www.ispor.org

ISPOR
Improving healthcare decisions

Health Economic Modeling in R: A Hands-on Introduction

Presented by:

Felicity Lamrock, PhD
Gianluca Baio, PhD
Howard Thom, PhD, MSc
Rose Hart, PhD

ISPOR Europe 2023 | Sunday, 12 November 2023

Health Economic Modeling in R: A Hands-on Introduction

Sunday 12th November 08:00 Copenhagen

QUEEN'S UNIVERSITY BELFAST

UCL

University of BRISTOL

Lumanity

ISPOR

Copyright, Trademark, and Confidentiality

ISPOR

Antitrust Compliance Statement

- ISPOR has a policy of strict compliance with both United States, and other applicable international antitrust laws and regulations.
- Antitrust laws prohibit competitors from engaging in actions that could result in an unreasonable restraint of trade.
- ISPOR members (and others attending ISPOR meetings and/or events) must avoid discussing certain topics when they are together including, prices, fees, rates, profit margins, or other terms or conditions of sale.
- Members (and others attending ISPOR meetings and/or events) have an obligation to terminate any discussion, seek legal counsel's advice, or, if necessary, terminate any meeting if the discussion might be construed to raise antitrust risks.
- The Antitrust policy is available on the ISPOR website.

The Antitrust policy is available on the ISPOR website at ispor.org/antitrust.

Welcome!

- 12 November 2023: Health Economic Modeling in R: A Hands-on Introduction
- LEVEL - Introductory
  TRACK - Economic Evaluation
  LENGTH - 4 Hours (with coffee break!)
- PREREQUISITE: Familiarity with R coding is not required for attendance, however it would be beneficial and background material can be provided to those with little R experience.
- Attendees will require a laptop with RStudio (v1.1.0 or higher) and R (v3.5.0 or higher) downloaded and installed

Thank you to ISPOR!

ISPOR
Improving healthcare decisions

Who are we?

Dr Felicity Lamrock

Professor Gianluca Baio

Dr Howard Thom

Dr Rose Hart

QUEEN'S UNIVERSITY BELFAST

UCL

University of BRISTOL

Lumanity

Agenda

08:00 – 08:05	Welcome and Introductions	Felicity Lamrock
08:05 – 09:00	Introduction to R for Health Economics using BCEA	Gianluca Baio
09:00 – 10:00	Discrete time Markov models (deterministic)	Felicity Lamrock
Coffee Break		
10:15 – 11:15	Discrete time Markov models (probabilistic)	Howard Thom
11:15 – 12:00	Additional Useful Packages for R Modelling	Rose Hart

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, Copenhagen, Denmark

12 November 2023

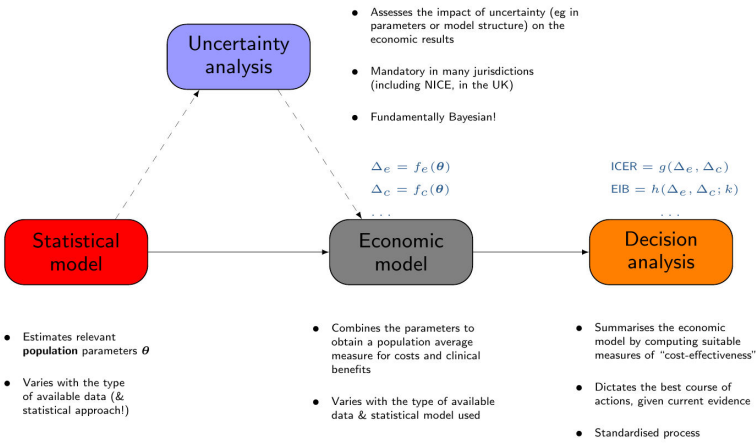
UCL Statistical Science

Disclaimer

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

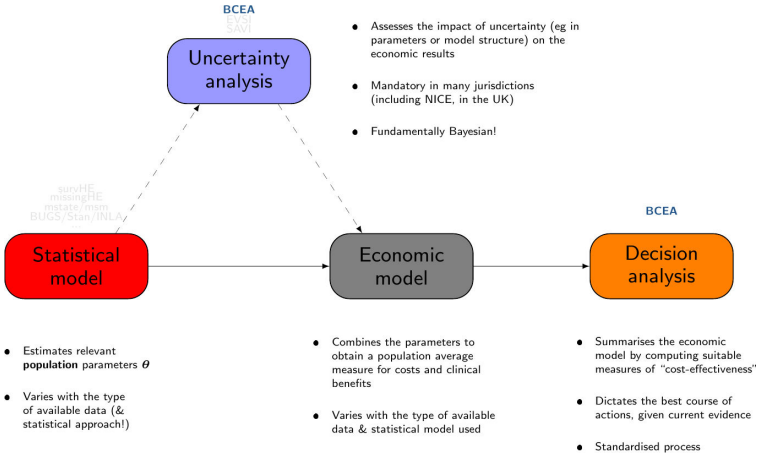
Health technology assessment (HTA)

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



Health technology assessment (HTA)

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



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**

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!

BCEA

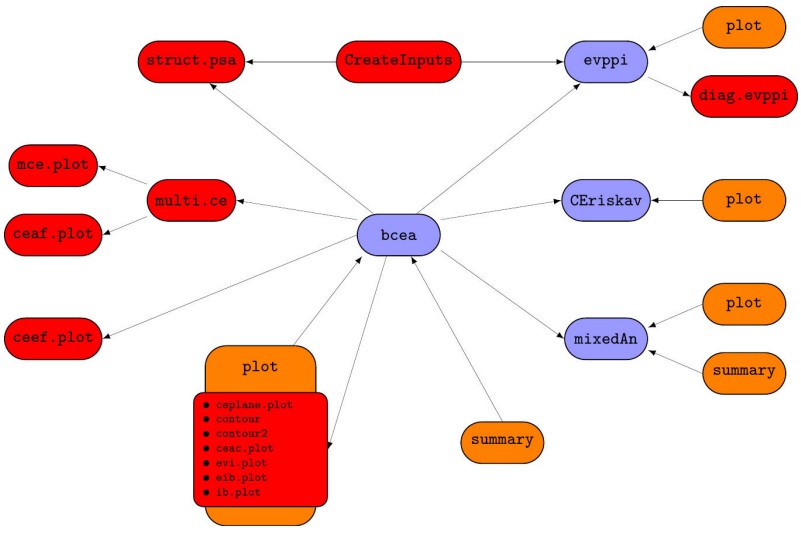
A R package for (Bayesian) cost-effectiveness analysis

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)

BCEA

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?

UCL

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
# ℹ 990 more rows
```

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] "eib"           "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)

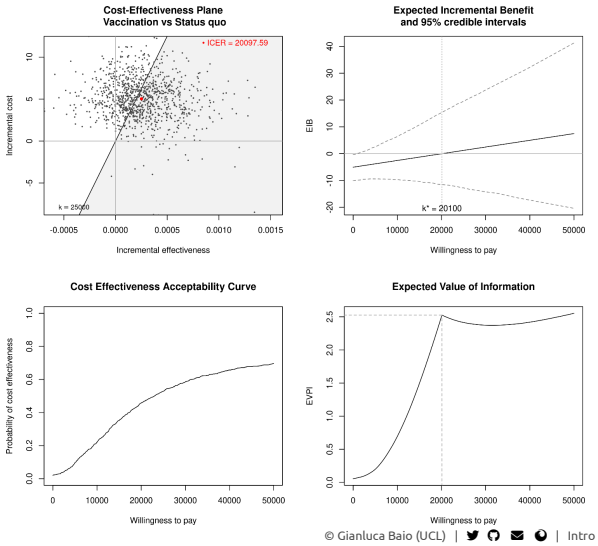
```
# The 'summary' "method" produces a tabular output
summary(m)
```

```
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
```

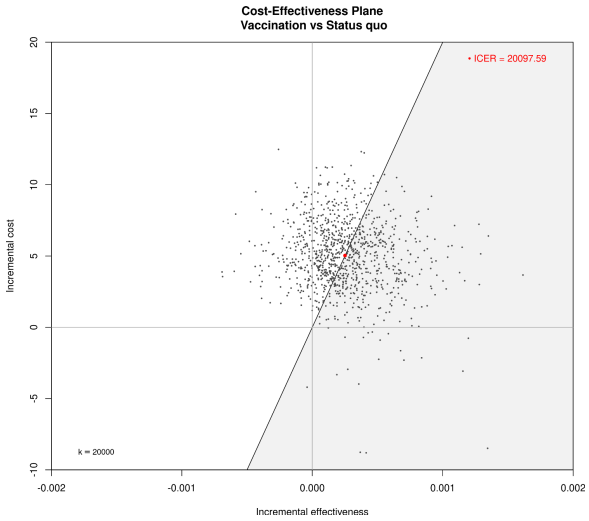
How does BCEA work?

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

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

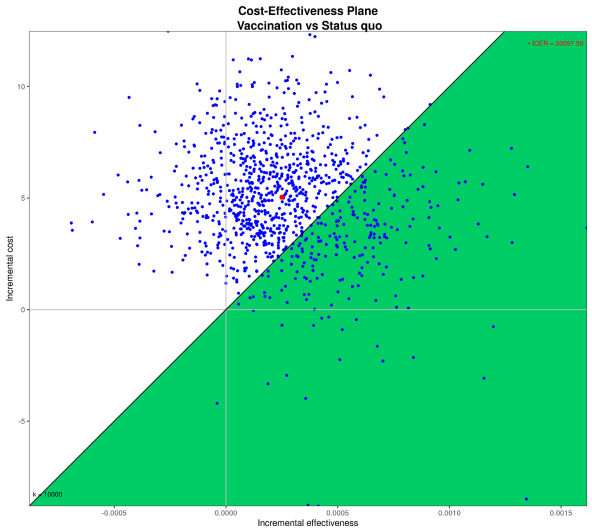



```
ceplane.plot(m,wtp=20000,xlim=c(-.002,.002),ylim=c(-10,20))
```



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"))
```

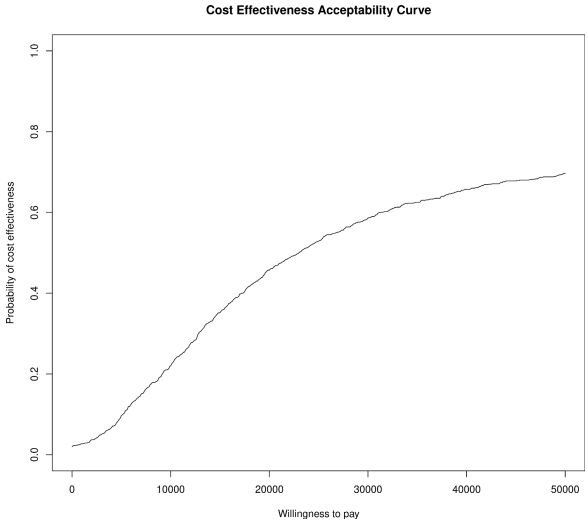


https://ggplot2.tidyverse.org/

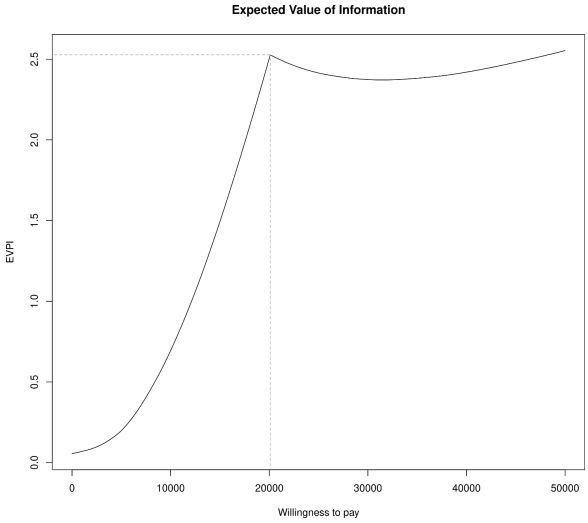
https://n8thangreen.github.io/BCEA/

How does BCEA work?

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



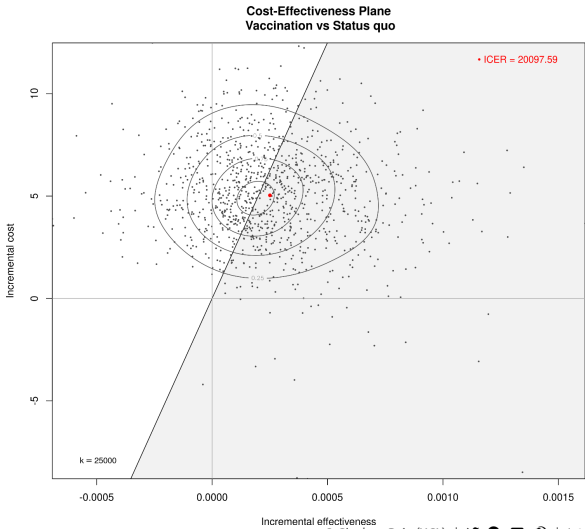

```
# Plots the Expected Value of Partial Information (EVPI)
evi.plot(m)
```



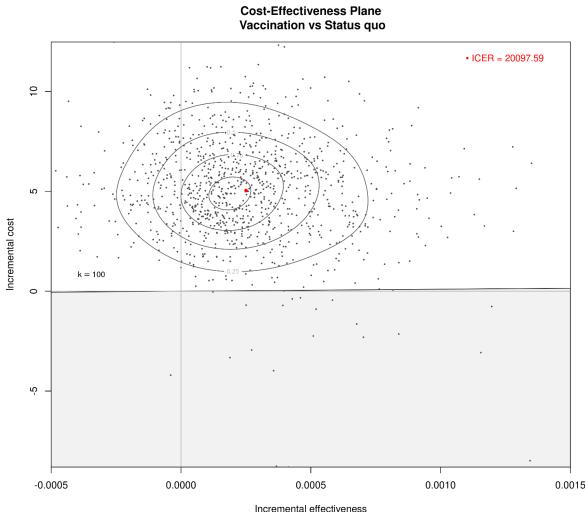
Specialised plots

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

```
contour2(m)
```




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



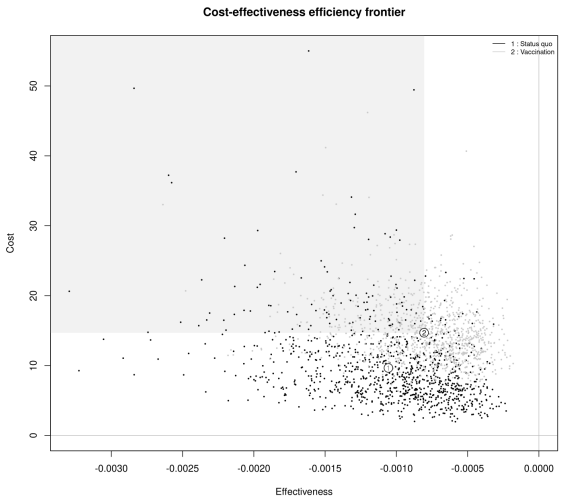
Specialised plots

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             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

Multiple treatment comparisons

Probabilistic "depression model"

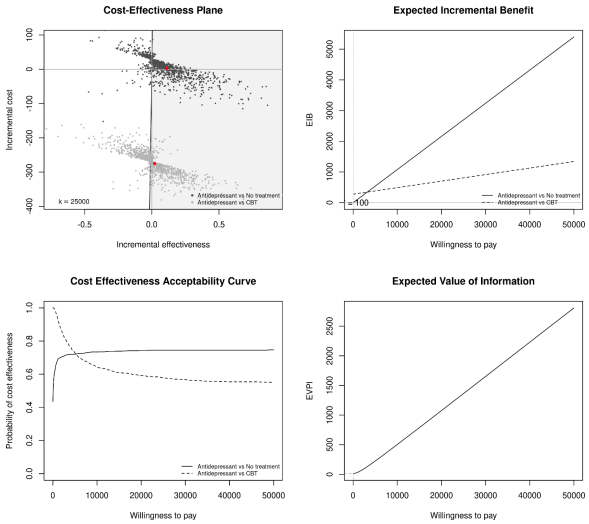
- Fictional model comparing antidepressants to cognitive behaviour therapy (CBT) and no treatment in people with depression
- Statistical modelling based on evidence synthesis
  - Benefits: based on QALYs
  - Costs: associated with treatments and various resources use
- Economic modelling: two matrices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

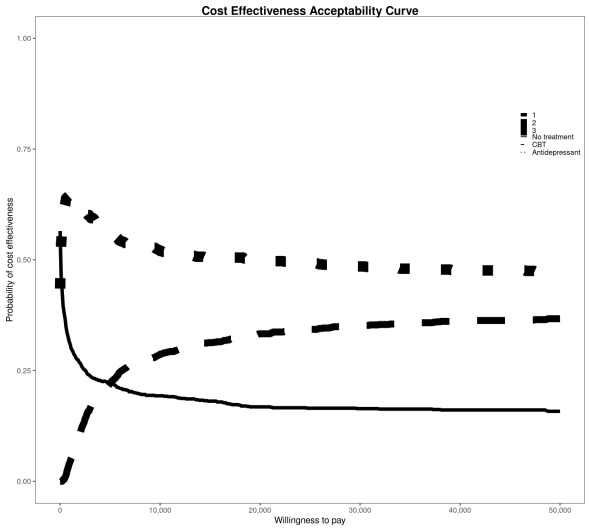


Multiple treatment comparisons

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`

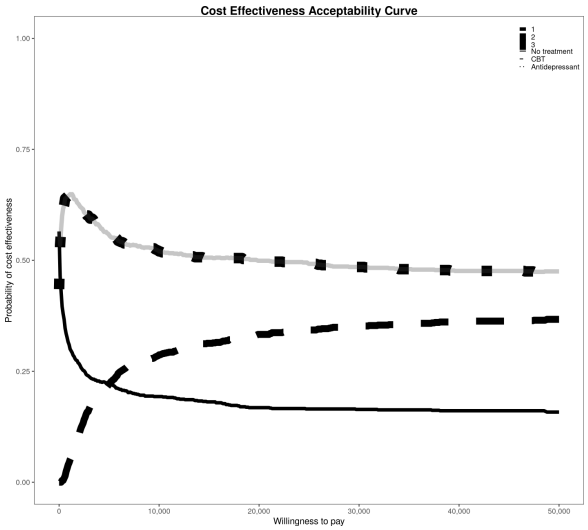


Multiple treatment comparisons

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)
```

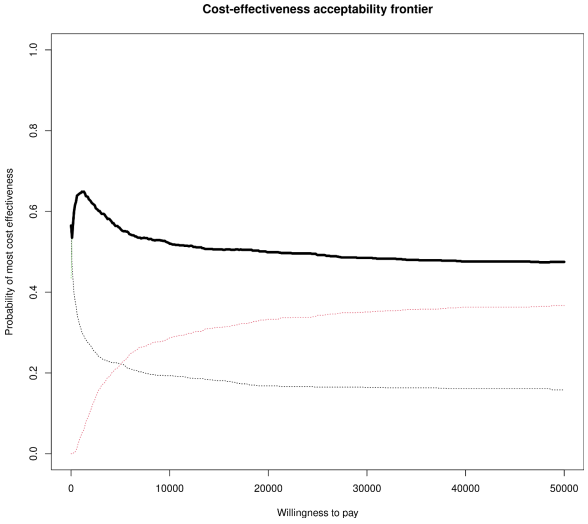


Multiple treatment comparisons

Probabilistic "depression model"

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

```
# Specialised plot
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

```
# 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)

QUEEN'S UNIVERSITY BELFAST | SCHOOL OF MATHEMATICS AND PHYSICS

Discrete Time Markov Models - deterministic

Dr Felicity Lamrock
Mathematical Sciences Research Centre

f.lamrock@qub.ac.uk

QUEEN'S UNIVERSITY BELFAST

Acknowledgement

These materials were adapted with permission from those used on the University of Bristol Medical School short courses "Economic Evaluation Modelling Using R" and "Introduction to Economic Evaluation"

University of BRISTOL

Overview

- Markov modelling
- Smoking cessation Markov model
  - Cohort simulation
  - Costs/QALYs
- Coding the smoking cessation Markov model in R
- Practical

Overview

- Markov modelling
- Smoking cessation Markov model
  - Cohort simulation
  - Costs/QALYs
- Coding the smoking cessation Markov model in R
- Practical

Multi-State Models

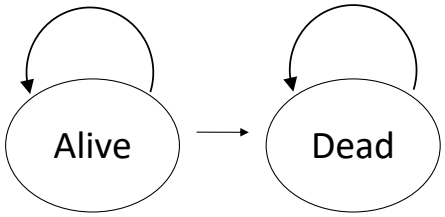
Used for analysing events that **repeat** *(e.g. mental health),* or events that play out over **time** *(e.g. cancer, heart disease)*

At any point in time individuals are in one of a finite set of states of health

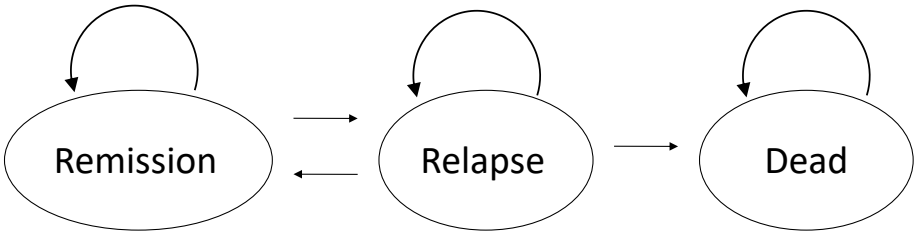
'Clinical' events are described by movements (**transitions**) between these states

Examples

(a)



(b)



Markov Models

Markov Models are a special type of Multi-State Model

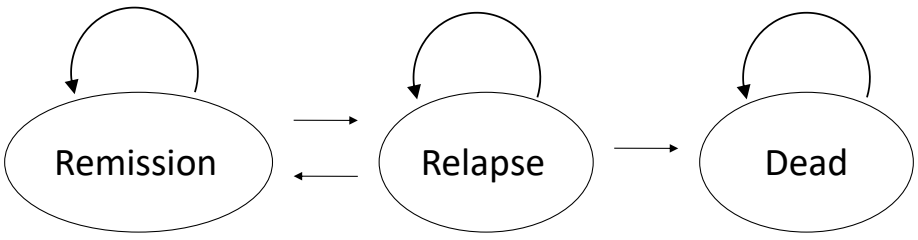
"Memoryless" property: future depends on the past only through current state

No need to remember previous movements between states

Assume rate of movement between any 2 states is constant over time

No need to remember how long been in state

Examples



Discrete-time Cohort Markov Models

Cohort models consider only aggregate behaviour of patient groups, **averaging over any individual behaviour**

Movements of patient groups between health states determined by **transition probabilities**

Here transition probabilities are **independent of time in state or past patient history**

9

Transition Probabilities

Movement through the health states is governed by the transition probabilities

Defined over a specific time period (e.g. the probability of death over 1 year)

In economic models this time period is called the **cycle length**

Transition probabilities from a state **MUST SUM TO 1**

10

Discounting

Costs and benefits that occur in the future are discounted to reflect society's rate of time preference

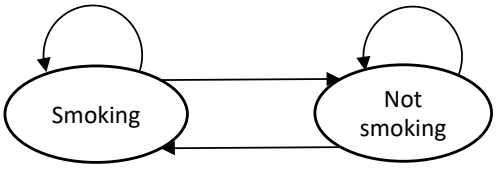
E.G. UK discount in first year is 1, second year is 1.035^{-1}, third is 1.035^{-2}, …, fifth is 1.035^{-4}

But if a cycle is 6 months…. 1 for first two cycles, 1.035^{-1} for third and fourth cycle, …, 1.035^{-4} for ninth and 10th cycle

Overview

- Markov modelling
- Smoking cessation Markov model
  - Cohort simulation
  - Costs/QALYs
- Coding the smoking cessation Markov model in R
- Practical

Smoking Cessation Example



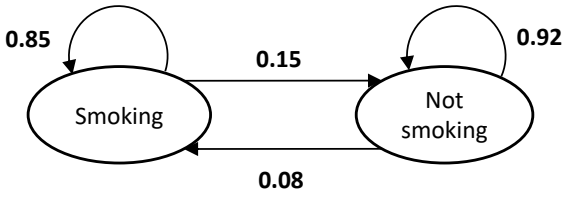
Arrows show movements that can occur in 1 time cycle

This includes staying in the same state (shown by curved arrows)

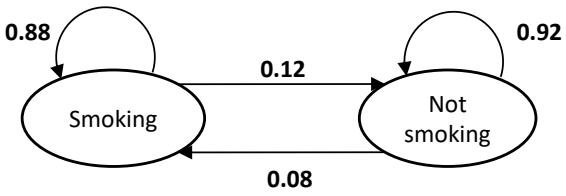
13

Smoking Cessation Example

6-month cycles, 5 year time horizon



Standard Care + Website



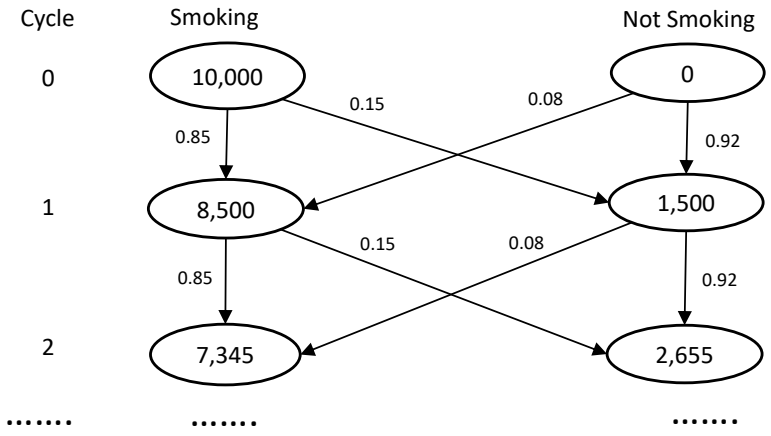
Standard Care

14

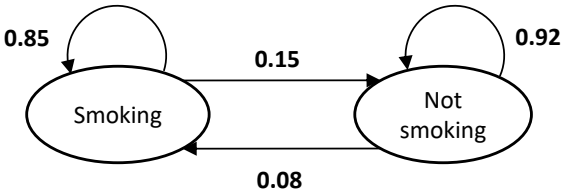
Cohort Simulation

Cohort vector π at time t (π_t) is the cohort vector at the previous time point (π_{t-1}) multiplied by the probability transition matrix P

$$\pi_t = \pi_{t-1}P$$



Smoking Cessation Example



Standard Care + Website

Updating cohort vector

Or in components

$$\left(\pi_{Smoking,t}, \pi_{Not\ smoking,t}\right) = \left(\pi_{Smoking,t-1}, \pi_{Not\ smoking,t-1}\right)\begin{pmatrix}0.85 & 0.15\\ 0.08 & 0.92\end{pmatrix}$$

Initial state ($t = 0$) may be everyone in smoking ($\pi_0 = (1,0)$) then this updates to $\pi_1 = (0.85,0.15)$ with 15% of patients quitting smoking

If initial state was $\pi_0 = (0.6, 0.4)$ this would update to

$\pi_1 = (\%\ still\ smoking\ +\ \%\ starting\ smoking,\ \%\ still\ not\ smoking + \%\ quitting)$

$$\left((0.6 * 0.85) + (0.4 * 0.08), (0.4 * 0.92) + (0.6\ * 0.15)\right) = (0.542, 0.458)$$

Updating cohort vector

Run this to time horizon to get state full occupancy

$10000 * 0.85 = 8500$

$10000 * 0.15 = 1500$

$8500 * 0.85 + 1500 * 0.08 = 7345$

$8500 * 0.15 + 1500 * 0.92 = 2655$

	Standard Care + Website		Standard Care	
Cycle number	Smoking	Not Smoking	Smoking	Not Smoking
0	10000	0	10000	0
1	8500	1500	8800	1200
2	7345	2655	7840	2160
3	6456	3544	7072	2928
4	5771	4229	6458	3542
5	5244	4756	5966	4034
6	4838	5162	5573	4427
7	4525	5475	5258	4742
8	4284	5716	5007	4993
9	4099	5901	4805	5195
10	3956	6044	4644	5356

Costs and QALYS

We know that

1-year in the smoking state = 0.95 QALYs

So 1 cycle is 0.475 QALYs

1-year in the non-smoking state = 1 QALYs

So 1 cycle is 0.5 QALYs

Website has a one-off cost of £50 per person

5 year horizon

19

Calculating costs and QALYs

If the current cohort vector is π_t

And cost and QALY per cycle spent in each state are $c_t = (0,0)$ and $q_t = (0.475, 0.5)$

Then total costs and utilities accumulated per cycle are

$$cycle\ costs = \pi_t \times c_t$$
$$cycle\ QALYs = \pi_t \times q_t$$

For time horizon T, total costs are $\sum_{t=1}^{T} \pi_t \times c_t$ and total QALYs $\sum_{t=1}^{T} \pi_t \times q_t$

20

Calculating costs and QALYs for Smoking Cessation

In the smoking cessation example $T = 10$ as 10 cycles of 6 months give 5-year time horizon

No cost per state but one-off treatment cost $d_i = (50,0)$ for standard care + website ($i = 1$) and standard of care alone ($i = 2$)

ICER = difference in costs of strategies / difference in QALY's of strategies

21

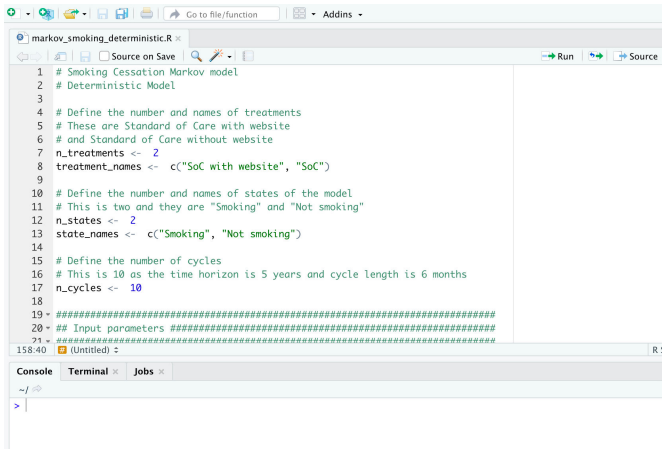
Overview

- Markov modelling
- Smoking cessation Markov model
  - Cohort simulation
  - Costs/QALYs
- Coding the smoking cessation Markov model in R
- Practical

22

Coding in R

Use Rstudio to open the file labelled “markov_smoking_deterministic.R”



Basic model specification

```
n_treatments <- 2
treatment_names <- c("SoC with website", "SoC")

n_states <- 2
state_names <- c("Smoking", "Not smoking")

n_cycles <- 10
```

An array to store transition matrices

```
transition_matrices <-  array(dim = c(n_treatments, n_states,
                                      n_states),
                              dimnames = list(treatment_names,
                                              state_names, state_names))
```

This produces an array with dimensions 2x2x2

They are currently blank so need to fill in with values…

25

Filling in the transition matrix

```
# First the transition matrix for Standard of Care with website
# Transitions from smoking
transition_matrices["SoC with website", "Smoking", ] <-  c(0.85, 0.15)

# Transitions from not smoking
transition_matrices["SoC with website", "Not smoking", ] <-  c(0.08, 0.92)

# Second the transition matrix for Standard of Care
# Transitions from smoking
transition_matrices["SoC", "Smoking", ] <-  c(0.88, 0.12)

# Transitions from not smoking
# These should be the same as the transition probabilities from not smoking for
# SoC with website as the website has no impact on probability of relapse
transition_matrices["SoC", "Not smoking", ] <-
  transition_matrices["SoC with website", "Not smoking", ]
```

26

Contents of array

Run the previous code ensuring you have filled in the *transition_matrices* array

Now look at elements of the array

```
> transition_matrices
, , Smoking

                 Smoking Not smoking
SoC with website    0.85        0.08
SoC                 0.88        0.08

, , Not smoking

                 Smoking Not smoking
SoC with website    0.15        0.92
SoC                 0.12        0.92
```

27

State QALYs

```
# Now define the QALYS associated with the states per cycle
# There is one for each state
# Store in an NA array and then fill in below
state_qalys <-  array(dim = c(n_states), dimnames = list(state_names))

# QALY associated with 1 - year in the smoking state is 0.95
# Divide by 2 as cycle length is 6 months
state_qalys["Smoking"] <-  0.95 / 2

# QALY associated with 1 - year in the not smoking state is 1
# Again divide by 2 as cycle length is 6 months
state_qalys["Not smoking"] <-  1.0 / 2
```

28

State costs

```
# And finally define the state costs
# These are all zero as the only cost is a one - off subscription fee of £50
# to the smoking cessation website
state_costs <-  array(0, dim = c(n_states), dimnames = list(state_names))
```

Can again inspect elements to make sure it's working as expected…

29

Treatment costs

```
# Define the treatment costs
# One for each treatment
# Treatment costs are actually fixed but this allows flexibility if we
# want to include uncertainty / randomness in the cost
treatment_costs <-  array(dim = c(n_treatments), dimnames = list(treatment_names))

# Cost of the smoking cessation website is a one - off subscription fee of £50
treatment_costs["SoC with website"] <-  50
# Zero cost for standard of care
treatment_costs["SoC"] <-  0
```

30

Initialise the cohort vector

```
# Build an array to store the cohort vector at each cycle
# Each cohort vector has 2 ( = n_states) elements: probability of being in smoking state,
# and probability of being in the not smoking state
# There is one cohort vector for each treatment,  for each cycle_
cohort_vectors <-  array(dim = c(n_treatments, n_cycles, n_states),
                         dimnames = list(treatment_names, NULL, state_names))

# Assume that everyone starts in the smoking state no matter the treatment
cohort_vectors[, 1, "Smoking"] <-  1
cohort_vectors[, 1, "Not smoking"] <-  0
```

These are the two-dimensional π_t in the Markov formula

Core loop

Loop over treatments

{

Loop over cycles

{

Update cohort vector

$$\pi_t = \pi_{t-1}P$$

or specifically…

$$(\pi_{Smoking,t}, \pi_{Not\ smoking,t}) = (\pi_{Smoking,t-1}, \pi_{Not\ smoking,t-1})P$$

}

1. Calculate cycle costs and QALYs
2. Calculate total costs and QALYs

}

Core loop

```
# Main model code
# Loop over the treatment options
for(i_treatment in 1:n_treatments)
{
  # Loop over the cycles
  # Cycle 1 is already defined so only need to update cycles 2:n_cycles
  for(i_cycle in 2:n_cycles)
  {
    # Markov update
    # Multiply previous cycle's cohort vector by transition matrix
    # i_e_ pi_j  =  pi_(j - 1) * P
    cohort_vectors[i_treatment, i_cycle, ] <-
      cohort_vectors[i_treatment, i_cycle - 1, ] %*%
      transition_matrices[i_treatment, , ]

    # 1. Calculate cycle costs and QALYs
    # 2. Calculate total costs and QALYs

  }
}
```

33

Cycle costs and cycle QALYs

```
# Build an array to store the costs and QALYs accrued per cycle
# One for each treatment, for each cycle
# These will be filled in below in the main model code
# Then discounted and summed to contribute to total costs and total QALYs
cycle_costs <-  array(dim = c(n_treatments, n_cycles),
                      dimnames = list(treatment_names, NULL))
cycle_qalys <-  array(dim = c(n_treatments, n_cycles),
                      dimnames = list(treatment_names, NULL))
```

Not strictly necessary to store these but might be interested in costs or QALYs accrued per cycle.

34

Arrays to store total cost and QALYs

```
# Build arrays to store the total costs and total QALYs
# There is one for each treatment
# These are filled in below using cycle_costs,  treatment_costs,  and cycle_qalys
total_costs <-  array(dim = c(n_treatments),
                      dimnames = list(treatment_names))
total_qalys <-  array(dim = c(n_treatments),
                      dimnames = list(treatment_names))
```

Once filled in by Markov loop, these are used to calculate net benefit and ICERs

35

Discount rate

The powers repeat so in R could write

```
> c(0,0,1,1,2,2,3,3,4,4)
 [1] 0 0 1 1 2 2 3 3 4 4
```

Or use the rep() function

```
> rep(c(0:4), each=2)
 [1] 0 0 1 1 2 2 3 3 4 4
```

Or (preferred) make it general to any number of cycles in our Markov model. Note that formula below only works for an even number of cycles:

```
> rep(c(0:(n_cycles / 2-1)), each = 2)
 [1] 0 0 1 1 2 2 3 3 4 4
```

36

Discount rate

The discount factor for costs and QALYs

```
(1 / 1.035)^rep(c(0:(n_cycles / 2-1)), each = 2)
```

```
[1] 1.0000000 1.0000000 0.9661836 0.9661836 0.9335107
0.9335107 0.9019427 0.9019427 0.8714422 0.8714422
```


Calculating cycle costs and QALYs

For each treatment we use the cohort_vectors[] to calculate costs and QALYs associated with each cycle

```
# Now use the cohort vectors to calculate the
# Total costs for each cycle
cycle_costs[i_treatment, ] <-  cohort_vectors[i_treatment, , ] %*% state_costs[]
# And total QALYs for each cycle
cycle_qalys[i_treatment, ] <-  cohort_vectors[i_treatment, , ] %*% state_qalys[]
```

Calculating total costs and QALYs

```
# Combine the cycle_costs and treatment_costs to get total costs
# Apply the discount factor
total_costs[i_treatment] <-  treatment_costs[i_treatment] +
  cycle_costs[i_treatment, ] %*%
  (1 / 1.035)^rep(c(0:(n_cycles / 2 - 1)), each = 2)

# Combine the cycle_qalys to get total qalys
# Apply the discount factor
total_qalys[i_treatment] <-  cycle_qalys[i_treatment, ] %*%
  (1 / 1.035)^rep(c(0:(n_cycles / 2 - 1)), each = 2)
```

Note treatment costs are added (and not discounted as only occur in first year)

39

Results

```
# Incremental costs and effects relative to standard of care
# No uncertainty in the costs as the website cost is fixed at £50
incremental_costs <-  total_costs["SoC with website"] - total_costs["SoC"]
incremental_effects <-  total_qalys["SoC with website"] - total_qalys["SoC"]


# The ICER comparing Standard of care with website to standard of care
# This is much lower than the £20, 000 willingness - to - pay threshold indicating
# good value for money
ICER <-  incremental_costs / incremental_effects
```

40

Overview

- Markov modelling
- Smoking cessation Markov model
  - Cohort simulation
  - Costs/QALYs
- Coding the smoking cessation Markov model in R
- Practical

41

Summary

We have explained the 2-state and 2-treatment option smoking cessation Markov model

We have explained the key steps in building a discrete time deterministic Markov model in R

- Define input parameters
- Update cohort vector and calculate total costs and QALYs

The code we provided is general

- Included state costs even though these are zero in smoking cessation
- Change numbers of states and input parameters to adapt

42

Overview

- Markov modelling
- Smoking cessation Markov model
  - Cohort simulation
  - Costs/QALYs
- Coding the smoking cessation Markov model in R
- Practical

43

Practical

So far, we have assumed that there are no costs associated with the health states. Add in a fixed cost of £100 per year associated with the smoking state by:

Updating the state costs array to replace the zeroes with a £100 per year state cost for the smoking state. Specify that the no smoking state is associated with zero costs.

Comment on how this changes the ICER

44

University of BRISTOL

Session 3 - Probabilistic Markov models

Howard Thom

University of BRISTOL

Background

- You've seen how to build deterministic Markov models
- We often don't know transition probabilities, utilities, or costs exactly.
- Best we can do is represent uncertainty around these parameters with probability distributions
- Simulating this uncertainty in economic evaluation models is called **probabilistic analysis**

Not probabilistic sensitivity analysis as the base case itself is uncertain
This is our recommended base case

University of BRISTOL

Outline

- We will adapt the processes and code from the previous session to do the following in probabilistic analysis
- Generating transition matrices
- Generating costs and QALYs
- Markov cohort simulation
- Analysing results

University of BRISTOL

Making transition matrices probabilistic

University of BRISTOL

Probabilistic analysis – transition matrices

- Transition matrix for SoC + website was previously assumed known exactly as

$$\begin{pmatrix} 0.85 & 0.15 \\ 0.08 & 0.92 \end{pmatrix}$$

- In reality, we might estimate this from study data.
- For example, a study of two cohorts of 100 patients followed over 6 months starting in smoking and non-smoking states and receiving standard of care + website.

SoC + website	Smoking at 6 months	Not smoking at 6 months
Smoking at baseline	85	15
Not smoking at baseline	8	92

University of BRISTOL

Probabilistic analysis – transition matrices

- Binary outcomes data is conveniently represented by a Beta distribution

SoC + website	Smoking at 6 months	Not smoking at 6 months	Distribution
Smoking at baseline	85	15	$Beta\,(85, 15)$
Not smoking at baseline	8	92	$Beta\,(8, 92)$

- If we had more states (e.g. smoking, reduced smoking, no smoking) could use a Dirichlet distribution to represent more than 2 uncertain transition probabilities

University of BRISTOL

Probabilistic analysis – transition matrices

- Each row of the transition matrix for SoC + website is therefore represented by a beta distribution

$$\begin{pmatrix} beta\ (85, 15) \\ beta\ (8, 92) \end{pmatrix}$$

- Similarly, the SoC transition matrix is represented by

$$\begin{pmatrix} beta(88, 12) \\ beta(8, 92) \end{pmatrix}$$

SoC alone	Smoking at 6 months	Not smoking at 6 months	Distribution
Smoking at baseline	88	12	$Beta\ (88, 12)$
Not smoking at baseline	8	92	$Beta\ (8, 92)$

University of BRISTOL

Probabilistic analysis – beta distribution in R

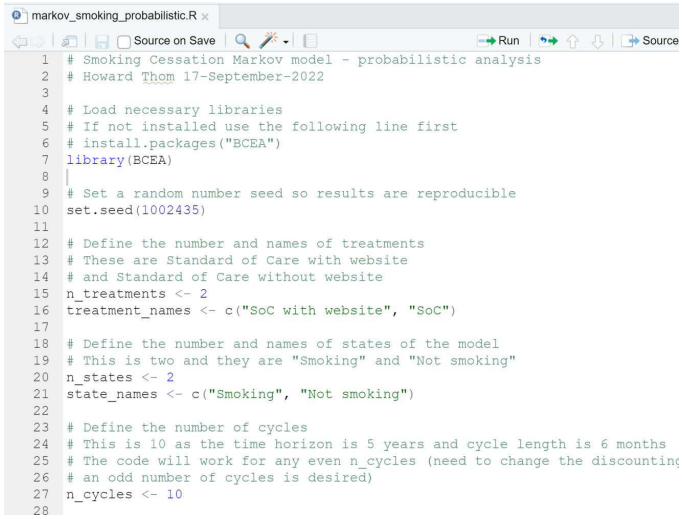
- The `rbeta()` function takes a number of samples 'n' and its α and β parameters

SoC + website	Smoking at 6 months	Not smoking at 6 months	Distribution
Smoking at baseline	85	15	$Beta\ (85, 15)$
Not smoking at baseline	8	92	$Beta\ (8, 92)$

```
> rbeta(n = 10, 85, 15)
 [1] 0.7970600 0.8053360 0.8801466 0.9074958 0.8868830 0.7625788 0.8323798
 [8] 0.8143802 0.8818394 0.8143785
> rbeta(n = 10, 8, 92)
 [1] 0.05580082 0.08638050 0.07016425 0.05184869 0.10193435 0.04942523 0.08096863
 [8] 0.08395457 0.06294023 0.09924210
```

Open the file

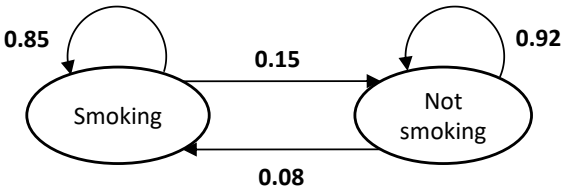
- If you haven't already, use R or Rstudio to open the file labelled "markov_smoking_probabilistic.R"
- Note the `set.seed()`
- This ensures results are same each time the model is run, making the analysis reproducible



Basic model specification

```
# Define the number and names of treatments
# These are Standard of Care with website
# and Standard of Care without website
n_treatments <- 2
treatment_names <- c("SoC with website", "SoC")

# Define the number and names of states of the model
# This is two and they are "Smoking" and "Not smoking"
n_states <- 2
state_names <- c("Smoking", "Not smoking")

# Define the number of cycles
# This is 10 as the time horizon is 5 years and cycle length is 6 months
# The code will work for any even n.cycles (need to change the discounting code if
# an odd number of cycles is desired)
n_cycles <- 10

# Define simulation parameters
# This is the number of samples to use
n_samples <- 1000
```

An array to store transition matrices




```
# The transition matrix is a 2x2 matrix
# Rows sum to 1
# Top left entry is transition probability from smoking to smoking
# Top right is transition probability from smoking to not smoking
# Bottom left is transition probability from not smoking to smoking
# Bottom right is transition probability from not smoking to not smoking

# There is one transition matrix for each treatment option and each sample
# Store them in an array with (before filling in below) NA entries
transition_matrices <- array(dim = c(n_treatments, n_samples, n_states, n_states),
        dimnames = list(treatment_names, NULL, state_names, state_names))
```

- This produces an array with dimensions 2x1000x2x2
- They are currently blank so need to fill in with values…

Filling in the transition matrix

```
# First the transition matrix for Standard of Care with website
# Transitions from smoking
temp <- rbeta(n_samples, 85, 15)
transition_matrices["SoC with website", , "Smoking", ] <-
                              matrix(c(temp, 1 - temp), ncol = 2)
# Transitions from not smoking
temp <- rbeta(n_samples, 8, 92)
transition_matrices["SoC with website", , "Not smoking", ] <-
                              matrix(c(temp, 1 - temp), ncol = 2)

# Second the transition matrix for Standard of Care
# Transitions from smoking
temp <- rbeta(n_samples, 88, 12)
transition_matrices["SoC", , "Smoking", ] <- matrix(c(temp, 1-temp), ncol = 2)
# Transitions from not smoking
# These should be the same as the transition probabilities from not smoking for SoC with website
# as the website has no impact on probability of relapse
transition_matrices["SoC", , "Not smoking", ] <-
                              transition_matrices["SoC with website", , "Not smoking", ]
```

Contents of array?

- Run the code up to line 64, ensuring you have filled in the transition matrices array
- Look at elements of the array
- For example, first sampled transition matrix for standard of care:

```
> transition_matrices["SoC", 1, ,]
              Smoking Not smoking
Smoking     0.8568441   0.1431559
Not smoking 0.1053515   0.8946485
```


Contents of array?

- Or the 10th sample for standard of care with website
- Or first 10 samples of transition probabilities from 'Smoking' on standard of care with website

```
> transition_matrices["SoC", 10, ,]
              Smoking Not smoking
Smoking     0.87416988   0.1258301
Not smoking 0.07119714   0.9288029
> transition_matrices["SoC", 1:10, "Smoking" ,]
         Smoking Not smoking
 [1,] 0.8568441   0.1431559
 [2,] 0.8696609   0.1303391
 [3,] 0.8373699   0.1626301
 [4,] 0.8784279   0.1215721
 [5,] 0.8927902   0.1072098
 [6,] 0.8735486   0.1264514
 [7,] 0.8843967   0.1156033
 [8,] 0.8617107   0.1382893
 [9,] 0.8109626   0.1890374
[10,] 0.8741699   0.1258301
```

University of BRISTOL

Exercise 1

a) Run the code as far as line 64 to fill in the transition matrices array.

b) One sample of the transition probabilities from Smoking on SoC with website are given by calling

```
transition_matrices["SoC with website", 1, "Smoking", ]
```

Use the `colMeans()` function to compare the average over all samples of the transition probabilities from Smoking to Smoking and Not smoking on SoC with website and SoC.

c) What about the transition probabilities from not smoking? Do they differ between SoC with website and SoC alone?

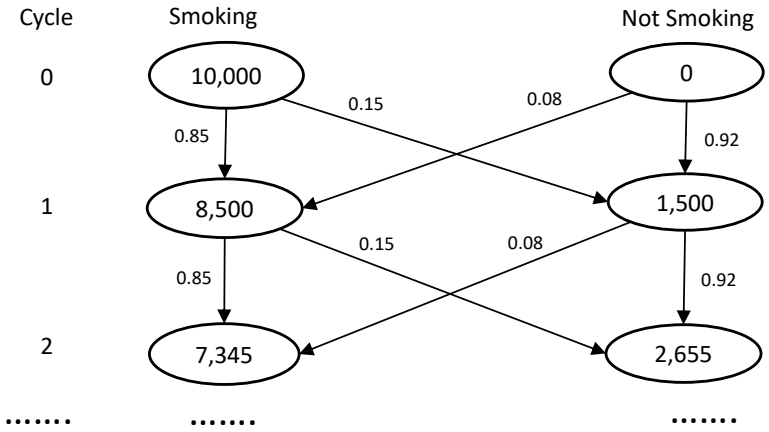
University of BRISTOL

Markov simulation in probabilistic analysis

Reminder: Cohort Simulation

- Cohort vector π at time t (π_t) is the cohort vector at the previous time point (π_{t-1}) multiplied by the probability transition matrix P

$$\pi_t = \pi_{t-1}P$$



Initialise the cohort vector

```
# Build an array to store the cohort vector at each cycle
# Each cohort vector has 2 (=n_states) elements: probability of being in smoking state,
# and probability of being in the not smoking state
# There is one cohort vector for each treatment, for each sample, for each cycle.
cohort_vectors <- array(dim = c(n_treatments, n_samples, n_cycles,
n_states), dimnames = list(treatment_names, NULL, NULL, state_names))

# Assume that everyone starts in the smoking state no matter the treatment
cohort_vectors[, , 1, "Smoking"] <- 1
cohort_vectors[, , 1, "Not smoking"] <- 0
```

- These are the two-dimensional π_t in the Markov formula

Core loop

Loop over treatments
{
Loop over samples
{
Loop over cycles
{
Update cohort vector

$$\pi_t = \pi_{t-1}P$$

or specifically…

$$(\pi_{Smoking,t}, \pi_{Not\ smoking,t}) = (\pi_{Smoking,t-1}, \pi_{Not\ smoking,t-1})P$$

}
1. Calculate cycle costs and QALYs for this sample
2. Calculate total costs and QALYs for this sample
}
}

Core loop

```
# Loop over the treatment options
for (i_treatment in 1:n_treatments)
{
        # Loop over the samples
        for (i_sample in 1:n_samples)
        {
                # Loop over the cycles
                # Cycle 1 is already defined so only need to update cycles 2:n_cycles
                for (i_cycle in 2:n_cycles)
                {
                        # Multiply previous cycle's cohort vector by transition matrix
                                cohort_vectors[i_treatment, i_sample, i_cycle, ] <-
                                cohort_vectors[i_treatment, i_sample, i_cycle-1, ] %*%
                                transition_matrices[i_treatment, i_sample, , ]
                }               }
                1. Calculate cycle costs and QALYs for this sample
                2. Calculate total costs and QALYs for this sample
        }
}
```

University of BRISTOL

Core loop

```
# Loop over the treatment options
for (i_treatment in 1:n_treatments)
{
        # Loop over the samples
        for (i_sample in 1:n_samples)
        {
                # Loop over the cycles
                # Cycle 1 is already defined so only need to update cycles 2:n_cycles
                for (i_cycle in 2:n_cycles)
                {
                        # Multiply previous cycle's cohort vector by transition matrix
                                cohort_vectors[i_treatment, i_sample, i_cycle, ] <-
                                cohort_vectors[i_treatment, i_sample, i_cycle-1, ] %*%
                                transition_matrices[i_treatment, i_sample, , ]
                }               }
                1. Calculate cycle costs and QALYs for this sample
                2. Calculate total costs and QALYs for this sample
        }
}
```

This will be implemented next

University of BRISTOL

Making costs and QALYs probabilistic

University of BRISTOL

Probabilistic sensitivity analysis – costs and QALYs

- The QALYs associated with 1 year in the smoking state are now normally distributed as Normal(mean = 0.95, sd = 0.01)
  - We divide the above by 2 to get QALYs for one 6-month cycle in the smoking state.
- The QALYs associated with 1 year in the non-smoking state remain fixed at 1.00 (perfect health)
- Cost of website also remains fixed as £50

University of BRISTOL

State QALYs

```
# Now define the QALYS associated with the states per cycle
# There is one for each sample and each state
# Store in an NA array and then fill in below
state_qalys <- array(dim = c(n_samples, n_states), dimnames = list(NULL,
state_names))

# QALY associated with 1-year in the smoking state is Normal(mean=0.95, SD=0.01)
# Divide by 2 as cycle length is 6 months
state_qalys[, "Smoking"] <- rnorm(n_samples, mean=0.95, sd=0.01) / 2

# QALY associated with 1-year in the not smoking state is 1 (no uncertainty)
# So all samples have the same value
# Again divide by 2 as cycle length is 6 months
state_qalys[, "Not smoking"] <- 1 / 2
```

State costs

```
# And finally define the state costs
# These are all zero as the only cost is a one-off subscription fee of £50
# to the smoking cessation website
state_costs <- array(0, dim = c(n_samples, n_states), dimnames = list(NULL,
state_names))
```

- Can again inspect elements to make sure it's working as expected…

Checking state costs and QALYs

```
> state_qalys[1:5, "Smoking"]
[1] 0.4728097 0.4751482 0.4762728 0.4768340 0.4711733
> state_qalys[1:5, "Not smoking"]
[1] 0.5 0.5 0.5 0.5 0.5
> state_costs[1:5, "Smoking"]
[1] 0 0 0 0 0
> state_costs[1:5, "Not smoking"]
[1] 0 0 0 0 0
```

- State QALYs in smoking state are uncertain but centred around 0.475
- QALYs in not smoking state are 0.5 (6 month cycle)
- And state costs are always zero

Treatment costs

```
# Define the treatment costs
# One for each  sample and each treatment
# Treatment costs are actually fixed but this allows flexibility if we
# want to include uncertainty/randomness in the cost
treatment_costs <- array(dim = c(n_treatments, n_samples), dimnames = list(treatment_names,
NULL))

# Cost of the smoking cessation website is a one-off subscription fee of £50
treatment_costs["SoC with website", ] <- 50
# Zero cost for standard of care
treatment_costs["SoC", ] <- 0
```


Cycle costs and cycle QALYs

```
# Build an array to store the costs and QALYs accrued per cycle
# One for each treatment, for each  sample, for each cycle
# These will be filled in below in the main model code
# Then discounted and summed to contribute to total costs and total QALYs
cycle_costs <- array(dim = c(n_treatments, n_samples, n_cycles),
       dimnames = list(treatment_names, NULL, NULL))
cycle_qalys <- array(dim = c(n_treatments, n_samples, n_cycles),
       dimnames = list(treatment_names,NULL, NULL))
```

- Not strictly necessary to store these but might be interested in costs or QALYs accrued per cycle.

Arrays to store total cost and QALYs

```
# Build arrays to store the total costs and total QALYs
# There is one for each treatment and each  sample
# These are filled in below using cycle_costs,
# treatment_costs, and cycle_qalys
total_costs <- array(dim = c(n_treatments, n_samples),
      dimnames = list(treatment_names, NULL))
total_qalys <- array(dim = c(n_treatments, n_samples),
      dimnames = list(treatment_names,NULL))
```

- Once filled in by Markov loop, these are used to calculate net benefit and ICERs

Calculating cycle costs and QALYs

- For each treatment and each sample, we use the cohort_vectors[] to calculate costs and QALYs associated with each cycle

```
# Now use the cohort vectors to calculate the total costs for each cycle
cycle_costs[i_treatment, i_sample, ] <- cohort_vectors[i_treatment, i_sample, , ]
%*% state_costs[i_sample, ]

# And total QALYs for each cycle
cycle_qalys[i_treatment, i_sample, ] <- cohort_vectors[i_treatment, i_sample, , ]
%*% state_qalys[i_sample, ]
```

Core loop

```
# Loop over the treatment options
for (i_treatment in 1:n_treatments)
{
        # Loop over the  samples
        for (i_sample in 1:n_samples)
        {
                # Loop over the cycles
                # Cycle 1 is already defined so only need to update cycles 2:n_cycles
                for (i_cycle in 2:n_cycles)
                {
                        # Multiply previous cycle's cohort vector by transition matrix
                        # i_e_ pi_j = pi_(j-1)*P
                        cohort_vectors[i_treatment, i_sample, i_cycle, ] <-
                                cohort_vectors[i_treatment, i_sample, i_cycle-1, ] %*%
                                transition_matrices[i_treatment, i_sample, , ]
                }
# Now use the cohort vectors to calculate the total costs for each cycle
cycle_costs[i_treatment, i_sample, ] <- cohort_vectors[i_treatment, i_sample, , ] %*% state_costs[i_sample, ]
# And total QALYs for each cycle
cycle_qalys[i_treatment, i_sample, ] <- cohort_vectors[i_treatment, i_sample, , ] %*% state_qalys[i_sample, ]

        2. Calculate total costs and QALYs for this  sample
        }
}
```

Implement this final step in R

Calculating total costs and QALYs

```
# Combine the cycle_costs and treatment_costs to get total costs
# Apply the discount factor
# (1 in first year, 1_035 in second, 1_035^2 in third, and so on)
# Each year acounts for two cycles so need to repeat the discount values
total_costs[i_treatment, i_sample] <- treatment_costs[i_treatment, i_sample]
+ cycle_costs[i_treatment, i_sample, ] %*% (1 / 1.035)^rep(c(0:(n_cycles / 2-
1)), each = 2 )

# Combine the cycle_qalys to get total qalys
# Apply the discount factor
# (1 in first year, 1_035 in second, 1_035^2 in third, and so on)
# Each year accounts for two cycles so need to repeat the discount values
total_qalys[i_treatment, i_sample] <- cycle_qalys[i_treatment, i_sample, ]
%*% (1 / 1.035)^rep(c(0:(n_cycles / 2-1)), each = 2)
```

- Note treatment costs are added (and not discounted as only occur in first year)

University of BRISTOL

Analysing results in probabilistic analysis

University of BRISTOL

Mean costs and effects

```
# Average costs
# These are £50 on the website and 0 on standard of care as there are no costs other than the website subscription cost
average_costs <- rowMeans(total_costs)
# Average effects (in QALY units)
# These are slightly higher on the website as higher probability of quitting smoking
average_effects <- rowMeans(total_qalys)
```

```
> average_costs
SoC with website              SoC
              50                0
> average_effects
SoC with website              SoC
        4.527508         4.514881
```

- So we see that costs are higher on website (knew that!) but that QALYs are also higher

Incremental Cost-Effectiveness Ratio

```
# Incremental costs and effects relative to standard of care
# No uncertainty in the costs as the website cost is fixed at £50
incremental_costs <- total_costs["SoC with website", ] - total_costs["SoC", ]

# In some samples the website leads to higher QALYs but in others it is negative
# There is uncertainty as to whether the website is an improvement over SoC
incremental_effects <- total_qalys["SoC with website", ] - total_qalys["SoC", ]

# The ICER comparing Standard of care with website to standard of care
# This is much lower than the £20,000 willingness-to-pay threshold indicating good value for money
ICER <- mean(incremental_costs) / mean(incremental_effects)
```

```
> ICER
[1] 3959.624
```

- Website likely cost-effective

Incremental net benefit

```
incremental_net_benefit <- 20000*incremental_effects - incremental_costs
```

```
> incremental_net_benefit[1:25]
 [1]  -59.18312 -339.24847 -661.95402  -92.19771  160.27551
 [6]   25.20792 -205.42779  185.04276 -568.57411   58.14497
[11]  -52.74228  143.52906   74.98032  618.18642   77.11779
[16]  -49.11706  276.05853  400.06418  507.20305  543.87674
[21]  588.72652 -197.52688  136.89929  407.32936  793.21667
```

- This is sometimes positive and sometimes negative
- Need to look at the average to get a clearer picture

```
> average_inb <- mean(incremental_net_benefit)
> average_inb
[1] 202.5492
```

- Positive so expected net benefit higher on website than on standard of care

Probability cost-effective

```
> probability_cost_effective <- sum(incremental_net_benefit > 0) / n_samples
> probability_cost_effective
[1] 0.688
```

- This is the proportion of samples for which the incremental net benefit is positive
- It is close to 70%, representing good degree of certainty in recommendation to adopt the smoking cessation website

Analysing the results using BCEA

- BCEA (Bayesian Cost Effectiveness Analysis) is a package to analyse the results (simulated total costs and total QALYs) and produce standard output such as ICERs, CEACs and EVPI.

Note: In this example can't use total_costs and total_qalys directly in BCEA as they are n_treatments by n_samples rather than n_samples by n_treatments. Use the t() function to transpose the total_costs and total_qalys matrices when inputting them to BCEA.

```
Smoking_bcea <- bcea(e = t(total_qalys), c = t(total_costs), ref = 1,
interventions = treatment_names)

summary(smoking_bcea, wtp = 20000)
```

University of BRISTOL

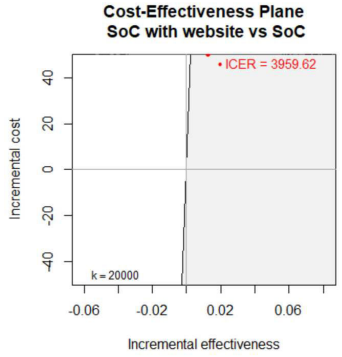
BCEA output

```
Cost-effectiveness analysis summary

Reference intervention:  SoC with website
Comparator intervention: SoC

Optimal decision: choose SoC for k <  and  for k >=


Analysis for willingness to pay parameter k = 20000

                 Expected net benefit
SoC with website                90500
SoC                             90298

                            EIB  CEAC   ICER
SoC with website vs SoC 202.55 0.688 3959.6

Optimal intervention (max expected net benefit) for k = 20000: SoC with website

EVPI 81.375
```


University of BRISTOL

Cost-effectiveness plane

```
ceplane.plot(smoking_bcea, wtp = 20000)
```

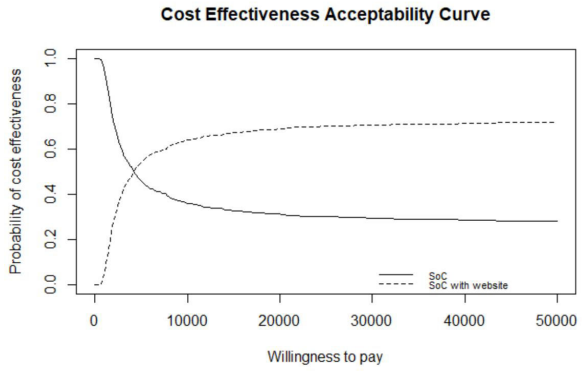


- Cost differential is always £50 as cost of website is fixed
- Variation in effectiveness over simulated sample plotted

University of BRISTOL

Cost Effectiveness Acceptability Curve

```
smoking_multi_ce <- multi.ce(smoking_bcea)
ceac.plot(smoking_multi_ce)
```



- SoC is optimal up to £3700 willingness-to-pay per QALY
- Above £4k SoC with website is optimal

University of BRISTOL

Summary

- We have explained key steps in building both deterministic and probabilistic discrete-time cohort Markov models
- To make a model probabilistic we need to
  - Sample probabilistic transition matrices
  - Simulate the Markov model using these probabilistic matrices
  - Sample probabilistic state costs and QALYs
  - Analyse the results accounting for uncertainty in costs, QALYs, and net benefits
- The code we provided is general
  - For example, included state costs even though these are zero in smoking cessation

University of BRISTOL

Exercise 2 – Adding a death state

In reality, models will have more than two states.

Go through the code and add in an extra state to represent death by:

a) Change number of states from 2 to 3 and naming the death state
b) Assume that there are 2 deaths in every 100 patients in the smoking state and 1 death in the non-smoking state, each represented by a beta distribution. **(See next slide for hint on implementation using two beta distributions)**
c) Define transitions from death so that it is an absorbing state that people cannot move back from.
d) Check that you have set up your transition matrix correctly using the code transition_matrices["SoC with website", 1, ,]
e) Assign a QALY of 0 and a cost of 0 to the death state
f) Rerun the simulation including the death state, assuming that no one starts in the death state
g) Analyse the results using BCEA. What impact does adding the death state have on the results?

University of BRISTOL

Exercise 2 – Transitions using two beta distributions

```
# Assume that people have a 2/100 probability of dying in the smoking state
# and a 1/100 probability of dying in the non-smoking state.
probability_of_death_smoking <- rbeta(n_samples, 2, 98)
probability_of_death_not_smoking <- rbeta(n_samples, 1, 99)

# Transitions from smoking
temp <- rbeta(n_samples, 85, 15)
transition_matrices["SoC with website", , "Smoking", ] <-
matrix(c((1 - probability_of_death_smoking) * c(temp, 1 - temp),
probability_of_death_smoking), ncol = 3)
```

University of BRISTOL

Exercise 2 – Solution

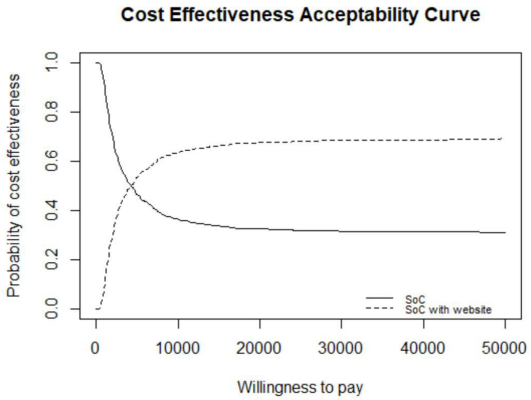
Using summary() of the new bcea object should give

	EIB	CEAC	ICER
SoC with website vs SoC	238.27	0.676	3469

The ICER is marginally reduced (from 3969) because the difference in effects is increased.

Uncertainty is marginally increased as CEAC has gone from 0.688 to 0.676, so closed to 0.50.

Code in session_3_exercise_2_solution.R



bristol.ac.uk

University of BRISTOL

Thank you!

Dr Howard Thom

Senior Lecturer, University of Bristol

Howard.thom@bristol.ac.uk

Additional useful packages for R modelling

ISPOR Europe 2023 - In-person short course

Rose Hart

Lumanity Inc., Sheffield, UK

12 November 2023

Lumanity

An overview of additional useful packages for health economic modelling

`hesim` - an R package for implementing and analysing health economic simulation models

`rmarkdown` - a file format for making documents with R

`shiny` - a package that makes it possible to create interactive web applications from R code

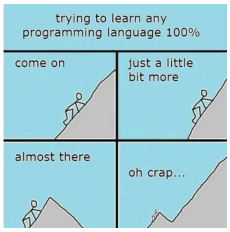
Session objectives

By the end of this session you should:

1) Be aware of additional useful packages in R that can be useful for building models, documenting them, and distributing them to a technical or non-technical audience

2) Know how to set up a default shiny app and rmarkdown script, and know locations of examples for further learning

3) Have considered learning pathways and objectives specific to your work requirements



Group discussion

In groups of two or three (those seated around you), please discuss the following questions and come up with at least 1 point for each:

1) What do you most want to learn how to do in R?

2) What are the key barriers to you developing and using R for what you need?

You have 5 minutes

The hesim package

`hesim` supports three types of health economic models:

(i) **Cohort discrete time state transition models (cDTSTMs)** - These are Markov cohort models and can be time-homogeneous or time-inhomogeneous

(ii) **N-state partitioned survival models (PSMs)** - Area under the curve model

(iii) **Individual-level continuous time state transition models (iCTSTMs)** - individual-level simulations that can encompass both Markov and semi-Markov processes

All models are implemented as R6 classes and have methods for simulating disease progression, QALYs, and costs.

This package is well documented in its CRAN vignette and publication

CRAN vignette: https://cran.r-project.org/web/packages/hesim/vignettes/intro.html
Publication: https://www.researchgate.net/publication/349424271

One of the examples is recreated in the short course GitHub repo (*hesim example model.R* script in the *Additional useful packages* folder)

hesim functionalities

Economic model (R6 class)	Statistical model	Parameter object	Model object
`hesim::CohortDtstm`	Custom	`hesim::tparams_transprobs`	`msm::msm`
	Multinomial logistic regressions	`hesim::params_mlogit_list`	`hesim::multinom_list`
`hesim::Psm`	Independent survival models	`hesim::params_surv_list`	`hesim::flexsurvreg_list`
`hesim::IndivCtstm`	Multi-state model (joint likelihood)	`hesim::params_surv`	`flexsurv::flexsurvreg`
	Multi-state model (transition-specific)	`hesim::params_surv_list`	`hesim::flexsurvreg_list`

You've made a model. What's next?

You will now need to communicate it to a wider audience.

Therefore, you will need to consider the following points for your project:

- Who is your audience?
  - Do they know R?
- What outputs do you need to effectively communicate this model, and make it as transparent as possible?
  - Graphs
  - Tables
  - Intermediate calculations
- What documentation is required?

R Markdown

`rmarkdown` - a file format for making documents with R

There are two examples of this in short course GitHub repo, one in html and one in pdf.

Please open the *Additional useful packages* folder and open the *R Markdown scripts* folder. There are two scripts there.

- Looking through, there are differences between a standard R script and an rmd script
  - It is text (white background) interspaced by code 'chunks' (grey background)
  - This means that you do not need to copy and paste R outputs to your report, you only need to write the report and interpret
- There is really good documentation on how you can create one
  - https://bookdown.org/yihui/rmarkdown - Online book
  - https://rmarkdown.rstudio.com/gallery.html - Examples

A default R Markdown document can be created from within RStudio (5 minute example exercise)

R Markdown

The examples in this project are called from the *hesim example model.R* script

```
library(rmarkdown)    # For creating markdown outputs (html and pdf)
library(bookdown)     # For creating markdown outputs (html and pdf)
library(knitr)        # For creating markdown outputs (html and pdf)
library(kableExtra)   # For creating nice-looking tables in rmarkdown

Export_params <- list(
  # Main results
  Stateprobs            = ictstm$stateprobs_,
  Summarisedf           = ce_sim_ictstm,
  labs_indiv            = labs_indiv
)

Markdown_location <- "./Additional useful packages/R Markdown scripts/"
```


R Markdown

```
# html document
rmarkdown::render(
  input = file.path(Markdown_location,"hesim html report.Rmd"),
  output_format = 'bookdown::html_document2',
  output_file = "./Additional useful packages/hesim-html-report.html",
  params = Export_params,
  envir = environment()
)

# pdf document
rmarkdown::render(
  input = file.path(Markdown_location,"hesim pdf report.Rmd"),
  output_format = 'bookdown::pdf_document2',
  output_file = "./Additional useful packages/hesim-pdf-report.pdf",
  params = Export_params,
  envir = environment()
)
```

If you are unable to generate these documents at present, the generated html and pdf documents are in the *Additional useful packages* folder at the top level.

Shiny

`shiny` - a package that makes it possible to create interactive web applications from R code

Creating an R shiny app considerably increases the accessibility of your R code. However, it is a further learning curve and can be time consuming.

At it's most basic, a shiny app usually consists of three scripts:

- app.R
  - The application is called from this script
- server.R
  - The app functionality. This is wrapped within a function: `function(input, output, session){}`
  - This contains the code for the app 'back-end'
- ui.R
  - This is the layout of the graphical user interface (GUI)
  - This contains the code for the app 'front-end'

However, it is possible to have the server function and ui within the app.R script; this is the case in the default app if created from RStudio (5 minute example exercise)

Shiny reactivity

Having your R code as a shiny app enables users to interact with the R code without seeing the R code. This works by the app 'front-end' changes causing reactions in the 'back-end' calculations.

1. The user interacts with the input boxes defined in the `ui.R` script
2. The functions which are reactive to front-end changes are wrapped in reactive functions (e.g. reactive(), observe()) in `server.R`
3. The reactive functions are always 'listening' for changes - when they detect a change the function will re-run
4. The re-running of the reactive functions causes a change in the output, which the user can then see

An object that is reactive (created using reactive()) is a function. This means that when referring to it later in the script, the syntax changes and brackets are needed (e.g. ictstm in the standard script becomes ictstm() in the server.R script)

Defining shiny inputs and outputs

There are two major lists that enable communication between the front-end and back-end:

input list

- These are mostly defined in the UI
- There are many different types, depending on the type of input
- Each has an id, which can then be used within functions to reference the input value. For example, a numeric input with an id = 'number' can be referenced in a function but using `input$number`
- Examples of inputs can be found here: https://shiny.rstudio.com/tutorial/written-tutorial/lesson3/

```
#Input example:
#This is written in the ui
numericInputIcon(
  inputId = "Input_discount_Costs",
  label = "Discount for Costs:",
  min = 0,
  max = 100,
  value = 3.5,
  icon = list(NULL, icon("percent"))
)
```

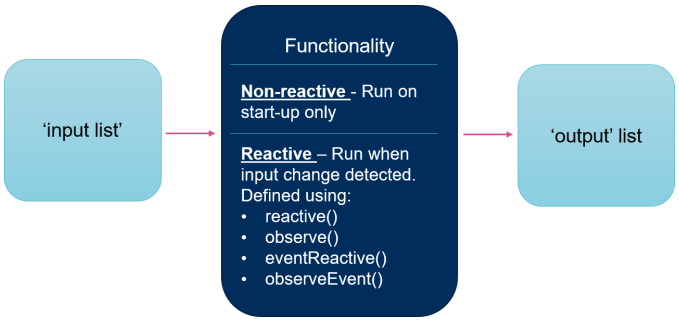

Defining shiny inputs and outputs

output list

- The results of your functions that you want to then display front-end are defined in the outputs list
- As with inputs, there are a variety of functions that can be used depending on the type of output you want to display
- Each has an id. The location of the output is defined in the ui, and then the output content is defined in the server

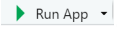
```
#Output example:
#This is written in the ui
plotOutput("Results_graph")

#This is written in the server
output$Results_graph <- renderPlot({
    autoplot(ictstm()$stateprobs_, labels = labs_indiv,
             ci = FALSE) + theme_bw()
  })
```

Summary of shiny



app.R

Please open the *hesim shiny app* folder within the *Additional useful packages* folder. Open the *app.R* script.

If you have all the packages previously installed, you should be able to run the app either by clicking the 'Run App' button in the top right hand corner, 

or by highlighting all the text and pressing **Ctrl** + **Enter**

Familiarize yourself with the layout of the app. All the functionality and graphs in the *hesim example model* are presented here

Shiny app learning and further examples

There are some great tutorials for learning how to create shiny apps:

1. https://mastering-shiny.org/basic-app.html
2. https://shiny.rstudio.com/tutorial/

There are also some great examples of shiny apps that others have created:

1. R Studio gallery - https://shiny.rstudio.com/gallery/
2. BCEA web application - https://gianluca--gianlucabaio.netlify.app/software/bceaweb/
3. HTA model proof-of-concept - https://bresmed-intrface-hypothetical-car-t-model.shinyapps.io/IntRface_Model-PharmacoEconomics/

Making shiny apps accessible

Shiny apps can be made available and deployed in a variety of ways, for example:

1. R Studio (or equivalent)
   - Local or online environment to view code and run app
2. shinyapps.io - https://www.shinyapps.io/
   - Online environment to deploy app publicly (free) or make available with log-in (paid), hosted on R Studio servers
3. R Studio Connect - https://www.rstudio.com/products/connect/
   - Hosting environment software to download on privately owned server (paid subscription)

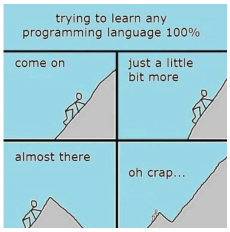
Session objectives (re-visited)

By the end of this session you should:

1) Be aware of additional useful packages in R that can be useful for building models, documenting them, and distributing them to a technical or non-technical audience

2) Know how to set up a default shiny app and rmarkdown script, and know locations of examples for further learning

3) Have considered learning pathways and objectives specific to your work requirements



Thank you for joining this ISPOR 2023 Short Course, Copenhagen

Felicity Lamrock, PhD; Howard Thom, PhD; Gianluca Baio, PhD; Rose Hart, PhD

Thank You!

Please complete your

Course Evaluation Link located in the Community in “Announcements”

Your Certificate of Course Completion can be downloaded from your ISPOR member profile