DECISION MODELLING
F **HEALTH ECONOMIC**
O
R **E V A L U A T I O N**

Decision Modelling for Health Economic Evaluation R Stream Reader - Foundation Course

Foundation Course Reader

Nichola Naylor & Jack Williams

May/June 2021 Course Version 2

Introduction

Welcome to the R Stream of Decision Modelling for Health Economic Evaluation Reader for the Foundation Course!

Over the course of the next few modules we will cover how to code Decision Trees (Module 2), Markov Models (Module 3) and Expected Value of Clinical & Diagnostic Information (Module 4).

This course stream is not intended to teach you how to use R, but rather to teach potential ways of constructing health economic evaluation models within R. For more information on how to use R, seek readily available resources such as:

- Free R books: http://www.cookbook-r.com

- Free forums online can solve *almost* any problem: http://www.stackoverflow.com

There are often many ways to do the same thing in R. Throughout this course we present one potential way of coding things, often opting for code which is easier to follow (for example, choosing a loop process over something more efficient but less intuitive), and also tending to use base R functions wherever possible. This is not to say another way is 'wrong' or that the solutions provided are the only way to code these models. We encourage you to play around with the code, checking whether the results align across the different methods. In some instances, we have also provided two different lines of code to produce the same output, to demonstrate the different approaches available.

R Exercise Structure

Each Module (excluding Module 1 which does not have an R exercise component) has the following:

- An instruction pdf (e.g. "F2.3.1_Decision_Trees_Instructions.pdf" which refers to Foundation Module 2, R stream, File 1): This provides a step-by-step guide for each exercise, providing important information on parameters and hints & tips on how to code your models. This should be used in conjunction with. . .

- A template R script (e.g. "F2.3.2_Decision_Trees_Template.R" which refers to Foundation Module 2, R stream, File 2): This provides an outline of the model code, with key variables, functions and/or loops to define. There are blank spaces to fill whilst following the Instructions pdf.

- A solutions R script (e.g. "F2.3.1_Decision_Trees_Solutions.pdf" which refers to Foundation Module 2, R stream, File 3) which provides a completed model code script, filled in for you. If you run this R script (having in mind the section below about directory set up), this will provide the fully functional models and results.

We mention 'sections' within the R scripts (such as the 'Parameters' section) for this course, similar to how you might refer to sheets in Excel. If you are using RStudio, sections can be selected using the bottom left of the Source panel or alternatively using the document outline functionality (top right of the Source panel).

Setting up your working directory

Once all of the above files have been downloaded from the course website, please make sure that you have a dedicated folder for the course code. This folder should become your working directory for the Foundation course.

If you want to manually set the working directory, then you can use the `setwd()` function, or alternatively, if you want to set the working directory automatically to the folder your are currently in, you can run the following code:

```r
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
```

Some R Reminders

We will use many base R functions throughout the course. You can see a simple cheat sheet available under the "Base R" category within the RStudio cheat sheet repository: https://www.rstudio.com/resources/cheatsheets/

Key terms to be familiar with before starting: vectors, matrices, functions, loops.

Vector

```r
(vector <- c(1, 2, 3))
```

```
## [1] 1 2 3
```

Matrix

```r
matrix(1:9, nrow = 3, ncol = 3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

Functions

```r
print.greeting <- function(){
  print("hello")
}
print.greeting()
```

```
## [1] "hello"
```

For Loop

```r
x <- c(1:5)
print(x)
```

```
## [1] 1 2 3 4 5
```

```r
for(i in 1:5){
  x[i] <- x[i]*2
}
print(x)
```

```
## [1]  2  4  6  8 10
```

Subsetting and selecting data

Another important aspect of the course is selecting the appropriate data. For the various vectors, data frames and matrices, here are some selections of data

Vector

```r
(vector <- c(10, 20, 30, 40))
```

```
## [1] 10 20 30 40
```

```r
vector[2] # Takes the second element of the vector
```

```
## [1] 20
```

```r
vector[3:4] # Takes the third and fourth element of the vector
```

```
## [1] 30 40
```

Matrix

```r
x <- matrix(1:9, nrow = 3, ncol = 3)
x
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```r
## we can provide column and row names to matrices:
colnames(x) <- c("a","b","c")
rownames(x) <- c("d","e","f")
x
```

```
##   a b c
## d 1 4 7
## e 2 5 8
## f 3 6 9
```

```r
## these can be used to call values
x["d","a"] ## remember: rowname first, then colname
```

```
## [1] 1
```

Data frame

```r
df <- data.frame(cost=1:5, effect=100:104)
print(df)
```

```
##   cost effect
## 1    1    100
## 2    2    101
## 3    3    102
## 4    4    103
## 5    5    104
```

```r
df[3,] # Third row (all columns)
```

```
##   cost effect
## 3    3    102
```

```r
df[,2] # Second column (all rows)
```

```
## [1] 100 101 102 103 104
```

```r
df[3,2] # Third row and second column value
```

```
## [1] 102
```

```
df$cost # We can also call columns by their column name
```

```
## [1] 1 2 3 4 5
```

```
df[1,"cost"] ## 1st row, cost value
```

```
## [1] 1
```

Other Information

Please do let us know (via the blog post tool) whether there are any issues in the R exercise information provided.

If you are doing the Advanced course following this course, there will be a separate reader with additional information related to material needed for the Advanced course.

Happy coding!