

# Session 3 - Probabilistic Markov models

Howard Thom

## 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

## 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

# Making transition matrices probabilistic

## 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

## 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	<i>Beta (85, 15)</i>
Not smoking at baseline	8	92	<i>Beta (8, 92)</i>

- 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

## Probabilistic analysis – transition matrices

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

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

- Similarly, the SoC transition matrix is represented by

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

SoC alone	Smoking at 6 months	Not smoking at 6 months	Distribution
Smoking at baseline	88	12	<i>Beta</i> (88, 12)
Not smoking at baseline	8	92	<i>Beta</i> (8, 92)

## Probabilistic analysis – beta distribution in R

- The `rbeta()` function takes a number of samples 'n' and its  $\alpha$  and  $\beta$  parameters

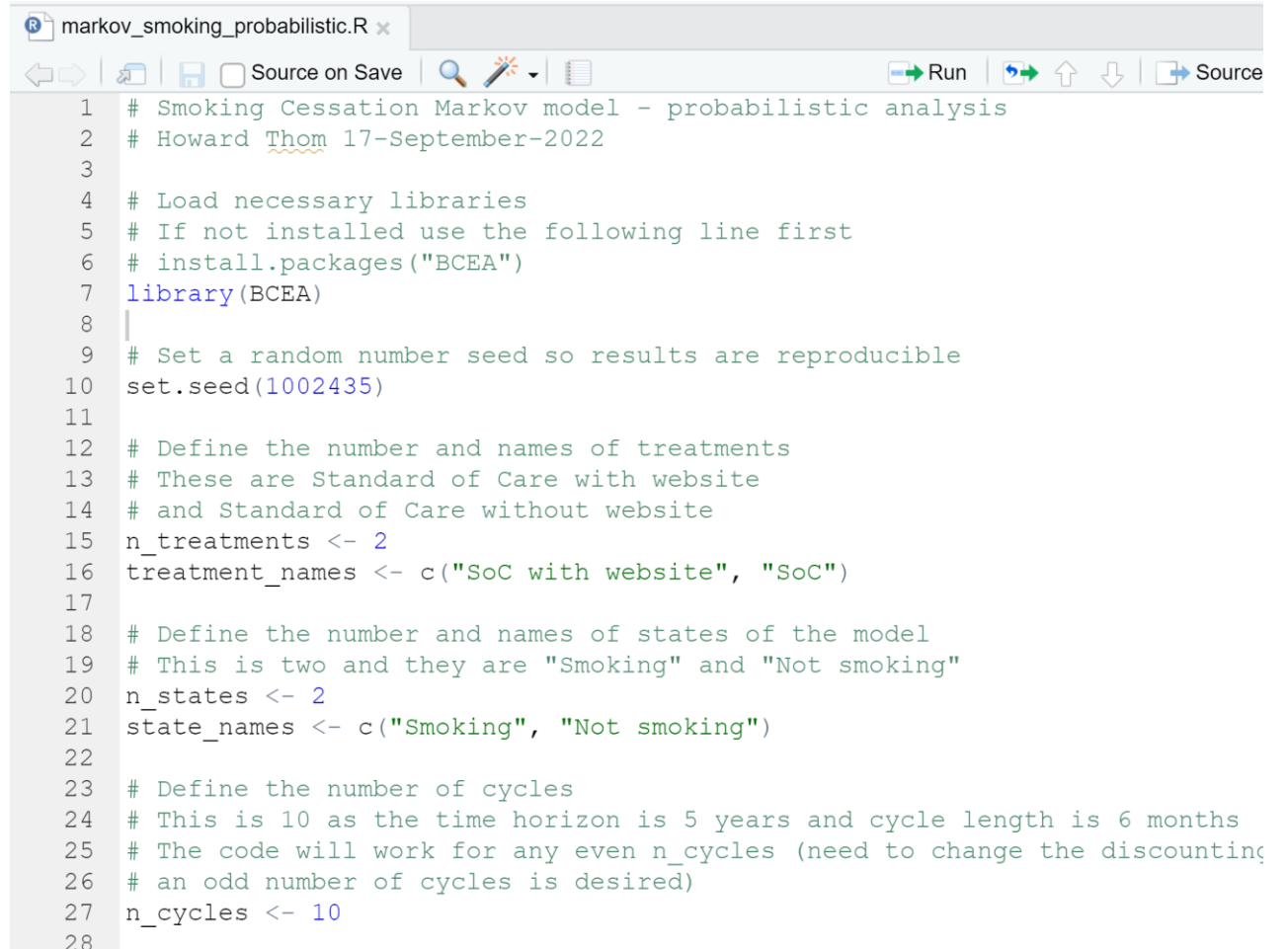
SoC + website	Smoking at 6 months	Not smoking at 6 months	Distribution
Smoking at baseline	85	15	<i>Beta (85, 15)</i>
Not smoking at baseline	8	92	<i>Beta (8, 92)</i>

```
> 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



```
1 # Smoking Cessation Markov model - probabilistic analysis
2 # Howard Thom 17-September-2022
3
4 # Load necessary libraries
5 # If not installed use the following line first
6 # install.packages("BCEA")
7 library(BCEA)
8
9 # Set a random number seed so results are reproducible
10 set.seed(1002435)
11
12 # Define the number and names of treatments
13 # These are Standard of Care with website
14 # and Standard of Care without website
15 n_treatments <- 2
16 treatment_names <- c("SoC with website", "SoC")
17
18 # Define the number and names of states of the model
19 # This is two and they are "Smoking" and "Not smoking"
20 n_states <- 2
21 state_names <- c("Smoking", "Not smoking")
22
23 # Define the number of cycles
24 # This is 10 as the time horizon is 5 years and cycle length is 6 months
25 # The code will work for any even n_cycles (need to change the discounting
26 # an odd number of cycles is desired)
27 n_cycles <- 10
28
```

## 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

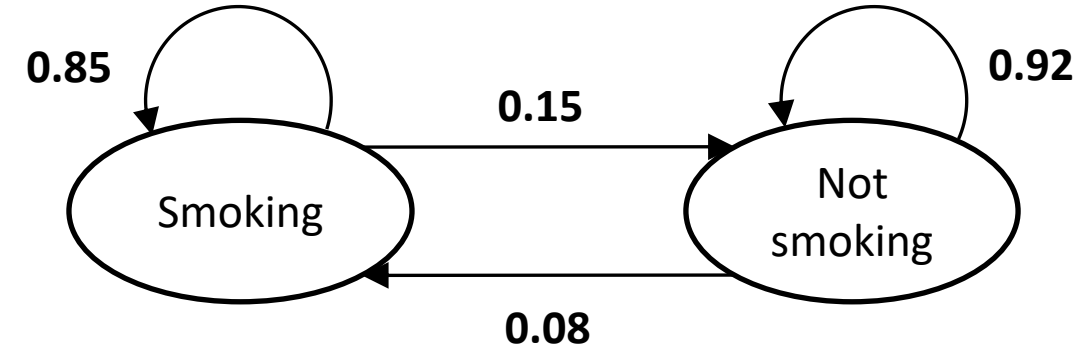
# Define simulation parameters
# This is the number of samples to use
n_samples <- 1000
```

# 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 10<sup>th</sup> 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
```

## 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.

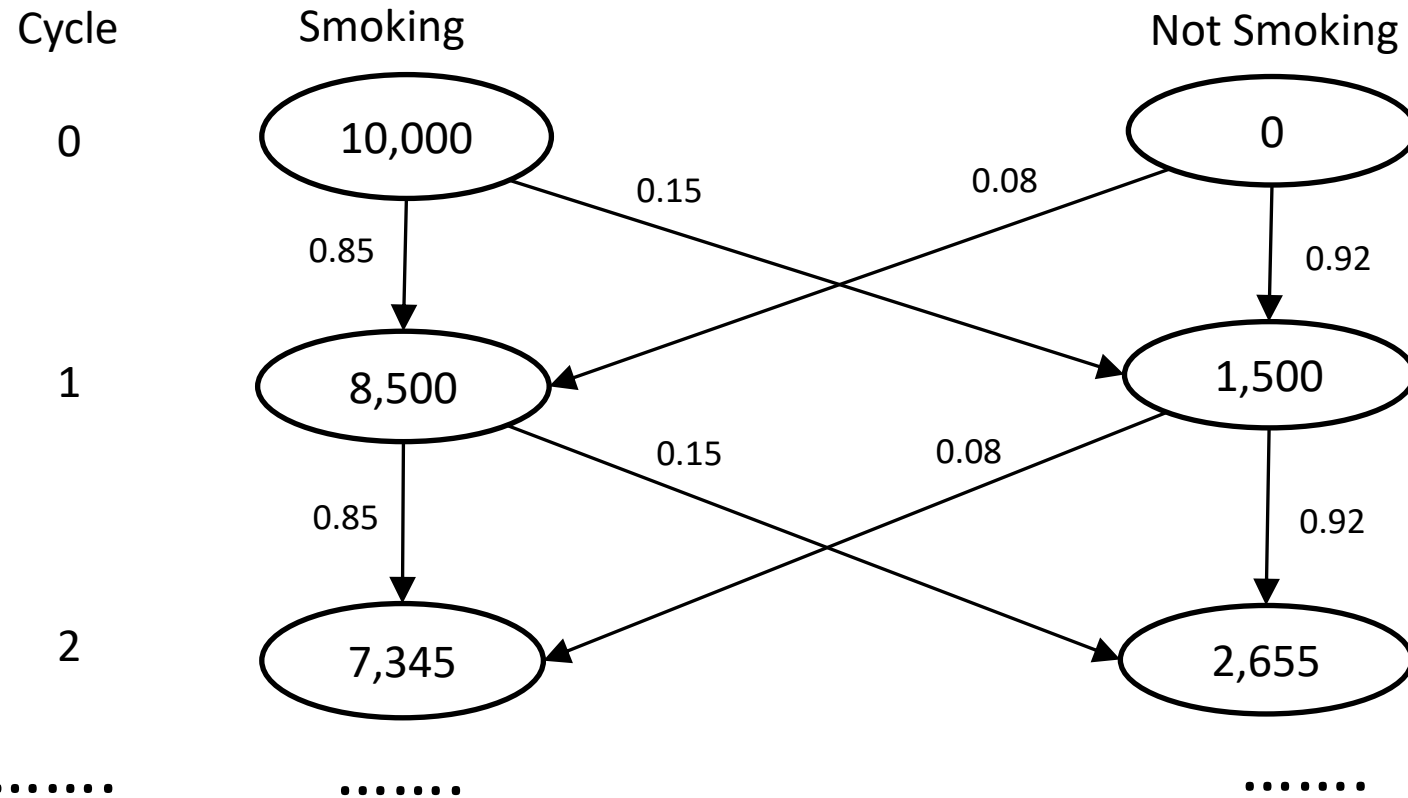
- c) What about the transition probabilities from not smoking? Do they differ between SoC with website and SoC alone?

# Markov simulation in probabilistic analysis



- Cohort vector  $\pi$  at time  $t$  ( $\pi_t$ ) is the cohort vector at the previous time point ( $\pi_{t-1}$ ) 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  $\pi_t$  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
  }
}
```

# 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

# Making costs and QALYs probabilistic

## 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

```
# 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
```



```
# 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

```
# 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

- 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 accounts 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)



# Analysing results in probabilistic analysis

# 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)
```

## BCEA output

Cost-effectiveness analysis summary

Reference intervention: SoC with website

Comparator intervention: SoC

Optimal decision: choose SoC for  $k <$  and for  $k \geq$

Analysis for willingness to pay parameter  $k = 20000$

	Expected net benefit
SoC with website	90500
SoC	90298

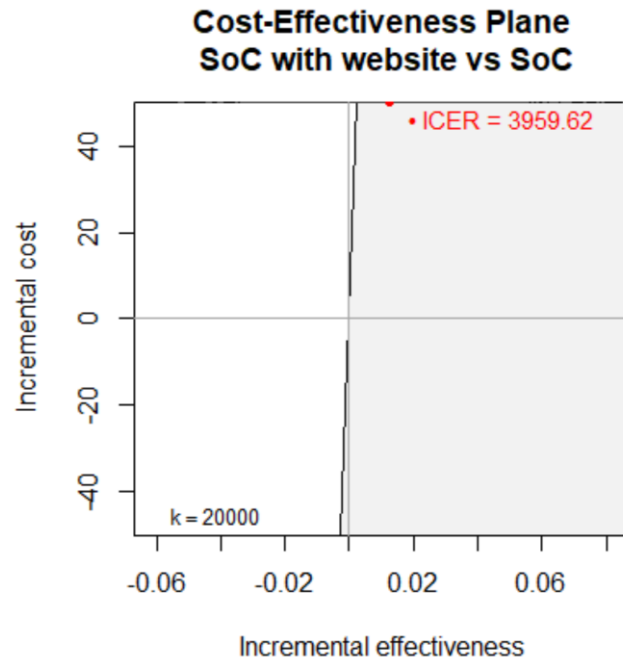
	EIB	CEAC	ICER
SoC with website vs SoC	202.55	0.688	3959.6

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

EVPI 81.375

# Cost-effectiveness plane

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

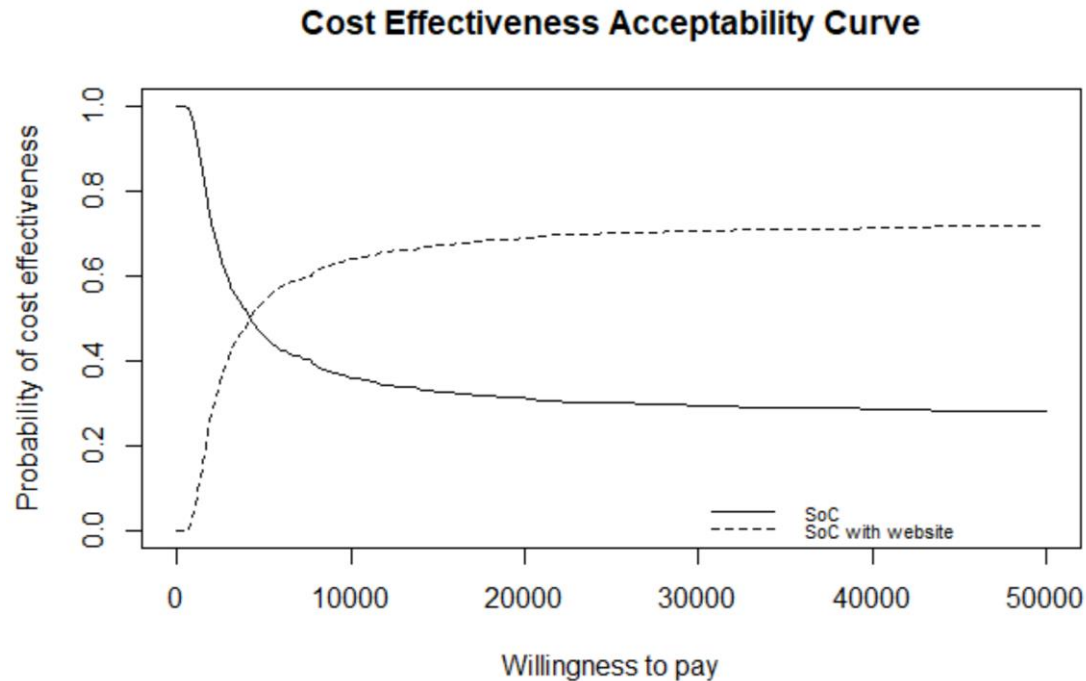


- Cost differential is always £50 as cost of website is fixed
- Variation in effectiveness over simulated sample plotted



# 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

- 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

## 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?

## 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)
```

## Exercise 2 – Solution

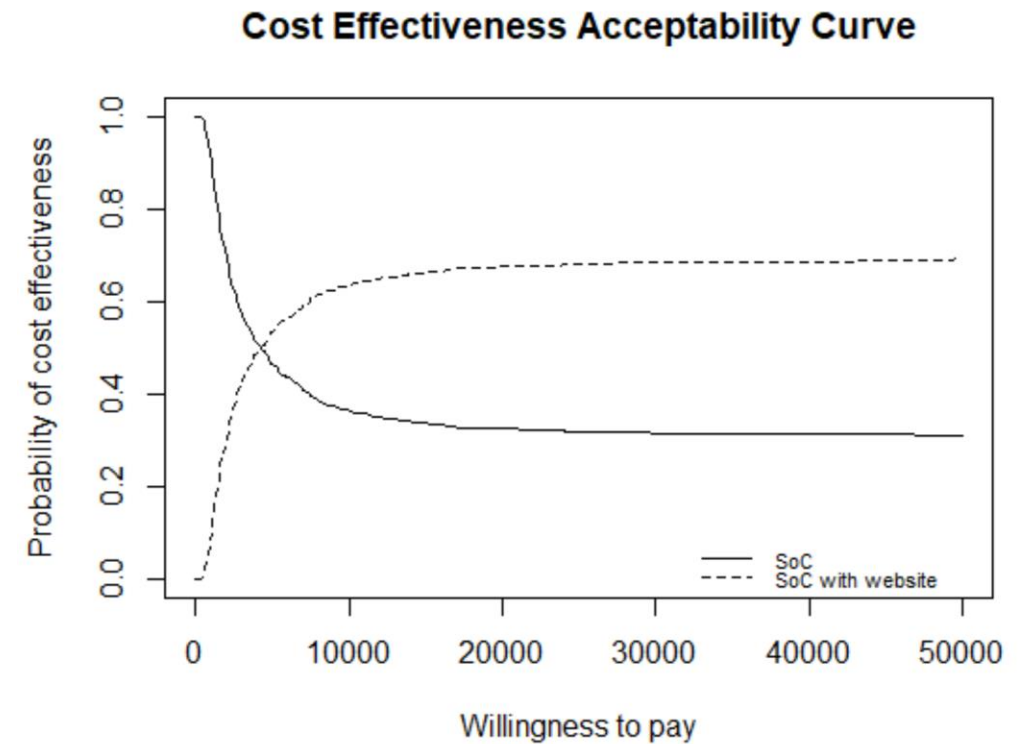
Using `summary()` of the new `bcea` object should give

	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`



# Thank you!

Dr Howard Thom

Senior Lecturer, University of Bristol

[Howard.thom@bristol.ac.uk](mailto:Howard.thom@bristol.ac.uk)