hesim pdf report

ISPOR short course

04 November 2025

Contents

Introduction

Producing automated reports for modelers and users is possible using R Markdown. This enables you to focus the writing of technical reports on the story surrounding the results and ensures that the outputs match the model. R Markdown has several features that make it ideal for this purpose:

1. Links directly to the code within the text and outputs
2. Allows passing of information from an R Shiny app to the markdown document script for rendering an up-to-date report
3. Reasonably flexible with respect to formatting, including tables, figures, Microsoft Word templates and so on
4. Easy to use and to understand the script documents
5. Allows presentation of the code used to generate a rendered output for simple and effective QC, communication and general transparency

The majority of any R Markdown document will be raw text, though the front matter may be the most important part of the document. The document will be generated that includes both content as well as the output of any embedded R code chunks within the document. There are a lot of cheat sheets and reference guides for Markdown and bookdown (e.g. Yihui Xie, Adam Prichard)

The purpose of this report is to demonstrate the use of R Markdown through outputting the results of the `hesim` example model.

Results

The results of the `hesim` example are passed to R Markdown via the `rmarkdown::render()` function. Within this function, `envir` is used to import the current user environment with loaded libraries. For instance, `ggplot2` was loaded in the main R script, so importing the script environment into R Markdown means it does not need to be loaded again here. This means that the data and values that were created in the R script are also loaded in. `params` can also be specified, which lists the data to be used in the report; this is particularly important if `rmarkdown::render()` is being called from a `shiny` application, as it may be that the outputs are dynamically generated. In this example, the state probability data frame, the summary data frame with the discounted outcomes, and the state and strategy labels were passed to R Markdown via the code below. The differences in how the information is passed from within the R script and from within the `shiny` application are shown.

In the code below, the `input` refers to the .Rmd file (this file), which lays out the R Markdown document. The output The `output_format` is the selected format of the document, in this case `"html_document"` is used to create a html output (see other options). `output_file` refers to the name of the output to be created, which needs to be consistent with the output format.

Gathering results from R script:

```
Note: This code chunk uses markdown instead of r. markdown shows the script
without running it, whereas r runs the script. Both markdown and r code chunks
look the same when rendered into the document
```

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

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

Gathering results from R shiny app:

```
output$Create_pdfreport <- downloadHandler(
  filename = "hesim-pdf-report.pdf",
  content = function(file) {

    ce_sim_ictstm <- ictstm()$summarize()

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

    # html document
    rmarkdown::render(
      input = "./hesim pdf report.Rmd",
      output_format = 'bookdown::pdf_document2',
      output_file = file,
      params = Export_params,
      envir = environment()
    )
  }
)
```

The two subsections below (as denoted by the **##** in the .Rmd script) below show how inputs from the main model can be used within the document

State transition probabilities

The code below is an r chunk, which means that is it run as the R Markdown document renders. It will therefore return the outputs of the code. `print(params$Stateprobs)` is called below, which prints the value of `params$Stateprobs`. If `echo = TRUE` is specified at the top of the chunk, then the code is also printed into the document.

```r
print(params$Stateprobs)
```

```
##           sample strategy_id grp_id state_id           t  prob
##           <num>       <int> <int>    <num>       <num> <num>
##       1:      1           1      1        1  0.00000000 1.000
##       2:      1           1      1        1  0.08333333 1.000
##       3:      1           1      1        1  0.16666667 1.000
##       4:      1           1      1        1  0.25000000 1.000
##       5:      1           1      1        1  0.33333333 0.999
##      ---
## 3248996:   1000           3      1        3 29.66666667 0.992
## 3248997:   1000           3      1        3 29.75000000 0.992
## 3248998:   1000           3      1        3 29.83333333 0.992
## 3248999:   1000           3      1        3 29.91666667 0.992
## 3249000:   1000           3      1        3 30.00000000 0.993
```

Figures can be printed into the document within the r chunks in exactly the same way as they are called in the main R script.

```r
hesim::autoplot(params$Stateprobs, labels = params$labs_indiv,
         ci = FALSE) + theme_bw() + ggplot2::theme(legend.position = "bottom")
```

Cost and survival summary

Tables can also be displayed in R Markdown documents in a similar way to figures. The `kableExtra` package can be used with `knitr` to format tables in the desired way.

Note that the `echo = FALSE` parameter can be added to code chunks to prevent printing of the R code that generated the output.
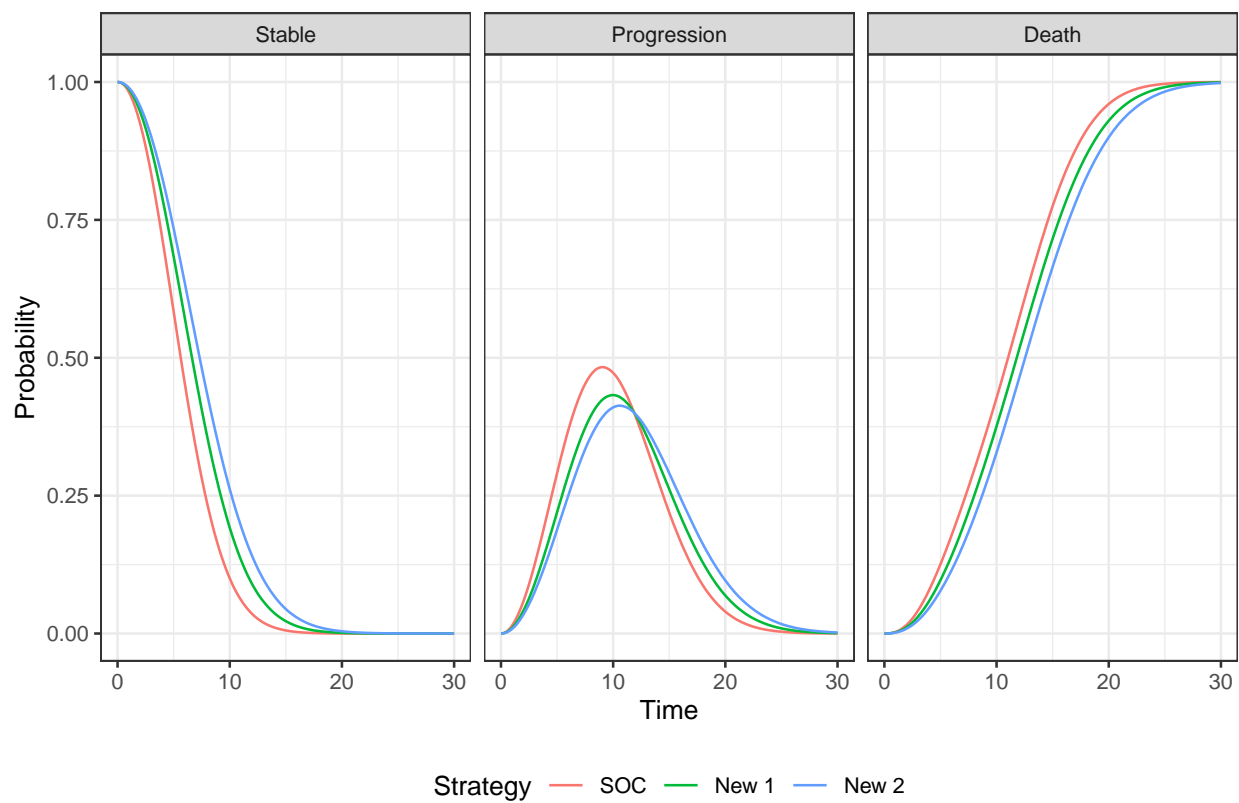
Figure 1: State transition probabilities over time

Table 1: Summary of total costs and QALYs

Discount rate	Outcome	SOC	New 1	New 2
0.00	QALYs	7.85 (7.05, 8.63)	8.58 (7.78, 9.41)	9.20 (8.34, 10.10)
0.00	Costs: Drug	18,114 (16,937, 19,287)	90,014 (83,743, 96,496)	122,012 (113,095, 131,154)
0.00	Costs: Medical	61,930 (6,934, 198,025)	62,556 (7,025, 194,522)	64,497 (7,390, 198,719)
0.00	Costs: total	80,044 (24,738, 216,495)	152,571 (96,714, 281,617)	186,509 (128,481, 320,137)
0.01	Costs: Drug	17,095 (16,040, 18,145)	85,838 (80,103, 91,737)	115,948 (107,845, 124,165)
0.01	Costs: Medical	56,766 (6,276, 181,828)	56,950 (6,441, 175,610)	58,410 (6,774, 177,613)
0.01	Costs: total	73,861 (23,476, 198,972)	142,787 (92,250, 259,529)	174,357 (121,736, 294,721)
0.03	QALYs	6.57 (5.98, 7.17)	7.10 (6.51, 7.69)	7.52 (6.90, 8.18)

Discussion

The purpose of R Markdown is to make reporting from R automated and accessible for modelers and intended audience. However, it is always important to leave room for thoughtful interpretation and messaging. It is recommended that you note throughout your document where further results interpretation is needed upon document finalization. Useful formatting tips include:

- Single asterisks italicize text *like this*.
- Double asterisks embolden text **like this**.
- Use text colour to mark sections

To assist with discussions and interpretations, the values that have been calculated or used in the R chunks can be directly imported into the text, for example, the top row of Table 1 shows that is the SOC regimen 1 outcome value is 7.85 (7.05, 8.63).