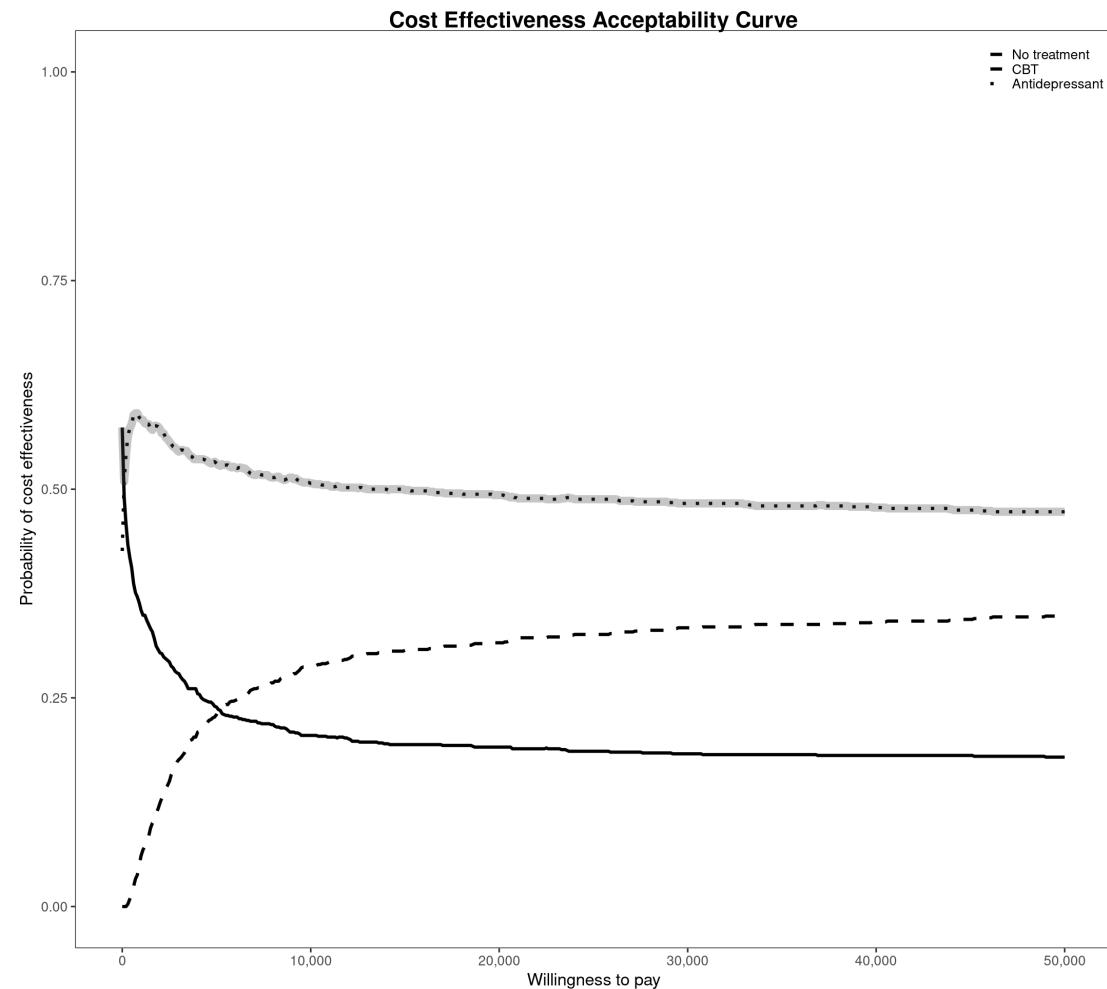
NB: In older releases of BCEA, this graph was done using the **deprecated** function `mce.plot`



## Probabilistic "depression model"

- Can use ggplot to customise the graph

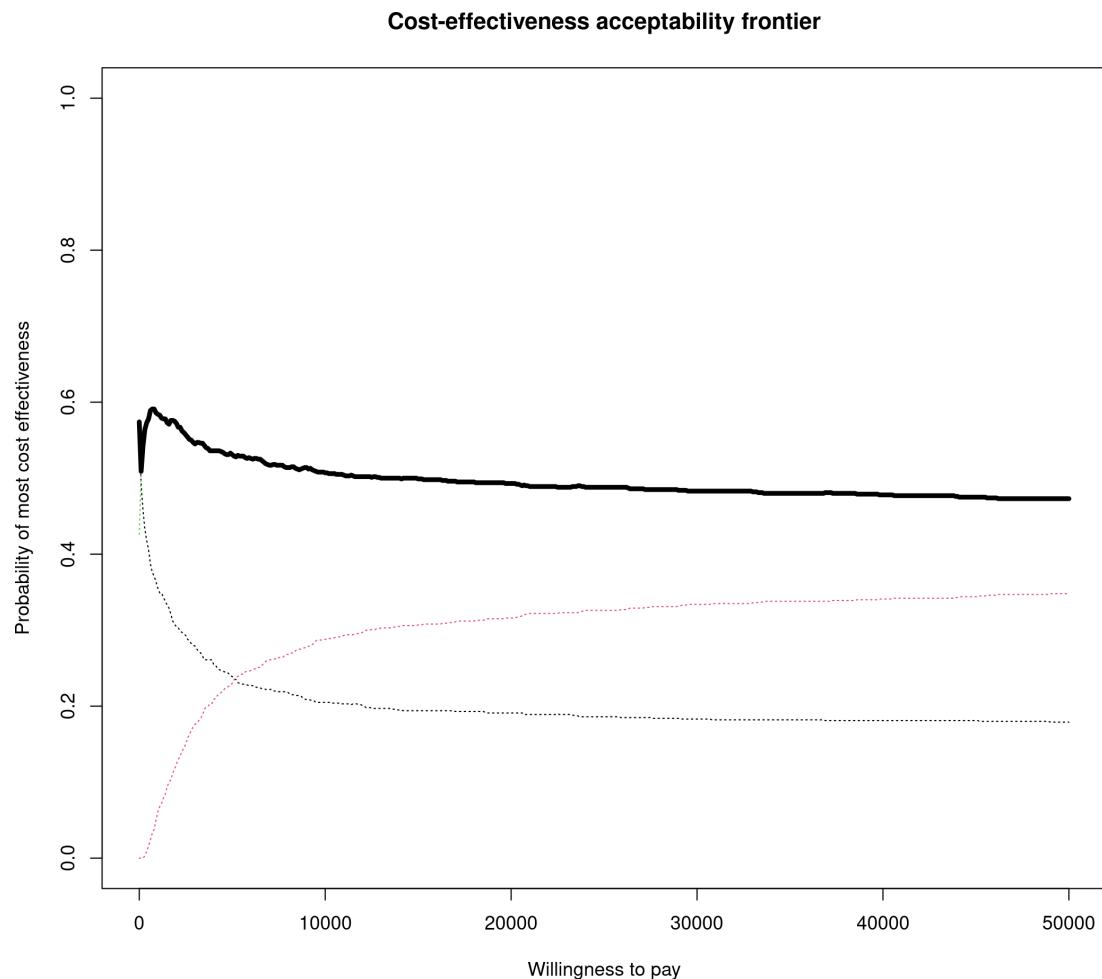
```
> ceac.plot(
+   depression.multi.ce, pos=c(1,1),
+   graph="ggplot2"
+ ) +
+   ggplot2::stat_summary(
+     fun=max, geom="line",
+     colour="grey25",
+     alpha=.3, lwd=2.5
+ )
```



## Probabilistic "depression model"

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

```
> # Specialised plot  
> ceaf.plot(depression.multi.ce)
```



- 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!)

- 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, ...

```
> # 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
+     )
```

```
> # 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.**