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@mas.to @gianlubaio

Health Economic Modeling in R: A Hands-on Introduction

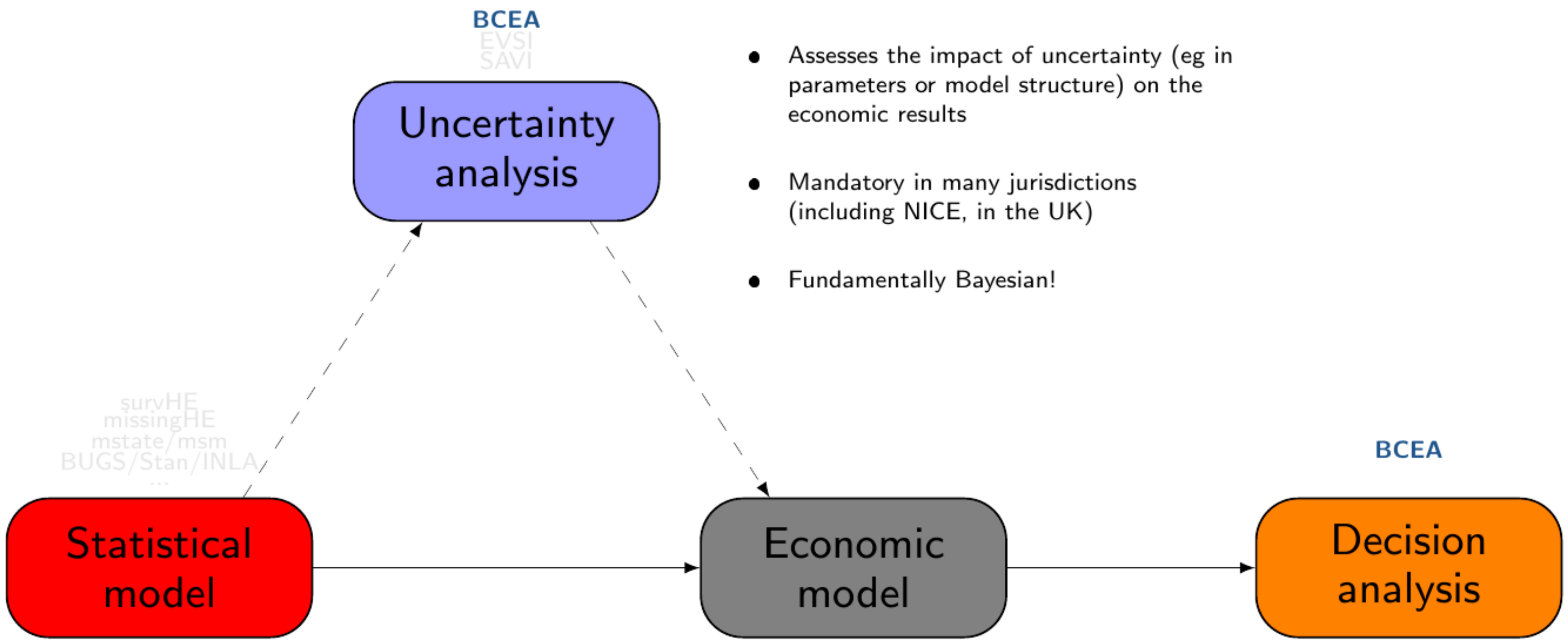
9 November 2025

Follow our departmental social media accounts + magazine "Sample Space"

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

UCL Statistical Science

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



Uncertainty analysis

- 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!

Statistical model

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

Economic model

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

Decision analysis

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

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.

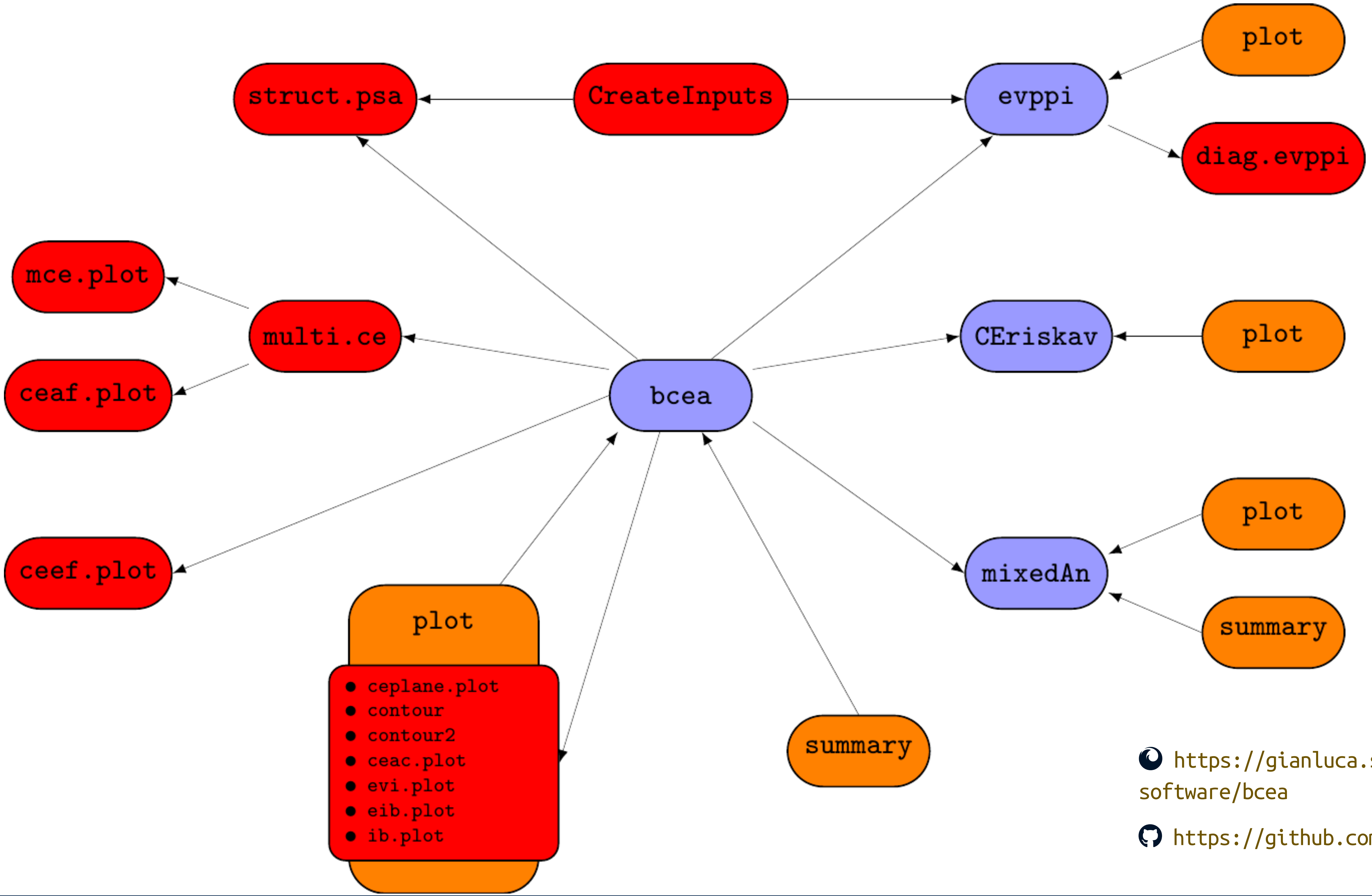
BCEA

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), etc
3. Perform uncertainty analysis
   - Standard PSA (mandatory): Cost-effectiveness Plane, CEAC, etc
   - 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?

Installation

- BCEA is available from CRAN
  - Current *stable* version: 2.4.81 (14 July 2025)
- But it is also under constant development in the GitHub repository
  - Current *development* version: 2.4.82

```r
# 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

# And the *development* version (from GitHub)
remotes::install_github("giabaio/BCEA")             # development version

# EVEN EASIER -- from r-universe.dev!
install.packages(
  'BCEA',
  repos = c('https://giabaio.r-universe.dev', 'https://cloud.r-project.org')
)
```

NB: The beauty of the GitHub and `r-universe` versions is that they can be updated on the fly and be immediately available for users!

Using BCEA

```r
library(dplyr)  # (Not necessary - but very 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 and 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

Show. Me. The. Data!

```r
# 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
inp$mat |> as_tibble()                    # "piping" ('%>%' in `tidyverse` or '|>` in newer base `R`) and formatting nicely
```

```
# A tibble: 1,000 × 56
   Adverse.events Death.1.1. Death.2.1. Death.2.2. GP.1.1. GP.2.1. GP.2.2. Hospital.1.1. Hospital.2.1. Hospital.2.2. Infected.1.1. Infected.2.1.
            <dbl>      <dbl>      <dbl>      <dbl>   <dbl>   <dbl>   <dbl>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
 1           1466          1          0          0    1664     958     230             0             1             0          5992          3401
 2           5329          1          1          0    1414     748     276             0             0             1          7471          4024
 3           5203          1          1          0     809     489      80             0             0             0          6718          4300
 4           2351          2          0          0    1761    1157     261             1             0             0          4837          3269
 5           8303          1          2          0    2472     964     432             1             1             0          4749          1894
 6           3607          1          1          0    2224    1342     260             1             0             0          4938          2976
 7           6304          4          1          1    3478    1107     591             2             1             0         11080          3547
 8           4337          1          1          1    1483     799     189             0             0             0          3867          2164
 9           5482          0          0          0    1587     798     279             0             0             0          5163          2532
10           3125          2          2          0    2578    1681     243             0             0             0          7265          4766
# ℹ 990 more rows
# ℹ 44 more variables: Infected.2.2. <dbl>, Mild.Compl.1.1. <dbl>, Mild.Compl.2.1. <dbl>, Mild.Compl.2.2. <dbl>, Pneumonia.1.1. <dbl>,
#   Pneumonia.2.1. <dbl>, 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>, beta.7. <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>, psi.1. <dbl>, psi.2. <dbl>,
#   psi.3. <dbl>, psi.4. <dbl>, psi.5. <dbl>, psi.6. <dbl>, psi.7. <dbl>, psi.8. <dbl>, q.1. <dbl>, q.4. <dbl>, q.5. <dbl>, q.6. <dbl>, q.7. <dbl>,
#   rho.2. <dbl>, xi <dbl>
```

Economic model

```r
# Defines the number of simulations considered
n.sims=inp$mat %>% nrow()                              # applies the function 'nrow' (number of rows) to the object 'inp$mat'

                                                       # NB: in R 4.1.0, can also use 'native' pipe ('|>')
                                                       # (probably a bit quicker, but in most cases, may be immaterial...)

# Process the model parameters
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

# Aggregates the model inputs to compute (e,c)
eff = -(QALYs.inf + QALYs.pne + QALYs.adv + QALYs.hosp + QALYs.death) + ...
cost = cost.GP + cost.hosp + cost.otc + ...
```

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

Cost & effects

```r
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
# ℹ 990 more rows
```

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

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

```r
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

```r
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"            
```

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

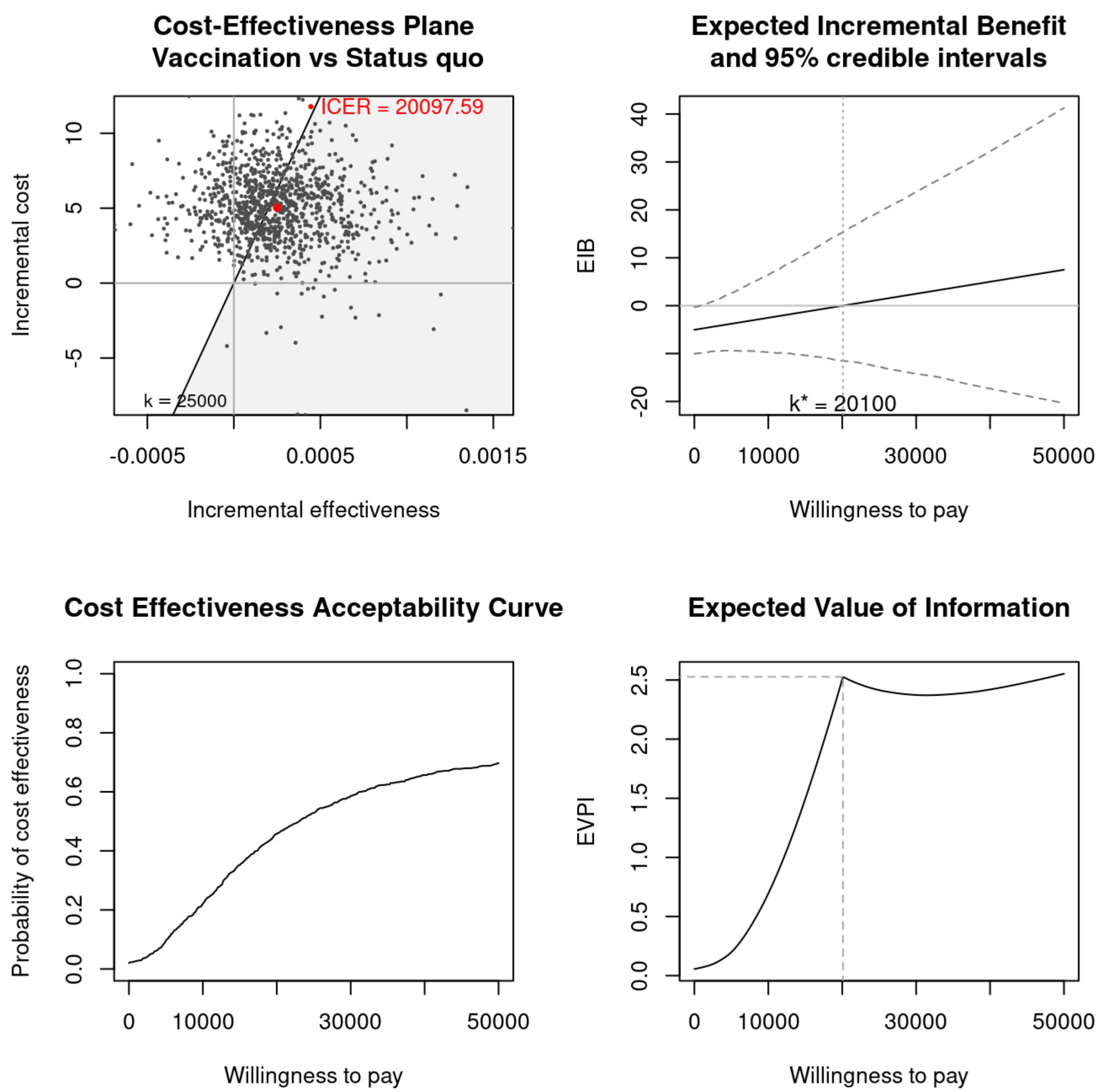
```r
# The 'summary' "method" produces a tabular output
summary(m)
```

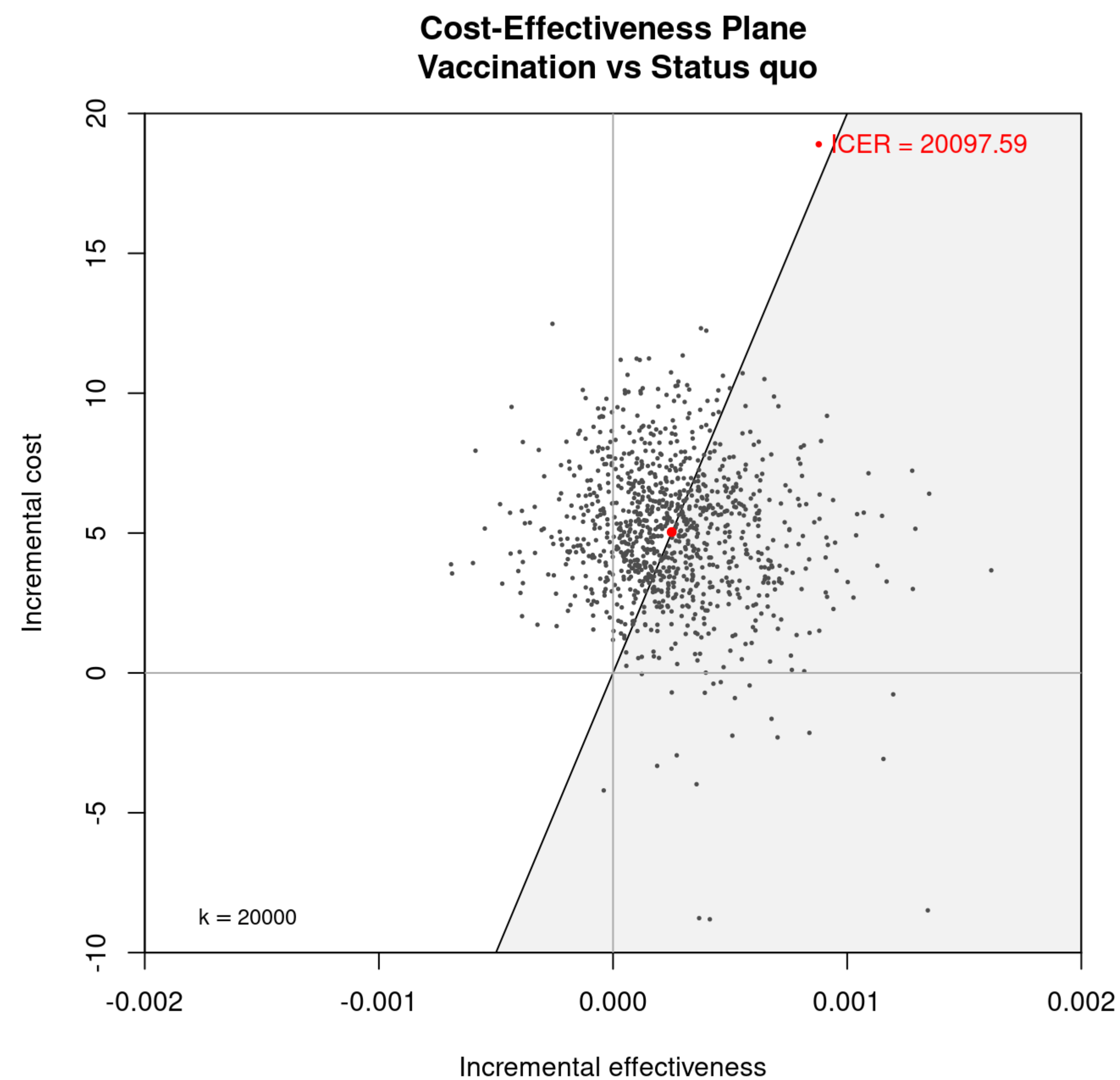
```
Cost-effectiveness analysis summary

Reference intervention:  Vaccination
Comparator intervention: Status quo

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


Analysis for willingness to pay parameter k = 25000

            Expected net benefit
Status quo               -36.054
Vaccination              -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
```

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

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

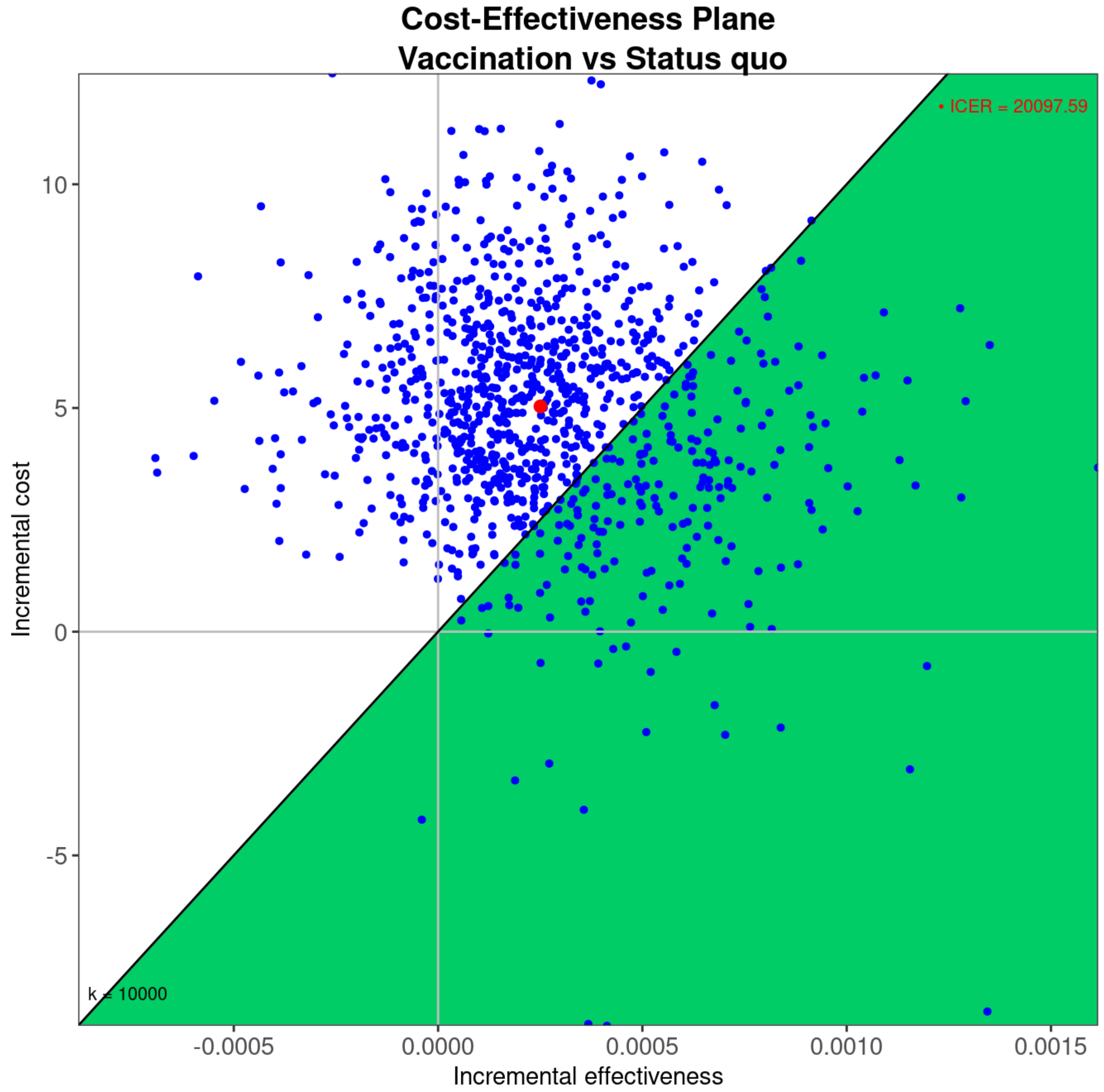
```r
ceplane.plot(m,wtp=20000,xlim=c(-.002,.002),ylim=c(-10,20))
```





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"))
```

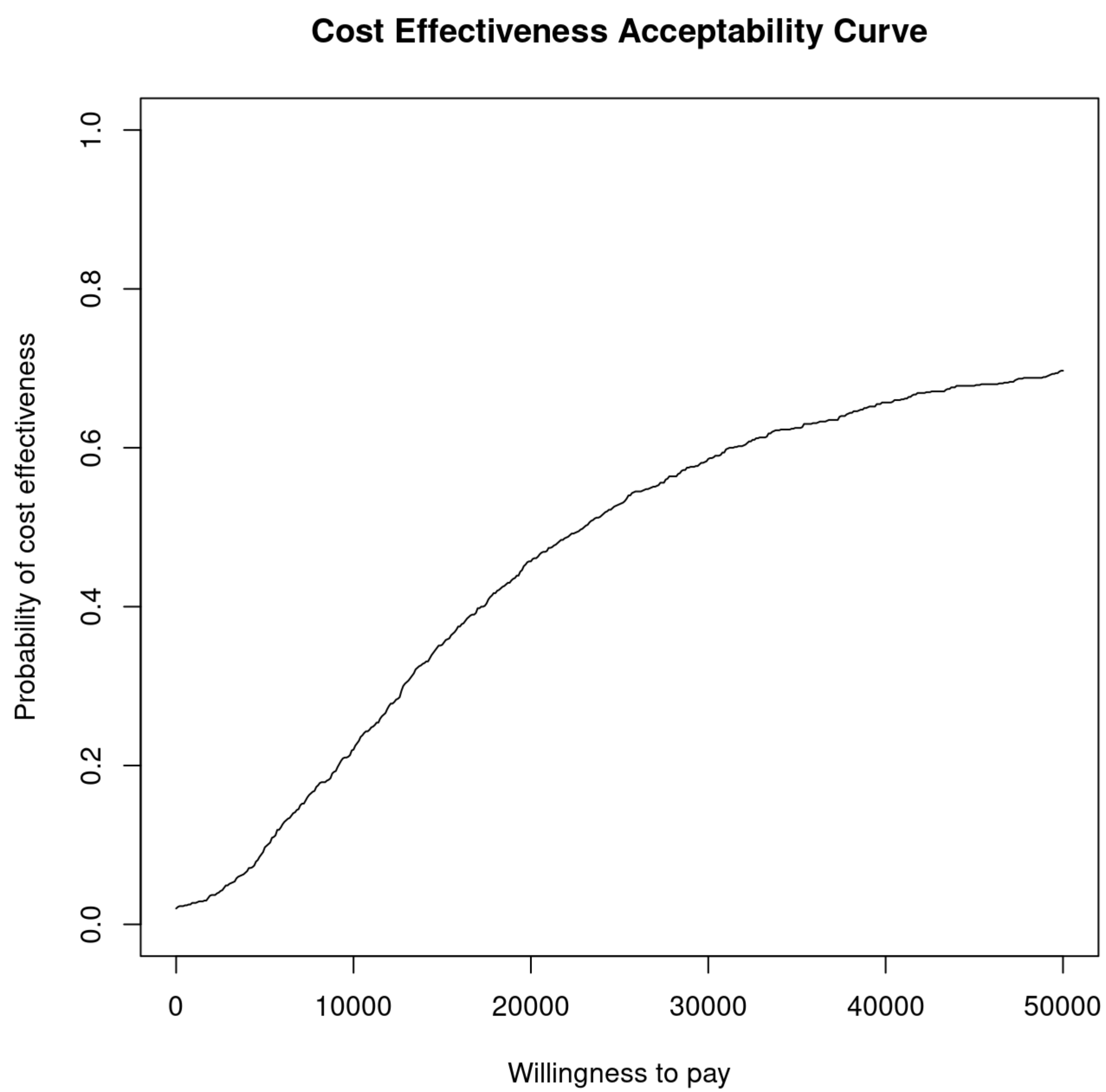


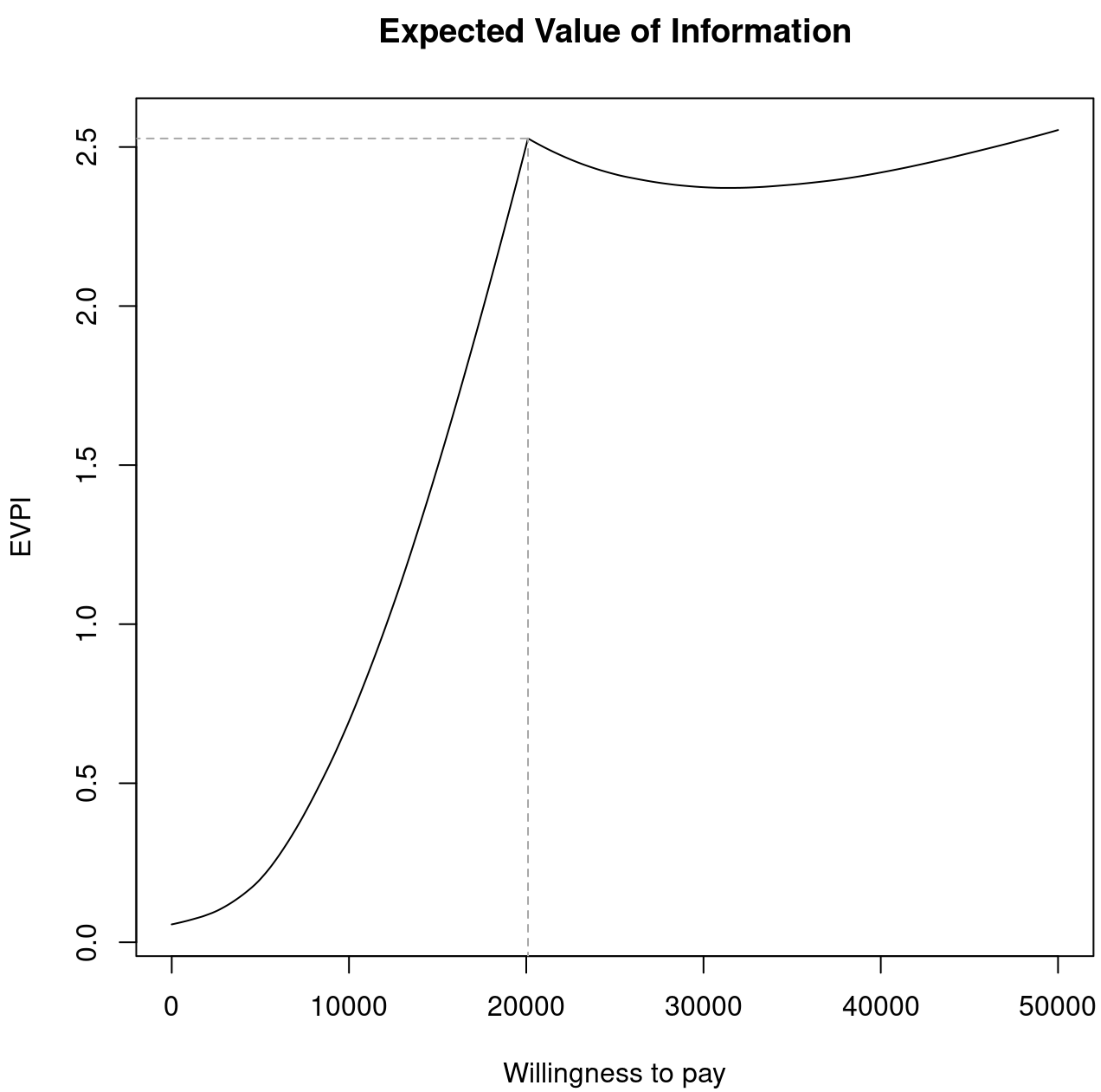
- https://ggplot2.tidyverse.org/
- https://n8thangreen.github.io/BCEA/

How does BCEA work?

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

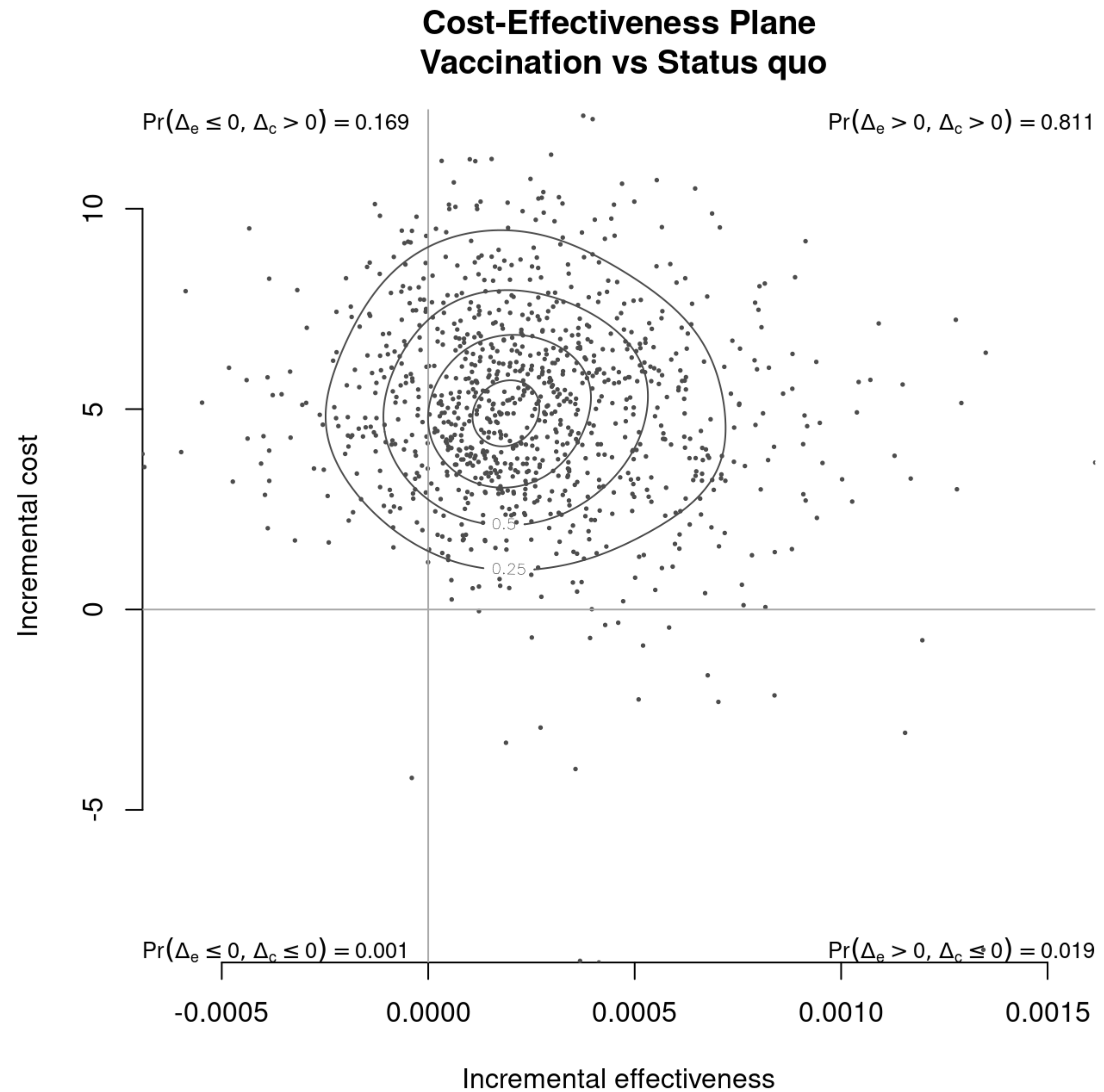
```
# Plots the Expected Value of Partial Information (EVPI)
evi.plot(m)
```





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

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

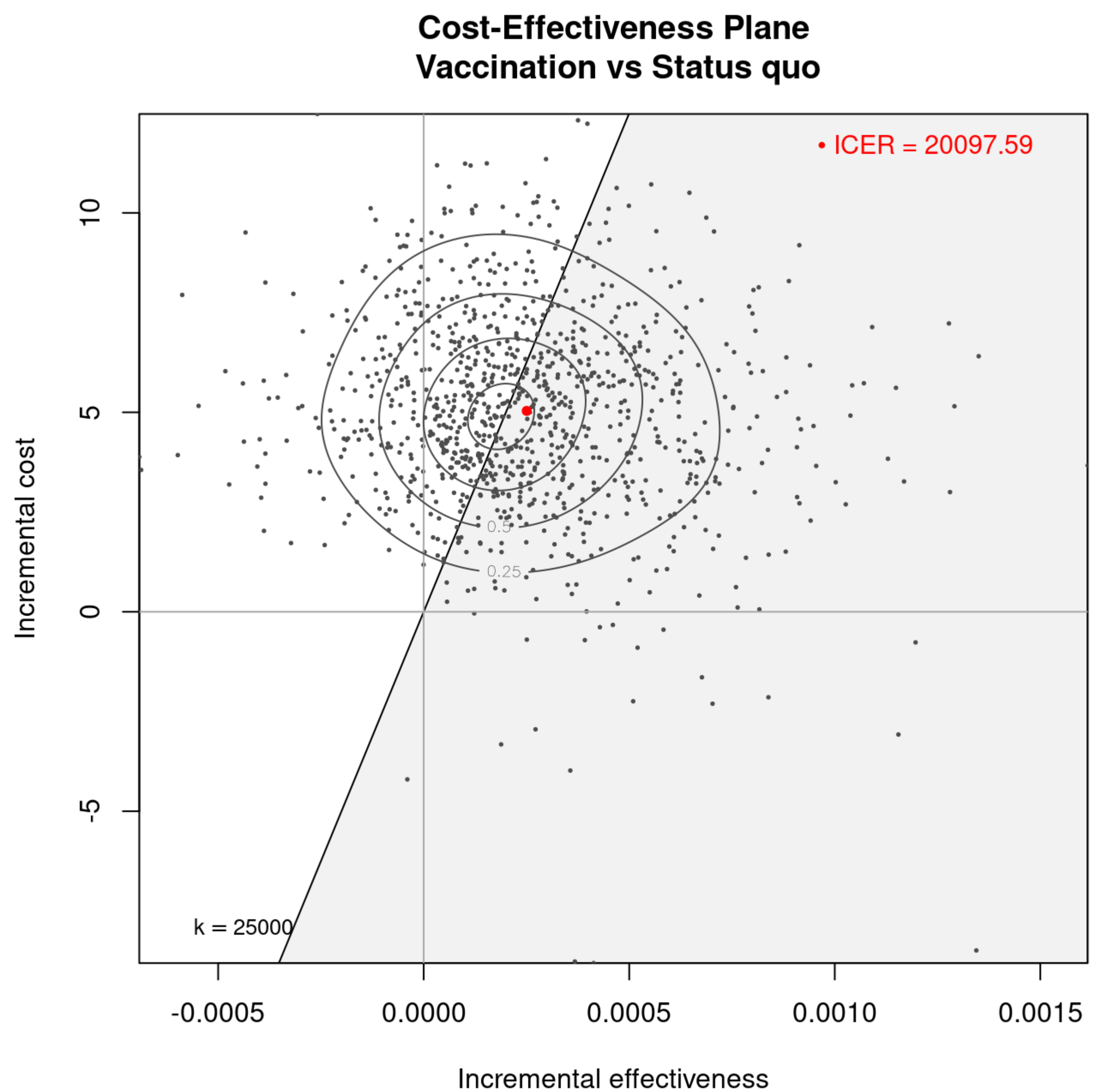


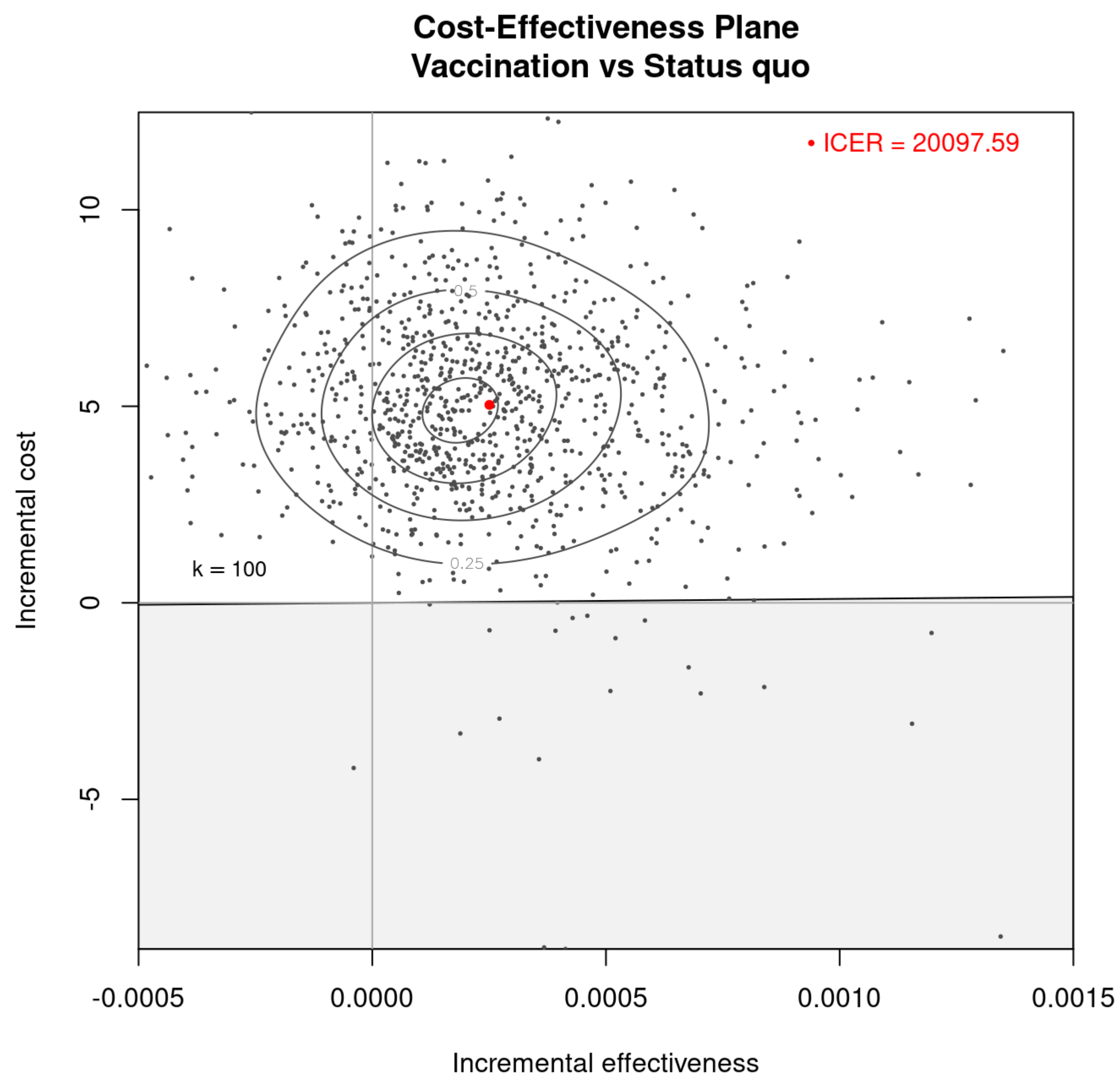
Specialised plots

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

```r
contour2(m)
```

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



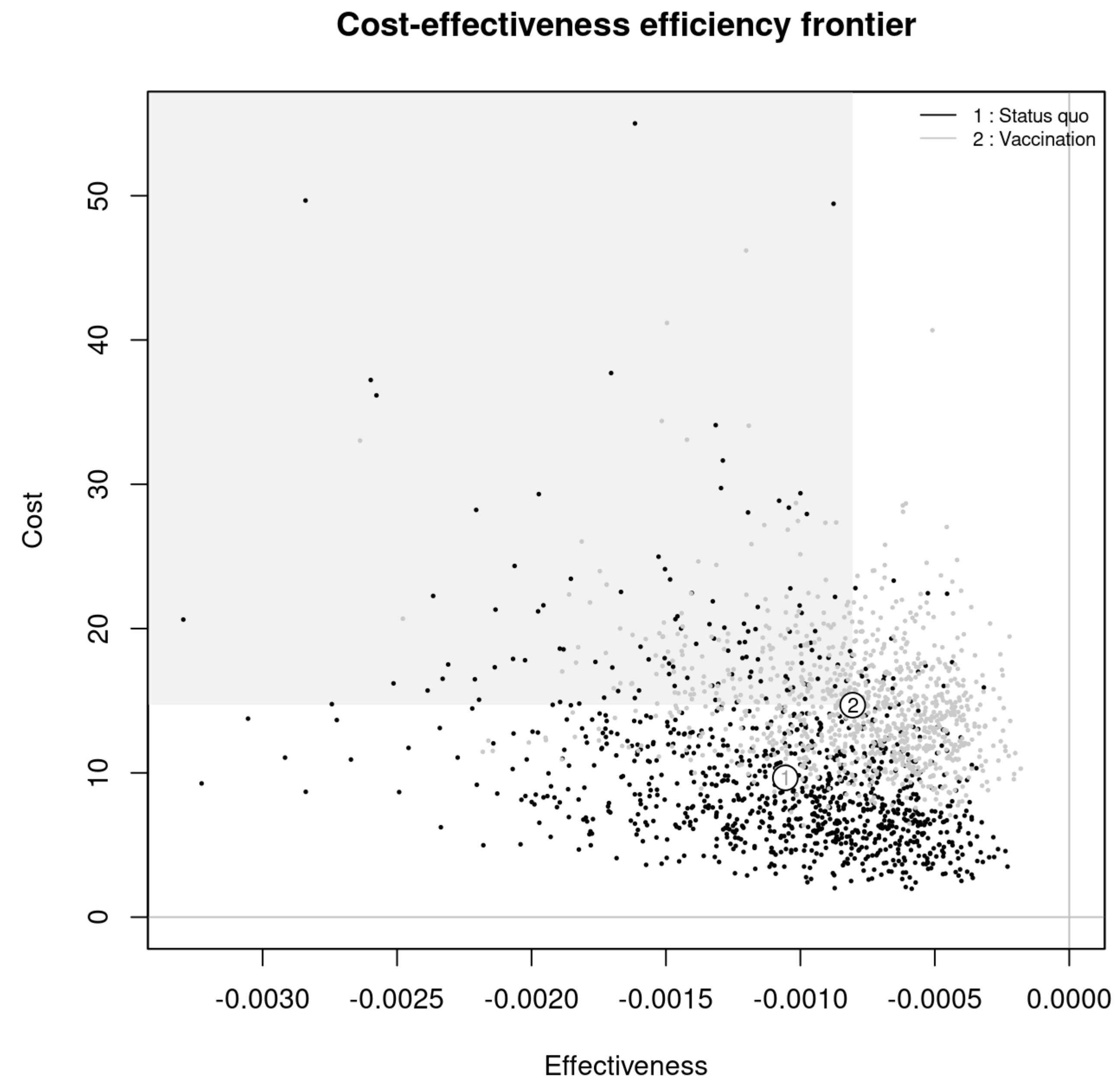


Cost-effectiveness efficiency frontier

```r
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
```

```r
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

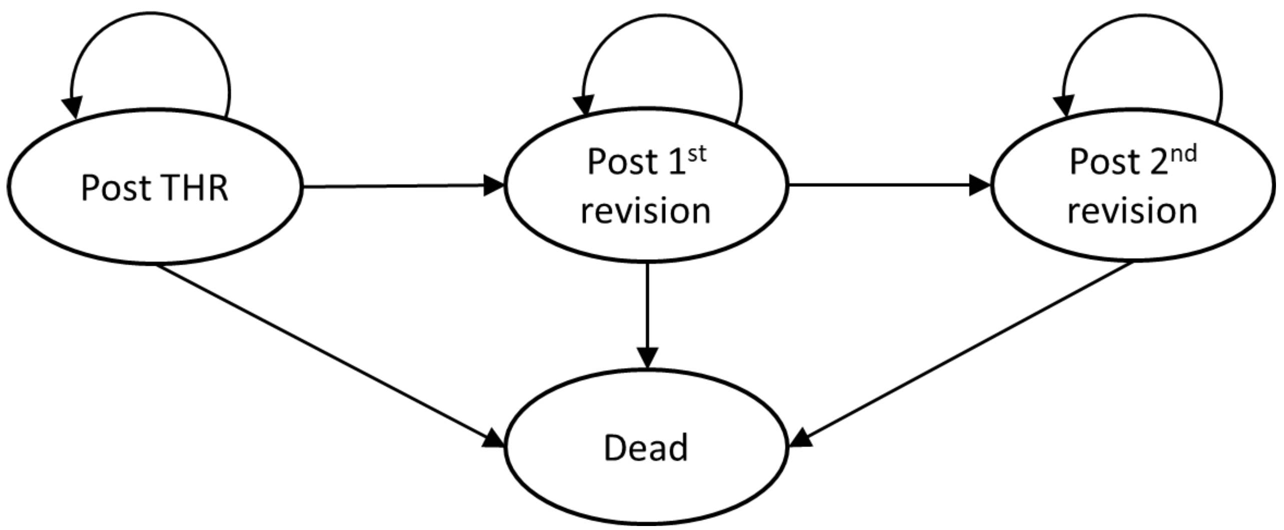
```
# "Opens" the graphical device
pdf("NAME_OF_THE_FILE",width=`8`,height=`8`)      `# for 'pdf', units are in inches`
# Makes the plot
ceplane.plot(`BCEA_OBJECT`)                        `# of course, specify whatever name you've chosen when creating the object...`
# "Closes" the graphical device
dev.off()


# "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

Hip replacement

- Four interventions: Cemented, Uncemented, Hybrid and Reverse Hybrid
  - Cemented: Cemented Metal-on-Plastic Small head
  - Uncemented: Uncemented Ceramic-on-Plastic Small head
  - Hybrid: Hybrid Ceramic-on-Plastic Large head
  - Reverse hybrid: Reverse hybrid Ceramic-on-Plastic Small head
- Only consider males aged 65-74 years old (choose start at 65 years old)
- 1-year cycle length, 30-year time horizon
- Probability of death increases with time as cohort ages
- Built on real model in Fawsitt 2019



Quick exercise...

a. Use the `R` function `read.csv` to load the costs and effects matrices and apply as.matrix to ensure correct format for `BCEA`

```
costs <- as.matrix(read.csv("total_costs.csv"))
effects <- as.matrix(read.csv("total_effects.csv"))
```

b. Use the `BCEA` function `bcea` to create a `BCEA` object and then `summary` method to try to identify the most cost-effective implant for hip replacement at £20,000/QALY (include the `wtp=...` argument). You will need to define a treatment names vector

```
treats = c("cemented", "uncemented", "hybrid", "reverse_hybrid")
```

c. Use the `BCEA` function `ceplane.plot` to plot the cost-effectiveness plane. Try changing the x and y axes with the arguments `xlim` and `ylim` to improve visualisation.

d. Apply the `BCEA` function `multi.ce` to your `BCEA` object and then apply the function `ceac.plot` to the resulting multiple comparison `BCEA` object. Which implant has the highest probability of being cost-effective at £20,000/QALY and £30,000/QALY?

Hip replacement — Solutions (b)

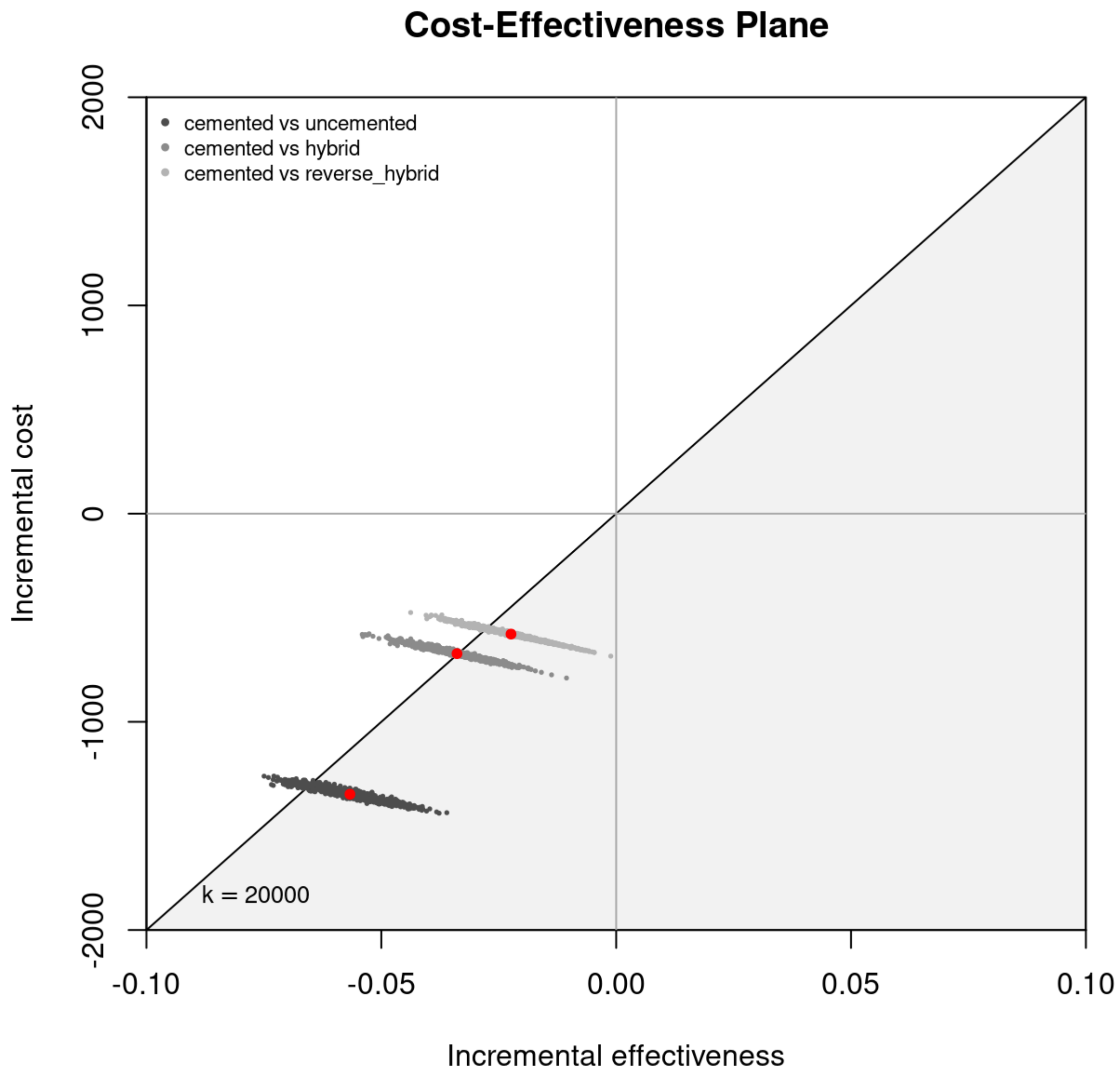
```r
# Creates the BCEA object
hips_bcea = bcea(
  # Selects the object containing the simulations for the effects
  e = effects,
  # Selects the object containing the simulations for the costs
  c = costs,
  # Selects the "reference" intervention
  ref = 1,
  # Specifies the vector of treatment lables
  interventions = treats,
  # Specifies the maximum value in the grid of wtp-s
  Kmax = 50000
)

# Prints the summary table for the economic analysis
summary(hips_bcea)
```

```
Cost-effectiveness analysis summary

Reference intervention:  cemented
Comparator intervention(s): uncemented
                          : hybrid
                          : reverse_hybrid

Optimal decision: choose cemented for k < 19900
                         hybrid for 19900 <= k < 29700
                         uncemented for k >= 29700


Analysis for willingness to pay parameter k = 25000

               Expected net benefit
cemented                     261389
uncemented                   261458
hybrid                       261564
reverse_hybrid               261370

                                EIB  CEAC  ICER
cemented vs uncemented      -69.158 0.351 23780
cemented vs hybrid         -174.676 0.180 19845
cemented vs reverse_hybrid   18.978 0.540 25848

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

EVPI 36.456
```

- Net benefits are very similar but hybrid had highest net benefit
- ICER of cemented vs hybrid is lowest

Hip replacement — Solutions (c)

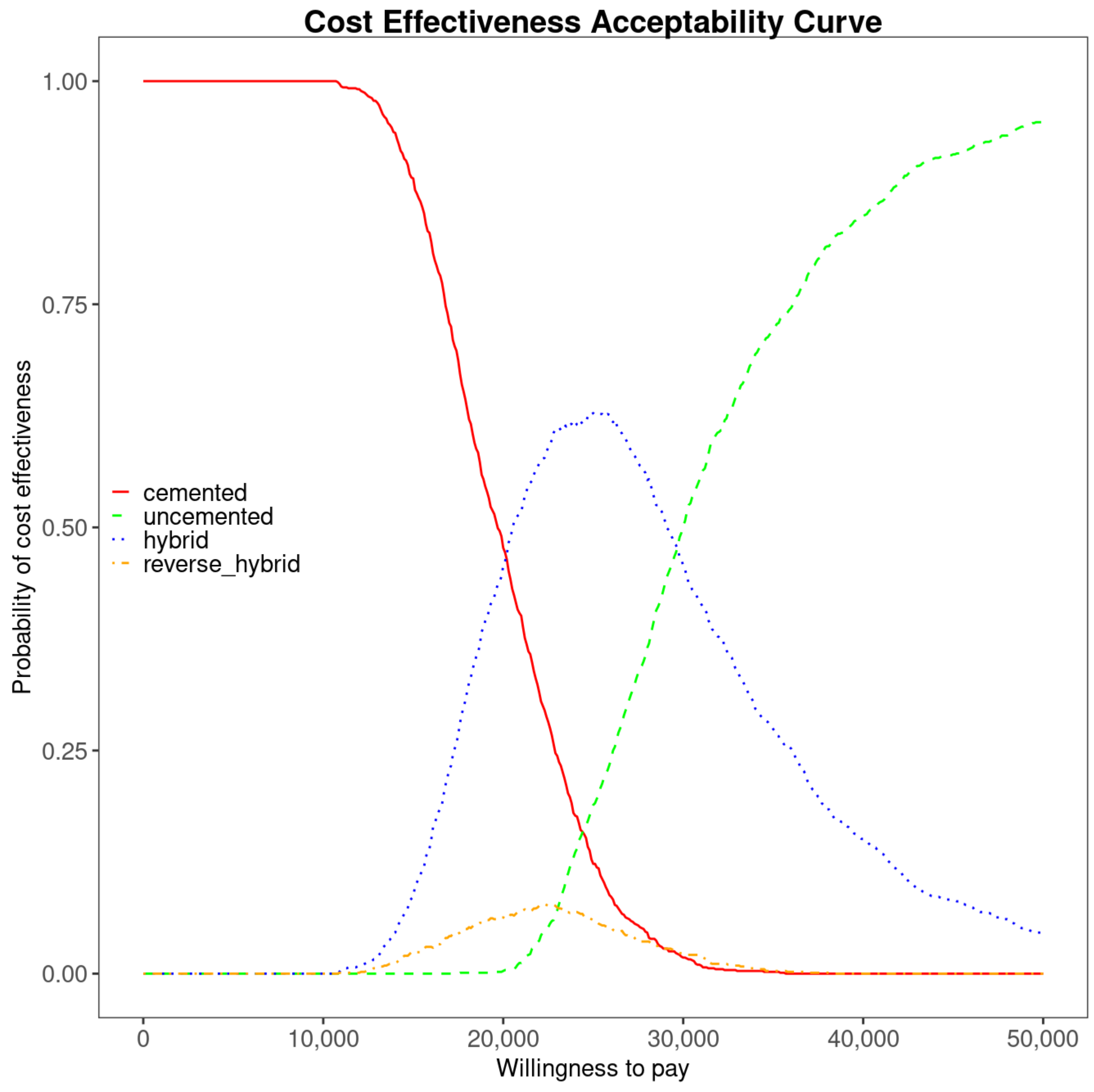
```r
# Plots the cost-effectiveness plane
ceplane.plot(
  hips_bcea,
  wtp = 20000,
  xlim = c(-0.1, 0.1),
  ylim = c(-2000, 2000)
)
```



- Limits chosen manually
- Note that comparisons are references vs intervention
- Appears there is some probability that each intervention is cost-effective compared to cemented (higher effects but also higher costs)

Hip replacement — Solutions (d)

```
# Plots a CEAC
hips_multi_ce = multi.ce(hips_bcea)
ceac.plot(
  hips_multi_ce,
  graph = "ggplot",
  line = list(
    color = c("red", "green", "blue", "orange")
  ),
  pos = c(0, 0.50)
)
```



- Hybrid is most cost-effective in the £20,000/QALY to £30,000/QALY range
- If more willing to spend more money, uncemented becomes most cost-effective