University of BRISTOL

Discrete-time cohort Markov models in R

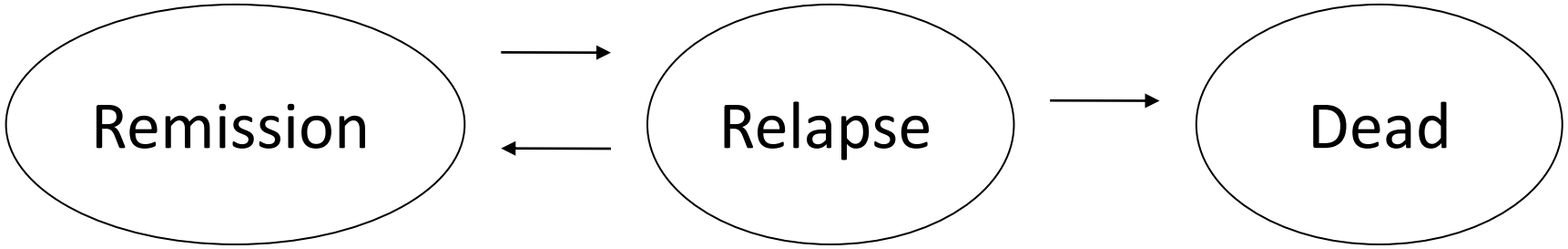
Howard Thom

Overview

- Brief introduction to smoking cessation Markov model
- R code for Smoking Cessation Markov model
  - Generating transition matrices
  - Markov cohort simulation
  - Summing costs and QALYs
  - Analysing results with BCEA
- Presented code is fully probabilistic. A deterministic version is provided as supplementary code.
- Practical exercises

Discrete-time cohort Markov models

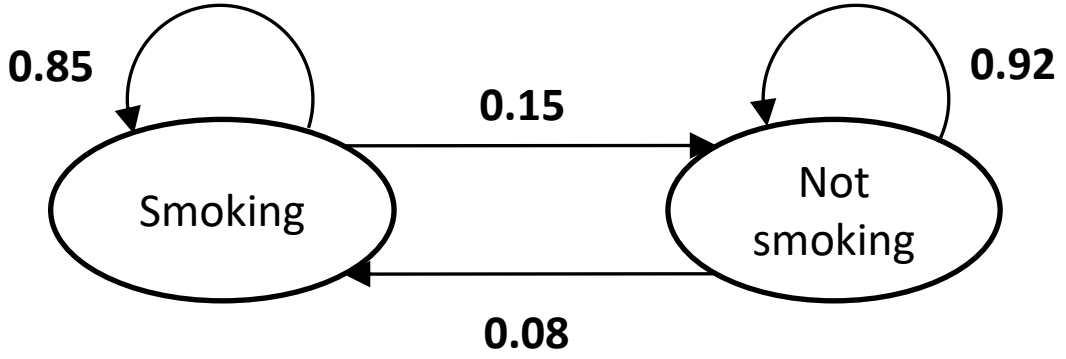
- Cohort models consider only aggregate behaviour of patient groups, averaging over any individual behaviour
- Multistate models divide health conditions into discrete set of mutually exclusive states
- Rates of transition are determined by transition probabilities



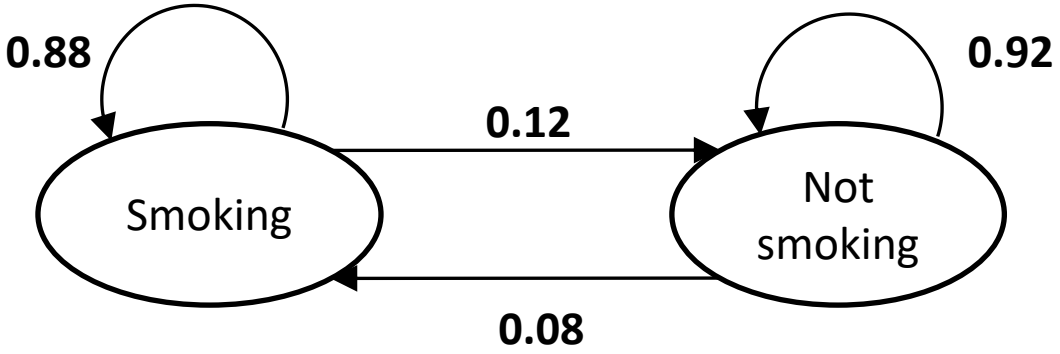
- Markov models are a type of multi-state model where transition probabilities are independent of time in state or past patient history

Example – smoking cessation

- 6-month cycles, 5 year time horizon



Standard Care + Website

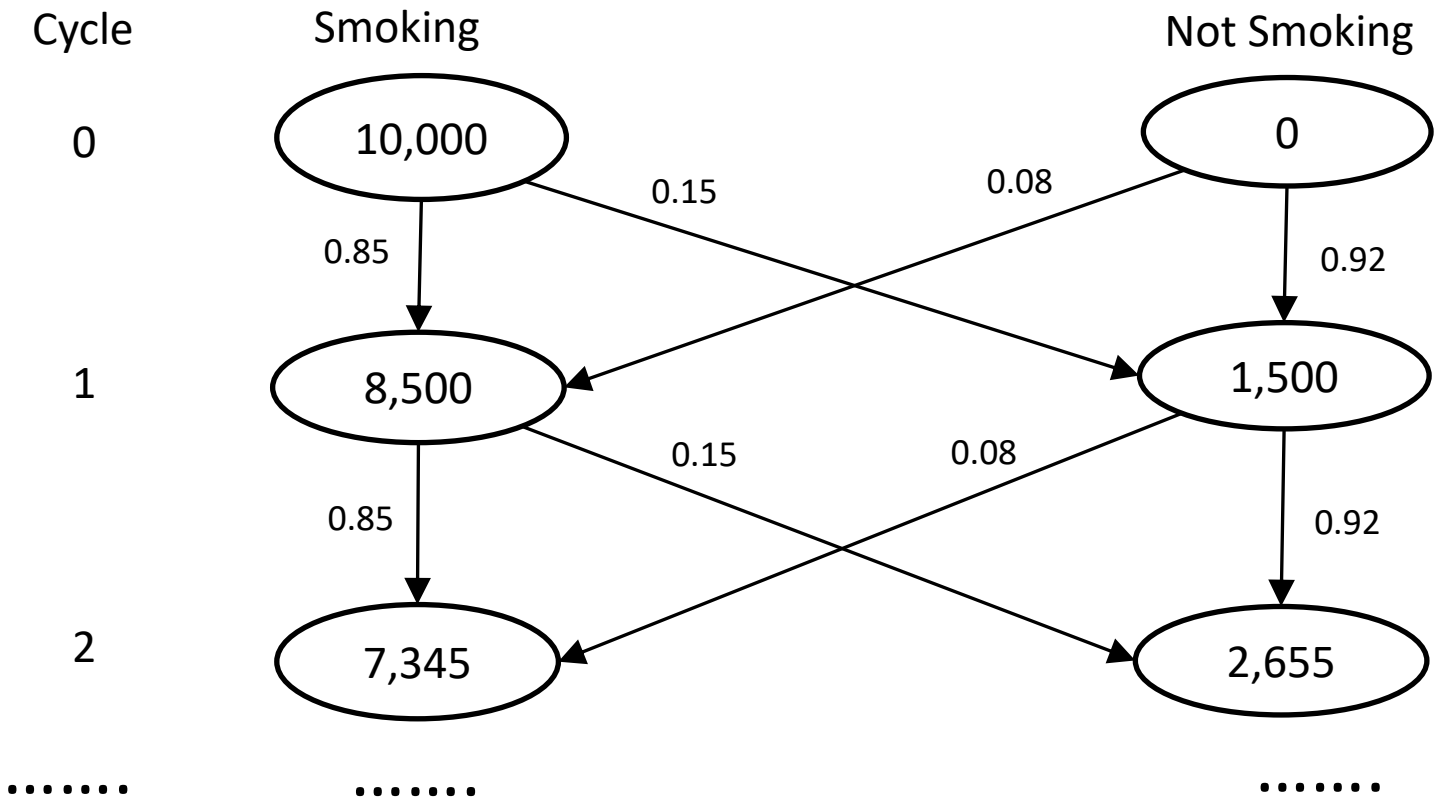


Standard Care

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

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



Updating cohort vector

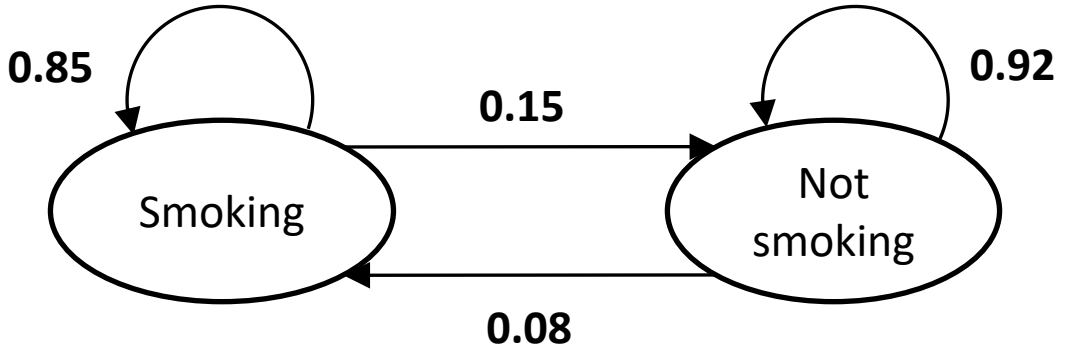
- Or in components

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

- 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)*

$$\left((0.6 * 0.85) + (0.4 * 0.08), (0.4 * 0.92) + (0.6 * 0.15)\right) = (0.465, 0.458)$$



Standard Care + Website

Probabilistic sensitivity analysis – transition matrices

- However, input parameters to the smoking cessation model are uncertain.
- The transition matrix for standard of care + website is not fixed as $\begin{pmatrix} 0.85 & 0.15 \\ 0.08 & 0.92 \end{pmatrix}$
- We instead assume that each row is a Dirichlet distribution

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

- This assumes that we had two sets of 100 patients followed over 6 months starting in smoking and non-smoking states and receiving standard of care + website.
- Similarly for the standard of care transition matrix instead of being fixed as $\begin{pmatrix} 0.88 & 0.12 \\ 0.08 & 0.92 \end{pmatrix}$, follows

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

University of BRISTOL

Why use the Dirichlet distribution?

- The Beta distribution is a natural choice for representing uncertainty when data are binomial e.g. number of quitters out of total cohort
- The Dirichlet is an extension of the Beta distribution for modelling probabilities of two or more disjoint events. The distribution creates *n* positive numbers (a set of random vectors $X_1 \ldots X_n$) that add up to 1. This is useful when there are more than two possible outcomes e.g. smoking, not smoking and death
- When there are only two outcomes e.g. smoking or not smoking, the Dirichlet is exactly equivalent to the beta distribution
- We will simulate from Dirichlet in R and get some intuition through experience…

Dirichlet distributions in R

- Use the VGAM library
- The rdiric function takes a number of samples ‘n’ and a set of parameters of the Dirichlet
- These parameters represent observed data of 85 “continued smoking” out of a total of 100 (85+15) smokers

```
> library(VGAM)
> rdiric(n=10,c(85,15))
              [,1]      [,2]
 [1,] 0.8278931 0.1721069
 [2,] 0.8203360 0.1796640
 [3,] 0.8761396 0.1238604
 [4,] 0.8380888 0.1619112
 [5,] 0.8272704 0.1727296
 [6,] 0.7811034 0.2188966
 [7,] 0.8664455 0.1335545
 [8,] 0.7944793 0.2055207
 [9,] 0.8661085 0.1338915
[10,] 0.8785183 0.1214817
```

Dirichlet distributions in R

- These parameters represent observed data of 88 “continued smoking” out of a total of 100 (88+12) smokers

```
> rdiric(n=10,c(88,12))
             [,1]       [,2]
 [1,] 0.9107257 0.08927432
 [2,] 0.8723911 0.12760886
 [3,] 0.8907410 0.10925899
 [4,] 0.8272278 0.17277215
 [5,] 0.8791464 0.12085365
 [6,] 0.9568808 0.04311916
 [7,] 0.9205452 0.07945481
 [8,] 0.8464086 0.15359138
 [9,] 0.8945235 0.10547646
[10,] 0.9120315 0.08796855
```

University of BRISTOL

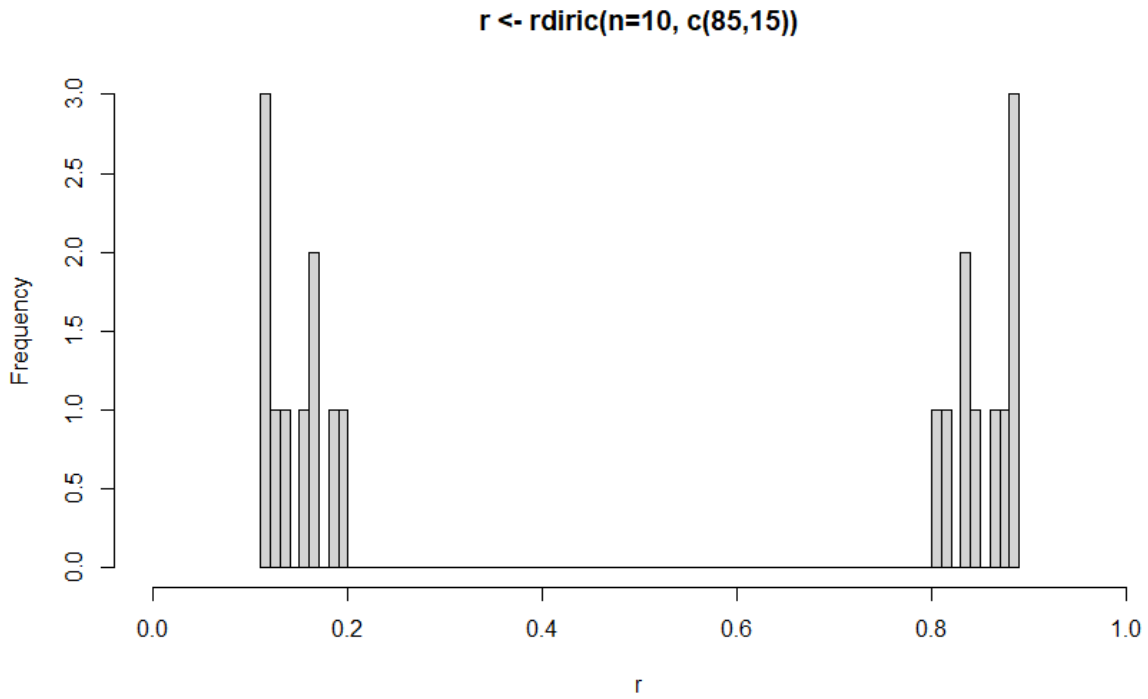
Dirichlet distributions in R

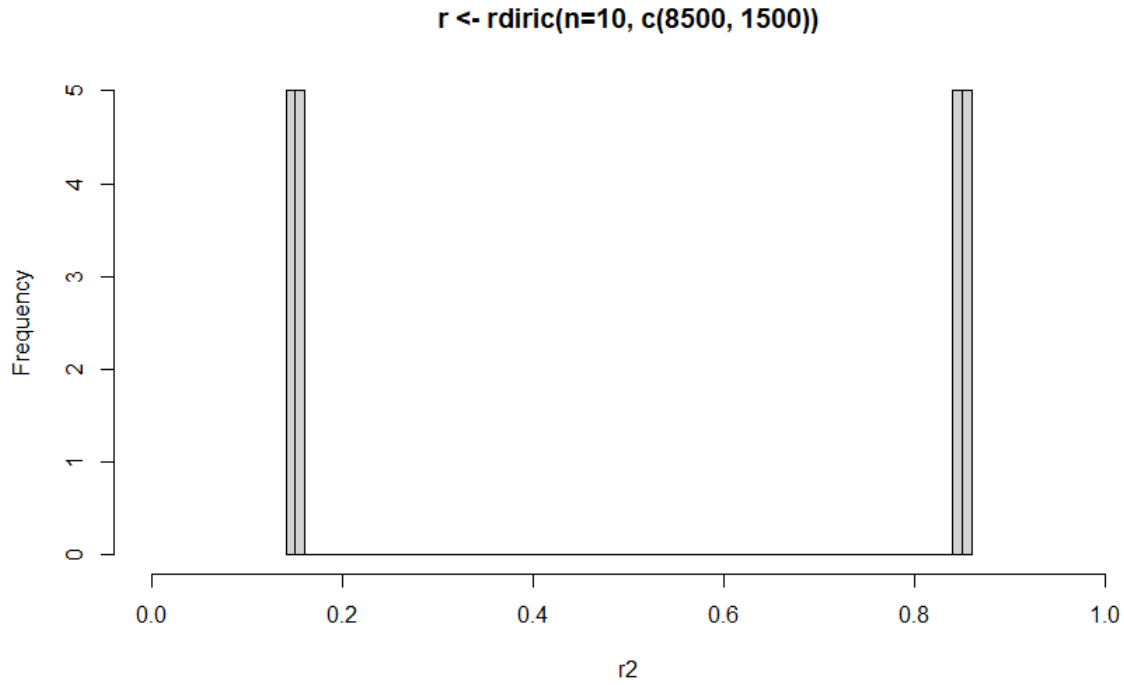
- Increasing the total number of observations to 10,000 reduces variation in sample of transition probabilities.

- In smoking cessation example we assume we only had cohorts of 100 smokers on two treatment options.

```
> rdiric(n=10,c(8500,1500))
             [,1]      [,2]
 [1,] 0.8471364 0.1528636
 [2,] 0.8460362 0.1539638
 [3,] 0.8539317 0.1460683
 [4,] 0.8496491 0.1503509
 [5,] 0.8563986 0.1436014
 [6,] 0.8499067 0.1500933
 [7,] 0.8557487 0.1442513
 [8,] 0.8488775 0.1511225
 [9,] 0.8515815 0.1484185
[10,] 0.8496059 0.1503941
```

Dirichlet distributions in R





Open the file

- If you haven't already, use R or Rstudio to open the file labelled "markov_smoking_probabilistic.R"

markov_smoking_probabilistic.R

Source on Save | Run | Source

```
1  # Smoking Cessation Markov model
2  # Howard Thom
3
4  # Load necessary libraries
5  # If not installed use the following line first
6  # install.packages("VGAM")
7  # install.packages("BCEA)
8  library(VGAM)
9  library(BCEA)
10
11 # Set a random number seed so results are reproducible
12 set.seed(1002435)
13
14 # Define the number and names of treatments
15 # These are Standard of Care with website
16 # and Standard of Care without website
17 n_treatments <- 2
18 treatment_names <- c("SoC with website", "SoC")
19
20 # Define the number and names of states of the model
21 # This is two and they are "Smoking" and "Not smoking"
22 n_states <- 2
23 state_names <- c("Smoking", "Not smoking")
24
25 # Define the number of cycles
26 # This is 10 as the time horizon is 5 years and cycle length is 6 months
27 # The code will work for any even n_cycles (need to change the discounting code if
28 # an odd number of cycles is desired)
29 n_cycles <- 10
30
```

University of BRISTOL

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

n_samples <- 1000
```

An array to store transition matrices

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

- For each treatment and each PSA sample there is one 2x2 transition matrix
- This produces an array with dimensions 2x1000x2x2
- They are currently blank (NA) 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
transition_matrices["SoC with website", , "Smoking", ] <- rdiric(n_samples,
                                                                 c(85, 15))
# Transitions from not smoking
transition_matrices["SoC with website", , "Not smoking", ] <- rdiric(n.samples,
                                                                 c(8, 92))

# Second the transition matrix for Standard of Care
# Transitions from smoking
transition_matrices["SoC", , "Smoking", ] <- rdiric(n_samples, c(88, 12))
# Transitions from not smoking – same as for SoC with website as website does not prevent relapse
transition_matrices["SoC", , "Not smoking", ] <- transition_matrices["SoC with
                                                                website", , "Not smoking", ]
```

Contents of array?

- Run the previous code ensuring you have filled in the *transition_matrices* array
- Now look at elements of the array
- For example the first sampled transition matrix for standard of care:

```
> transition_matrices["SoC",1,,]
                Smoking Not smoking
Smoking     0.95373294  0.04626706
Not smoking 0.07513275  0.92486725
```

Contents of array?

- Or the 10th sample for standard of care with website

```
> transition_matrices["SoC",10,,]
              Smoking Not smoking
Smoking     0.85944133   0.1405587
Not smoking 0.06444297   0.9355570
```

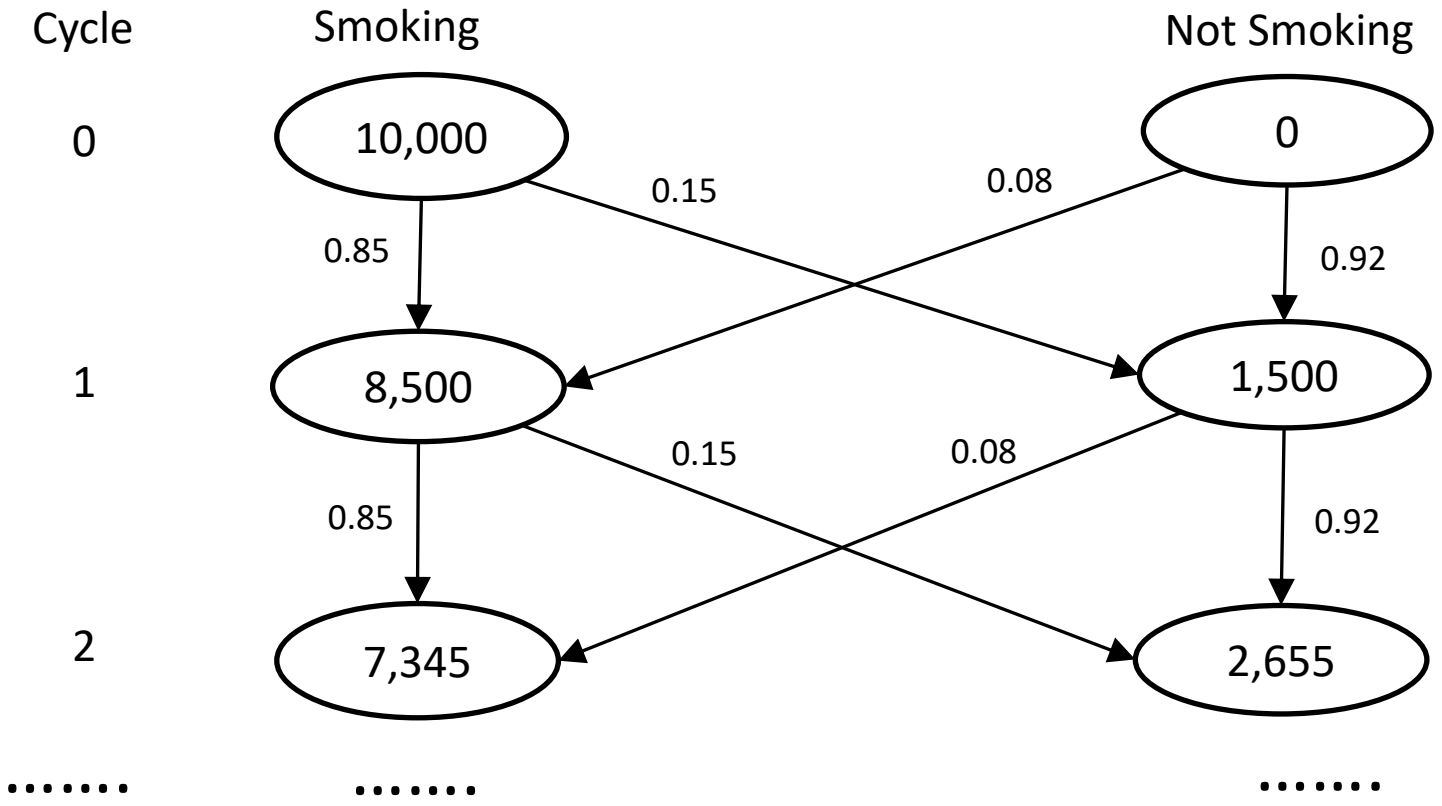
- Or first 10 samples of transition probabilities from 'Smoking' on standard of care with website

```
> transition_matrices["SoC with website",1:10,"Smoking",]
        Smoking Not smoking
 [1,] 0.8757491   0.1242509
 [2,] 0.8682170   0.1317830
 [3,] 0.8010984   0.1989016
 [4,] 0.8629500   0.1370500
 [5,] 0.8450373   0.1549627
 [6,] 0.8514735   0.1485265
 [7,] 0.8284889   0.1715111
 [8,] 0.8671505   0.1328495
 [9,] 0.8384200   0.1615800
[10,] 0.8383359   0.1616641
```

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

- For each treatment, PSA sample, and cycle, there is one cohort vector
- These are the two-dimensional π_t in the Markov formula

```
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
```

Core loop

Loop over treatments

{

Loop over PSA samples

{

Loop over cycles

{

Update cohort vector

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

or specifically…

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

}

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

}

}

Core loop

```
# Loop over the treatment options
for (i_treatment in 1:n_treatments)
{
        # Loop over the PSA 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 PSA sample
                2. Calculate total costs and QALYs for this PSA sample
        }
}
```

Core loop

```
# Loop over the treatment options
for (i_treatment in 1:n_treatments)
{
        # Loop over the PSA 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 PSA sample
                2. Calculate total costs and QALYs for this PSA sample
        }
}
```

This will be implemented next

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 $\sum_{t=1}^{T} \pi_t \times q_t$
- Finally, the net benefit is

$$NB = \lambda \sum_{t=1}^{T} \pi_t \times q_t - \sum_{t=1}^{T} \pi_t \times c_t$$

University of BRISTOL

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
- $c_t = (0,0)$ as no cost per state
- But there is a one-off treatment cost $d_i = (50,0)$ for standard care + website ($i = 1$) and standard of care alone ($i = 2$)
- $q_t = (0.475, 0.5)$ as the QALYs per cycle in smoking and non-smoking states
- The net benefit for option i is therefore

$$NB = \lambda \sum_{t=1}^{T} \pi_t \times q_t - d_i$$

Defining state QALYs

- The QALYs associated with 1 year in the smoking state are now normally distributed as Normal(mean = 0.95, sd = 0.01)
  - Divide by 2 as 6-month cycles in the smoking state.
- The QALYs associated with 1 year in the non-smoking state remain fixed at 1.00 (perfect health)

```
state_qalys <- array(dim = c(n_samples, n_states), dimnames = list(NULL,
state_names))
state_qalys[, "Smoking"] <- rnorm(n_samples, mean=0.95, sd=0.01) / 2
state_qalys[, "Not smoking"] <- 1 / 2
```

University of BRISTOL

Defining state and treatment costs

- Cost of website also remains fixed as £50
- Treatment costs are fixed at £50 for website and £0 for SoC

```
state_costs <- array(0, dim = c(n_samples, n_states), dimnames = list(NULL,
state_names))

treatment_costs <- array(dim = c(n_treatments, n_samples), dimnames =
list(treatment_names, NULL))
treatment_costs["SoC with website", ] <- 50
treatment_costs["SoC", ] <- 0
```

Cycle costs and cycle 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))
```

- For each treatment and PSA sample, there is one accrued cycle cost and QALY
- These are filled in by the model loop
- Not strictly necessary to store these but might be interested in costs or QALYs accrued per cycle.

Arrays to store total cost and 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 PSA 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 PSA 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 PSA sample
        }
}
```

Implement this final step in R

Discount factor

- Annual discount factor is 3.5%
- The discount factor for each 6-month cycle is calculated using

```
(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 total costs and QALYs

- Sum and discount the cycle costs and QALYs
- Add treatment costs to total costs

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

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.

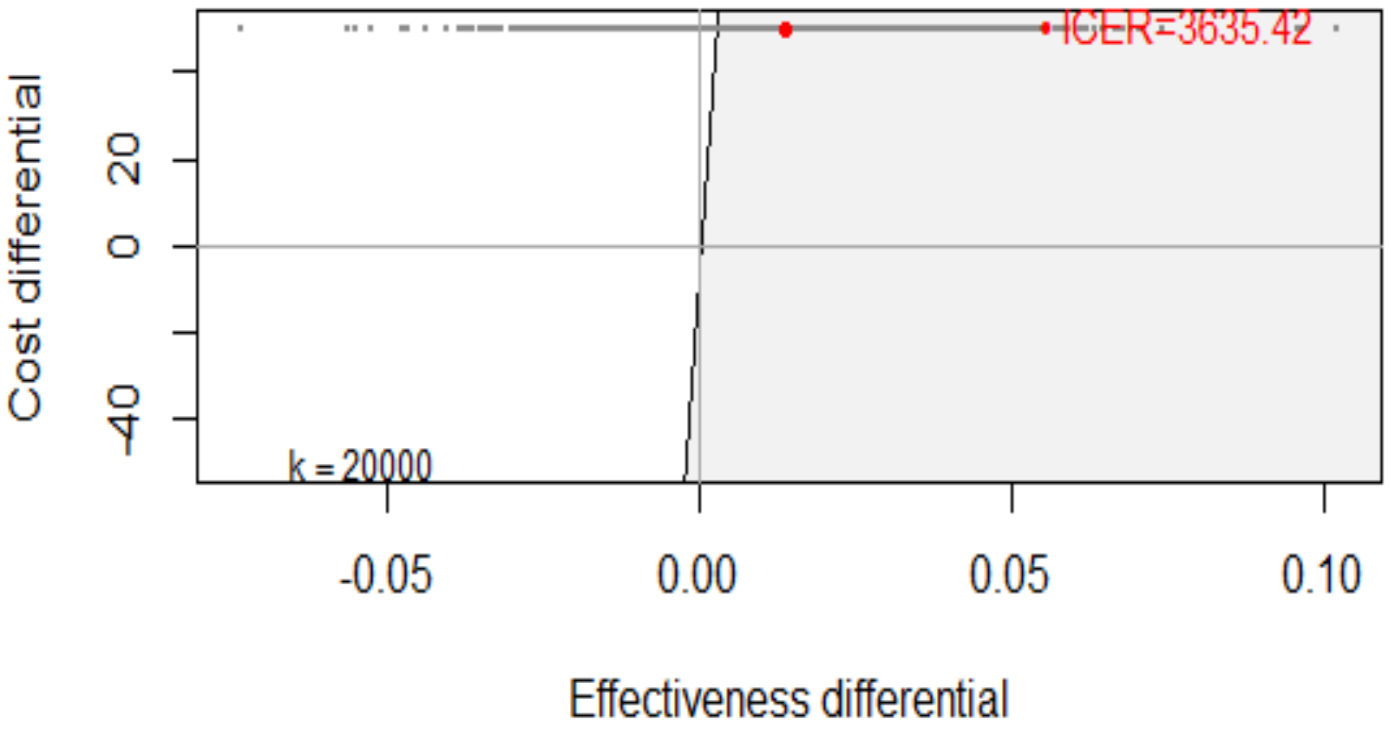
```
Smoking_bcea <- bcea(e = t(total_qalys), c = t(total_costs), ref = 1,
interventions = treatment_names)

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<3700 and SoC with website for k>=3700


Analysis for willingness to pay parameter k = 20000

                 Expected utility
SoC with website            90562
SoC                         90337

                           EIB  CEAC   ICER
SoC with website vs SoC 225.07 0.691 3635.4

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

EVPI 82.037
```

Cost-effectiveness plane

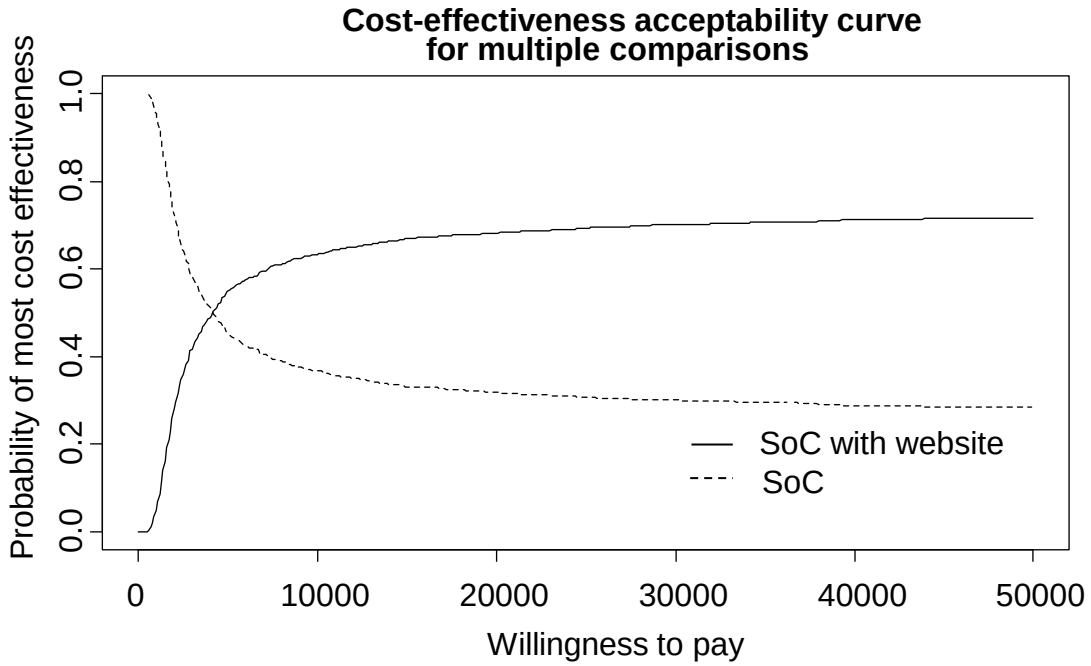
```
ceplane.plot(smoking_bcea, wtp = 20000)
```



- SoC with website vs SoC
- 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)
mce.plot(smoking_multi_ce, pos = c(1,0))
```



- SoC is optimal up to £3700 willingness-to-pay per QALY
- Above £4k SoC with website is optimal
- Can also use `ceac.plot(smoking_bcea)` to plot only reference intervention

University of BRISTOL

Practical

- Please complete Exercises 1 and 2

Exercise 2 adding state costs solution

Using summary() of the new bcea object should give

	EIB	CEAC	ICER
SoC with website vs SoC	258.52	0.715	1571.8

ICER approximately halves because difference in costs is reduced (i.e. website is now saving money as well as improving quality of life)

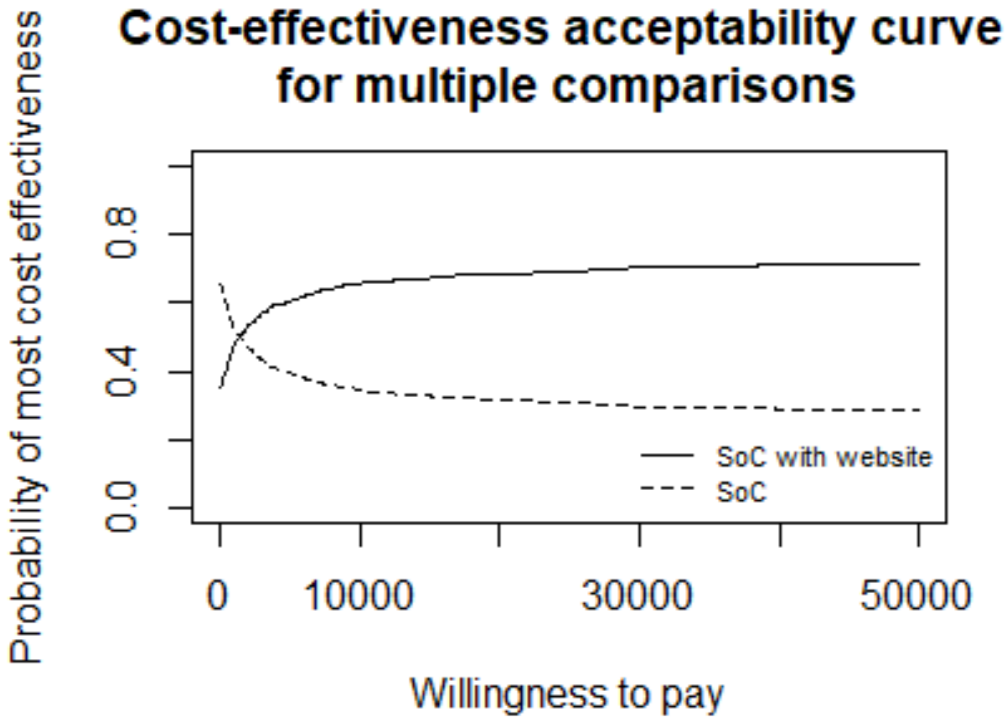
> mean(incremental_costs)

£24.74059

> mean(incremental_effects)

0.01257059

But the probability it is the best remains the same as this is driven by the cases where the incremental effects are less than zero.



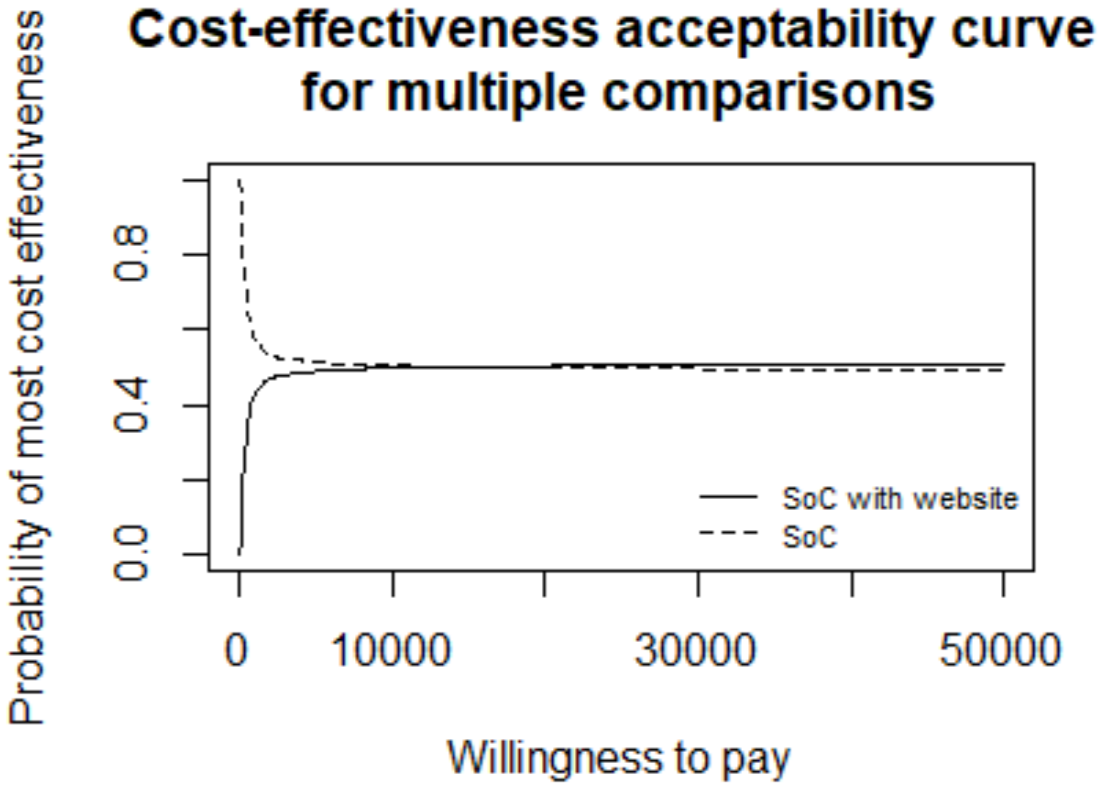
Exercise 3 adding dead state solution

Using summary() of the new bcea object should give

	EIB	CEAC	ICER
SoC with website vs SoC	431.13	0.518	2078.4

The ICER is reduced because the difference in effects is increased.

However, the uncertainty is increased so the CEAC is closer to 50%.



Summary

- We have explained the 2-state and 2-treatment option smoking cessation Markov model.
- We have explained the key steps in building a Markov model in R
  - Define input parameters
  - Update cohort vector and calculate total costs and QALYs
  - Analyse results
- The code we provided is general
  - For example, included state costs even though these are zero in smoking cessation
  - Change numbers of states and input parameters to adapt to other examples

Back up slides

Discount factor

- We are discounting at 3.5% per year.
- So factor in first year is 1, second year is 1.035^{-1}, third is 1.035^{-2}, …, fifth is 1.035^{-4}
- But cycle is 6 months so actually 1 for first two cycles, 1.035^{-1} for third and fourth cycle, …, 1.035^{-4} for ninth and 10th cycle.
- 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
```

Discount factor

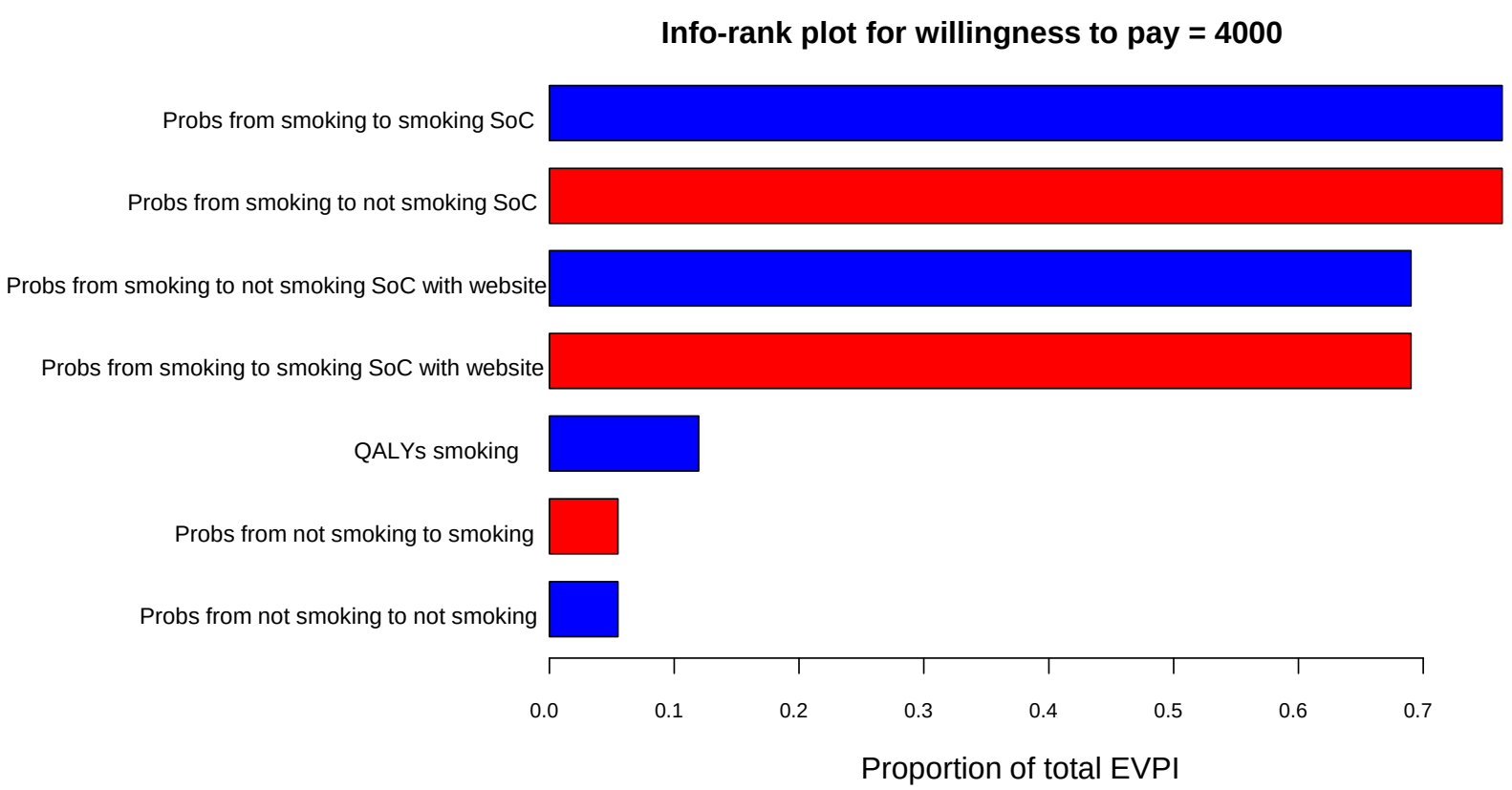
- The discount factor is then

```
(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
```

Deterministic Sensitivity Analysis using EVPI

```
info.rank(input_parameters$parameters, input_parameters$mat, smoking_bcea)
```



- info.rank() function can estimate proportion of total decision uncertainty, quantified by expected value of perfect information (EVPI) , to which each of the uncertain parameters contribute
- Need to specify all uncertain parameters and feed in their simulated samples