APPLIED COST-EFFECTIVENESS MODELING WITH R

SMDM
October 6, 2021

Devin Incerti
Jeroen Jansen
Learning objectives

- Understand how R can be used to perform model-based cost-effectiveness analysis with existing packages;
- Develop own models in R by modifying existing code for commonly used model types;
- Understand how using R can improve reproducibility and transparency of model-based cost-effectiveness analysis
Agenda

- Introduction
- Basic model taxonomy
- Simple Markov cohort model
- Semi-Markov multi-state model
- Partitioned-survival model
- Cost-effectiveness analysis
- Summary
Structure

- Presentation
- Exercises using R
Online tutorial

https://hesim-dev.github.io/rcea/

rcea

This is the repository for the rcea package, which contains a range of models for health economic evaluations.

The course materials are available at https://hesim-dev.github.io/rcea.

Installation and setup

All required R packages and course materials can be installed with the following steps.

1. Open an R session. We recommend using RStudio.

2. Install the rcea package from GitHub, which will also install all other required packages.

   ```
   # install.packages("devtools") # You must install the "devtools" R package first.
   devtools::install_github("hesim-dev/rcea")
   ```

3. Create a new project in your desired directory.

   ```
   # Create a project named "rcea-exercises" within a directory named "Projects"
   usethis::create_project("~/Projects/rcea-exercises")
   ```

4. Add the course materials (R scripts for the tutorials) to your new project.

   ```
   rcea::use_rcea("~/Projects/rcea-exercises")
   ```
R scripts for exercises
## RStudio Cloud

### Your Workspace / RCEA

---

1. ## Overview

   ```
   library("rce")
   library("knitr")
   library("kableExtra")
   library("magrittr")
   library("tibble")
   library("rcr")

   ## Knitr R-packages

   # Knitr transition-probabilities
   p_hd <- 0.002 # constant probability of dying when Healthy (all-cause mortality)
   p_hs1 <- 0.15 # probability of becoming Sick when Healthy
   p_slh <- 0.5 # probability of becoming Healthy when Sick
   p_sls2 <- 0.105 # probability of becoming Sicker when Sick
   p_sld <- 0.006 # constant probability of dying when Sick
   p_s2d <- 0.02 # constant probability of dying when Sicker

   ## Knitr transition-probability-complements
   p_Hn <- 1 - p_hs1 - p_hd
   p_sls1 <- 1 - p_sls2 - p_sld
   ```

---
Introduction
Criteria that economic models should strive to meet

- **Clinical realism**
  - A model should reflect the state of evidence, the current understanding of the disease, and be accepted by clinical experts.

- **Quantifying decision uncertainty**
  - A model should be capable of quantifying decision uncertainty and informing prioritization of future research.

- **Transparency and reproducibility**
  - Resources should exist so that a model can be completely understood, reproduced, and pressure tested.

- **Reusability and adaptability**
  - It should be possible to easily update a model to reflect new clinical evidence or adapt it for a new market, indication, or intervention.
Structural uncertainty

![Bar Chart]

- Incremental net monetary benefit
- Model structure

- Y-axis: $\text{Incremental net monetary benefit}$
- X-axis: $\text{Model structure}$

Values range from $-$100,000 to $150,000.
Plugging in model input parameter estimates
What do we mean with model transparency?

- Concept, math
- Face validity
- Implementation/programming
- Open-source, open-access

- Familiarity with software?

Benefits
Risks
Costs
Modeling in Excel

- Excel has been dominant software platform used by HE modelers, especially for HTA submissions

- Reasons are not surprising
  - Practically everyone with a computer has access to Excel
  - Does not require that you learn a new programming language

- Many consider its “transparency” to be an attribute

- With models in Excel, you can follow calculations that are being performed in every single cell of every single worksheet
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><strong>Variables and scenarios</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><strong>Scenario variables</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Where is Waldo
Time to change?

ASOK, ACCORDING TO MY SPREADSHEET, YOU HAVE BEEN DOING A TERRIBLE JOB.

PERHAPS YOUR SPREADSHEET IS POORLY CONCEIVED AND DOES NOT CAPTURE THE COMPLEXITY OF THE REAL WORLD.

AND LET’S NOT FORGET THE NEAR CERTAINTY THAT YOUR FORMULAE ARE POINTING TO THE WRONG CELLS.

NUMBERS DON’T LIE.
Alternative

BCEA  heemod  hesim  ...
What is R

- Statistical programing language and environment for statistical computing
- Free to use (open source, user developed packages that are transparent)
- Very good for data management, statistical analysis, and visualization
- Scripts contain all steps to perform an analysis
- CEA models can be coded from ‘scratch’ using base R or via convenient and improving packages
Basic model taxonomy
Health states describing course of disease over time

\[ S \rightarrow P \rightarrow D \]
Markov model

Clock forward

transition rates depend only on time in model
Semi-Markov model

Clock reset

transition rates depend on time in model
some transitions depend on time in an intermediate health state
Partitioned survival model

### Summary of model types

<table>
<thead>
<tr>
<th>State transition models</th>
<th>discrete time</th>
<th>continuous time</th>
</tr>
</thead>
</table>
| Markov ("Clock forward") | cohort discrete time state transition models (cDTSTM)  
\* time-homogeneous Markov models  
\* time-inhomogeneous Markov models | cohort continuous time state transition models |
| Individual-level | individual-level discrete time state transition models (iDTSTM) | individual-level continuous time state transition models (iCTSTM) |
| Semi-Markov ("Clock reset")* | Individual-level | iDTSTM | iCTSTM |
| Partitioned survival model | Cohort | | ✓ |

*`tunnel states can be used in a cohort model to approximate a semi-Markov process`
Simple Markov cohort model
Model

Healthy → Sick → Sicker → Death → Healthy
Model

- Annual transition probabilities with SOC

<table>
<thead>
<tr>
<th></th>
<th>Healthy</th>
<th>Sick</th>
<th>Sicker</th>
<th>Death</th>
</tr>
</thead>
<tbody>
<tr>
<td>Healthy</td>
<td>0.848</td>
<td>0.15</td>
<td>0</td>
<td>0.002</td>
</tr>
<tr>
<td>Sick</td>
<td>0.5</td>
<td>0.389</td>
<td>0.105</td>
<td>0.006</td>
</tr>
<tr>
<td>Sicker</td>
<td>0</td>
<td>0</td>
<td>0.98</td>
<td>0.02</td>
</tr>
<tr>
<td>Death</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.00</td>
</tr>
</tbody>
</table>

- Relative risk of progression to a worse health state with new intervention is 0.8

- Drug costs

<table>
<thead>
<tr>
<th></th>
<th>New</th>
<th>SOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drug costs</td>
<td>12000</td>
<td>2000</td>
</tr>
</tbody>
</table>

- Other annual costs and utility

<table>
<thead>
<tr>
<th></th>
<th>Healthy</th>
<th>Sick</th>
<th>Sicker</th>
<th>Death</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct medical</td>
<td>2000</td>
<td>4000</td>
<td>15000</td>
<td>0</td>
</tr>
<tr>
<td>Utility</td>
<td>1</td>
<td>0.75</td>
<td>0.5</td>
<td>0</td>
</tr>
</tbody>
</table>

- Annual discount rates of 3% for costs and 3% for QALYs
<table>
<thead>
<tr>
<th>Cycle</th>
<th>Healthy QALYs</th>
<th>Sick QALYs</th>
<th>Sicker QALYs</th>
<th>Death QALYs</th>
<th>Discounted medical costs</th>
<th>Discounted treatment costs</th>
<th>Discounted total costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>1.00000</td>
<td>2000.000</td>
<td>2000.000</td>
</tr>
<tr>
<td>1</td>
<td>0.84800</td>
<td>0.15000</td>
<td>0.00000</td>
<td>0.00200</td>
<td>0.93252</td>
<td>2229.126</td>
<td>1937.864</td>
</tr>
<tr>
<td>2</td>
<td>0.79410</td>
<td>0.18555</td>
<td>0.01575</td>
<td>0.00460</td>
<td>0.88712</td>
<td>2419.321</td>
<td>1876.527</td>
</tr>
<tr>
<td>3</td>
<td>0.76618</td>
<td>0.19129</td>
<td>0.03492</td>
<td>0.00761</td>
<td>0.84843</td>
<td>2581.884</td>
<td>1816.350</td>
</tr>
<tr>
<td>4</td>
<td>0.74536</td>
<td>0.18934</td>
<td>0.05431</td>
<td>0.01099</td>
<td>0.81254</td>
<td>2721.140</td>
<td>1757.443</td>
</tr>
<tr>
<td>5</td>
<td>0.72674</td>
<td>0.18546</td>
<td>0.07310</td>
<td>0.01470</td>
<td>0.77840</td>
<td>2839.541</td>
<td>1699.850</td>
</tr>
<tr>
<td>6</td>
<td>0.70900</td>
<td>0.18115</td>
<td>0.09111</td>
<td>0.01873</td>
<td>0.74572</td>
<td>2938.972</td>
<td>1643.593</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Healthy Utility</th>
<th>Sick Utility</th>
<th>Sicker Utility</th>
<th>Death Utility</th>
<th>Discounted medical costs</th>
<th>Discounted treatment costs</th>
<th>Discounted total costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>0.11578</td>
<td>0.02960</td>
<td>0.29714</td>
<td>0.55748</td>
<td>0.02693</td>
<td>451.752</td>
<td>83.173</td>
</tr>
<tr>
<td>81</td>
<td>0.11298</td>
<td>0.02888</td>
<td>0.29430</td>
<td>0.56384</td>
<td>0.02571</td>
<td>433.941</td>
<td>79.591</td>
</tr>
<tr>
<td>82</td>
<td>0.11025</td>
<td>0.02818</td>
<td>0.29145</td>
<td>0.57012</td>
<td>0.02455</td>
<td>416.778</td>
<td>76.159</td>
</tr>
<tr>
<td>83</td>
<td>0.10758</td>
<td>0.02750</td>
<td>0.28858</td>
<td>0.57634</td>
<td>0.02344</td>
<td>400.244</td>
<td>72.871</td>
</tr>
<tr>
<td>84</td>
<td>0.10498</td>
<td>0.02683</td>
<td>0.28570</td>
<td>0.58249</td>
<td>0.02237</td>
<td>384.317</td>
<td>69.722</td>
</tr>
<tr>
<td>85</td>
<td>0.10244</td>
<td>0.02619</td>
<td>0.28280</td>
<td>0.58858</td>
<td>0.02136</td>
<td>368.979</td>
<td>66.704</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sum</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Healthy Drug costs</th>
<th>Sick Drug costs</th>
<th>Sicker Drug costs</th>
<th>Death Drug costs</th>
<th>Discounted medical costs</th>
<th>Discounted treatment costs</th>
<th>Discounted total costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>0.11578</td>
<td>0.02960</td>
<td>0.29714</td>
<td>0.55748</td>
<td>0.02693</td>
<td>451.752</td>
<td>83.173</td>
</tr>
<tr>
<td>81</td>
<td>0.11298</td>
<td>0.02888</td>
<td>0.29430</td>
<td>0.56384</td>
<td>0.02571</td>
<td>433.941</td>
<td>79.591</td>
</tr>
<tr>
<td>82</td>
<td>0.11025</td>
<td>0.02818</td>
<td>0.29145</td>
<td>0.57012</td>
<td>0.02455</td>
<td>416.778</td>
<td>76.159</td>
</tr>
<tr>
<td>83</td>
<td>0.10758</td>
<td>0.02750</td>
<td>0.28858</td>
<td>0.57634</td>
<td>0.02344</td>
<td>400.244</td>
<td>72.871</td>
</tr>
<tr>
<td>84</td>
<td>0.10498</td>
<td>0.02683</td>
<td>0.28570</td>
<td>0.58249</td>
<td>0.02237</td>
<td>384.317</td>
<td>69.722</td>
</tr>
<tr>
<td>85</td>
<td>0.10244</td>
<td>0.02619</td>
<td>0.28280</td>
<td>0.58858</td>
<td>0.02136</td>
<td>368.979</td>
<td>66.704</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sum</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

SOC
Steps

- Define transition matrix SOC
- Define transition matrix New treatment
- Define utility and cost values by health state
- Calculate health state probabilities over time
- Calculate expected QALYs and costs
- Cost-effectiveness analysis
Define transition matrix with standard of care

```r
# constant probability of dying when Healthy (all-cause mortality)
p_hd <- 0.002

# probability of becoming Sick when Healthy
p_hs1 <- 0.15

# probability of becoming Healthy when Sick
p_s1h <- 0.5

# probability of becoming Sicker when Sick
p_s1s2 <- 0.105

# constant probability of dying when Sick
p_s1d <- 0.006

# constant probability of dying when Sicker
p_s2d <- 0.02

# transition matrix
p_soc <- matrix(
  c(p_hh, p_hs1, 0, p_hd,
    p_s1h, p_s1s1, p_s1s2, p_s1d,
    0, 0, p_s2s2, p_s2d,
    0, 0, 0, 1),
  byrow = TRUE,
  nrow = 4, ncol = 4)

state_names <- c("H", "S1", "S2", "D")
colnames(p_soc) <- rownames(p_soc) <- state_names

print(p_soc)
```

```
     H     S1     S2     D
  H 0.848 0.150 0.000 0.002
  S1 0.500 0.389 0.105 0.006
  S2 0.000 0.000 0.980 0.020
  D 0.000 0.000 0.000 1.000
```
Relative risk and transition matrix with New treatment

```r
apply_rr <- function(p, rr = 0.8){
  p[H, S1] <- p[H, S1] * rr
  p[H, D]  <- p[H, D] * rr
  p[H, H]  <- 1 - sum(p[H, -1])

  p[S1, S2] <- p[S1, S2] * rr
  p[S1, D]  <- p[S1, D] * rr
  p[S1, S1] <- 1 - sum(p[S1, -2])

  p[S2, D]  <- p[S2, D] * rr
  p[S2, S2] <- 1 - sum(p[S2, -3])

  return(p)
}
```

```r
p_new <- apply_rr(p_soc, rr = 0.8)
```

<table>
<thead>
<tr>
<th></th>
<th>H</th>
<th>S1</th>
<th>S2</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>0.8784</td>
<td>0.1200</td>
<td>0.000</td>
<td>0.0016</td>
</tr>
<tr>
<td>S1</td>
<td>0.5000</td>
<td>0.4112</td>
<td>0.084</td>
<td>0.0048</td>
</tr>
<tr>
<td>S2</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.984</td>
<td>0.0160</td>
</tr>
<tr>
<td>D</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>
Utility and costs

```r
utility <- c(1, .75, .5, 0)
costs_medical <- c(2000, 4000, 15000, 0)
costs_treat_soc <- c(rep(2000, 3), 0)
costs_treat_new <- c(rep(12000, 3), 0)
```

```r
> utility
[1] 1.000 0.75 0.500 0.000

> costs_medical
[1] 2000 4000 15000 0

> costs_treat_soc

> costs_treat_new
[1] 12000 12000 12000 0
```
Simulation – health state probabilities

```
x_init <- c(1, 0, 0, 0)
x_init %*% p_soc

H  S1  S2  D
[1,] 0.848 0.15 0 0.002
```

Matrix multiplication

```
x_init %*% p_soc %*% p_soc

H  S1  S2  D
[1,] 0.794104 0.18555 0.01575 0.004596
```
Simulation – health state probabilities with a function

x0  The state vector at model cycle 0 (i.e., the initial state vector)
p   The transition probability matrix
n_cycles  The number of model cycles. (Default is 85)

```r
sim_markov_chain <- function(x0, p, n_cycles = 85){
  x <- matrix(NA, ncol = length(x0), nrow = n_cycles)  # Initialize Markov trace
  x <- rbind(x0, x)                                   # Markov trace at cycle 0 is initial state vector
  colnames(x) <- colnames(p)                          # Columns are the health states
  rownames(x) <- 0:n_cycles                           # Rows are the model cycles

  for (t in 1:n_cycles){
    x[t + 1, ] <- x[t, ] %*% p                        # Simulating state vectors at each cycle with for loop
  }

  return(x)
}
```

x0  The state vector at model cycle 0 (i.e., the initial state vector)
p   The transition probability matrix
n_cycles  The number of model cycles. (Default is 85)
Simulation – health state probabilities with a function

\[
x_{soc} \leftarrow \text{sim_markov_chain}(x_{init}, p_{soc})
\]

<table>
<thead>
<tr>
<th></th>
<th>H</th>
<th>S1</th>
<th>S2</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>1</td>
<td>0.8480000</td>
<td>0.15000000</td>
<td>0.00000000</td>
<td>0.00200000</td>
</tr>
<tr>
<td>2</td>
<td>0.7941040</td>
<td>0.18555000</td>
<td>0.01575000</td>
<td>0.00459600</td>
</tr>
<tr>
<td>3</td>
<td>0.7661752</td>
<td>0.19129775</td>
<td>0.03491775</td>
<td>0.00761250</td>
</tr>
<tr>
<td>4</td>
<td>0.7453638</td>
<td>0.18933986</td>
<td>0.05430532</td>
<td>0.00109909</td>
</tr>
<tr>
<td>5</td>
<td>0.7267385</td>
<td>0.18545778</td>
<td>0.07309990</td>
<td>0.00147038</td>
</tr>
<tr>
<td>6</td>
<td>0.7090031</td>
<td>0.18115385</td>
<td>0.09111097</td>
<td>0.00387321</td>
</tr>
<tr>
<td>7</td>
<td>0.6918116</td>
<td>0.17681931</td>
<td>0.10830990</td>
<td>0.00722303</td>
</tr>
<tr>
<td>8</td>
<td>0.6750659</td>
<td>0.17255445</td>
<td>0.12470973</td>
<td>0.01066241</td>
</tr>
<tr>
<td>9</td>
<td>0.6587331</td>
<td>0.16838356</td>
<td>0.14033376</td>
<td>0.01423685</td>
</tr>
<tr>
<td>---</td>
<td>---------</td>
<td>------------</td>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>79</td>
<td>0.2230518</td>
<td>0.04678750</td>
<td>0.31880122</td>
<td>0.41135950</td>
</tr>
<tr>
<td>80</td>
<td>0.2193224</td>
<td>0.04600523</td>
<td>0.31763055</td>
<td>0.41704179</td>
</tr>
<tr>
<td>81</td>
<td>0.2156554</td>
<td>0.04523604</td>
<td>0.31641290</td>
<td>0.42269516</td>
</tr>
<tr>
<td>82</td>
<td>0.2120498</td>
<td>0.04447971</td>
<td>0.31515012</td>
<td>0.42832040</td>
</tr>
<tr>
<td>83</td>
<td>0.2085044</td>
<td>0.04373603</td>
<td>0.31384402</td>
<td>0.43391559</td>
</tr>
<tr>
<td>84</td>
<td>0.2050182</td>
<td>0.04300478</td>
<td>0.31249634</td>
<td>0.43948063</td>
</tr>
<tr>
<td>85</td>
<td>0.2015904</td>
<td>0.04228575</td>
<td>0.31110880</td>
<td>0.44501528</td>
</tr>
</tbody>
</table>

\[
x_{new} \leftarrow \text{sim_markov_chain}(x_{init}, p_{new})
\]

<table>
<thead>
<tr>
<th></th>
<th>H</th>
<th>S1</th>
<th>S2</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>1</td>
<td>0.8784000</td>
<td>0.12000000</td>
<td>0.00000000</td>
<td>0.00160000</td>
</tr>
<tr>
<td>2</td>
<td>0.8315866</td>
<td>0.15475200</td>
<td>0.01008000</td>
<td>0.00358144</td>
</tr>
<tr>
<td>3</td>
<td>0.8078416</td>
<td>0.16342441</td>
<td>0.02291789</td>
<td>0.00825973</td>
</tr>
<tr>
<td>4</td>
<td>0.7913203</td>
<td>0.16411111</td>
<td>0.03627885</td>
<td>0.01089419</td>
</tr>
<tr>
<td>5</td>
<td>0.7771663</td>
<td>0.16245326</td>
<td>0.04948624</td>
<td>0.01370921</td>
</tr>
<tr>
<td>6</td>
<td>0.7638895</td>
<td>0.16060747</td>
<td>0.06234054</td>
<td>0.01697175</td>
</tr>
<tr>
<td>7</td>
<td>0.7510309</td>
<td>0.15743837</td>
<td>0.07478819</td>
<td>0.01985135</td>
</tr>
<tr>
<td>8</td>
<td>0.7384744</td>
<td>0.15488101</td>
<td>0.08682021</td>
<td>0.02316542</td>
</tr>
<tr>
<td>9</td>
<td>0.7260927</td>
<td>0.15230076</td>
<td>0.09844109</td>
<td>0.02615425</td>
</tr>
<tr>
<td>---</td>
<td>---------</td>
<td>------------</td>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>79</td>
<td>0.1186504</td>
<td>0.03032935</td>
<td>0.29995426</td>
<td>0.55106599</td>
</tr>
<tr>
<td>80</td>
<td>0.1157802</td>
<td>0.02959568</td>
<td>0.29713976</td>
<td>0.55748435</td>
</tr>
<tr>
<td>81</td>
<td>0.1129795</td>
<td>0.02887975</td>
<td>0.29430451</td>
<td>0.56383628</td>
</tr>
<tr>
<td>82</td>
<td>0.1102465</td>
<td>0.02818114</td>
<td>0.29145079</td>
<td>0.56912611</td>
</tr>
<tr>
<td>83</td>
<td>0.1075796</td>
<td>0.02749943</td>
<td>0.28858079</td>
<td>0.57634020</td>
</tr>
<tr>
<td>84</td>
<td>0.1049772</td>
<td>0.02683421</td>
<td>0.28569662</td>
<td>0.58249197</td>
</tr>
<tr>
<td>85</td>
<td>0.1024378</td>
<td>0.02618509</td>
<td>0.28280028</td>
<td>0.58857687</td>
</tr>
</tbody>
</table>
Computing a present value with a function

```
pv <- function(z, dr, t) {
  z/(1 + dr)^t
}
```

`z` A numeric quantity
`dr` Discount rate
`t` Vector of times to compute the present value

```
pv(1000, dr = .03, t = 0:4)
```

```
[1] 1000.0000  970.8738  942.5959  915.1417  888.4870
```
QALYs after 1st cycle

```r
x_soc[2, ] # State occupancy probabilities after 1st cycle

    H  S1  S2  D
  0.848 0.150 0.000 0.002

sum(x_soc[2, 1:3]) # Expected life-years after 1st cycle

[1] 0.998

sum(x_soc[2, ] * utility) # Expected utility after 1st cycle

[1] 0.9605

sum(pv(x_soc[2, ] * utility, .03, 1)) # Expected discounted utility after 1st cycle

[1] 0.9325
```
Discounted QALYs for each cycle

```r
compute_qalys <- function(x, utility, dr = .03){
  n_cycles <- nrow(x) - 1
  pv(x %**% utility, dr, 0:n_cycles)
}

dqalys_soc <- compute_qalys(x_soc, utility = utility)
dqalys_new <- compute_qalys(x_new, utility = utility)

head(dqalys_soc)
head(dqalys_new)
```

```
[,1]      [,1]
 0 1.0000000 0 1.0000000
 1 0.9325243 1 0.9401942
 2 0.8871161 2 0.8980022
 3 0.8484324 3 0.8619435
 4 0.8125404 4 0.8285724
 5 0.7784024 5 0.7968343
```
Discounted cost for each cycle

```r
compute_costs <- function(x, costs_medical, costs_treat, dr = .03){
  n_cycles <- nrow(x) - 1
  costs <- cbind(pv(x %*% costs_medical, dr, 0:n_cycles),
                 pv(x %*% costs_treat, dr, 0:n_cycles))
  colnames(costs) <- c("medical", "treatment")
  return(costs)
}

dcosts_soc <- compute_costs(x_soc, costs_medical, costs_treat_soc)
dcosts_new <- compute_costs(x_new, costs_medical, costs_treat_new)

head(dcosts_soc)
head(dcosts_new)
```

<table>
<thead>
<tr>
<th>medical</th>
<th>treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 2000.000</td>
<td>2000.000</td>
</tr>
<tr>
<td>1 2229.126</td>
<td>1937.864</td>
</tr>
<tr>
<td>2 2419.321</td>
<td>1876.527</td>
</tr>
<tr>
<td>3 2581.884</td>
<td>1816.350</td>
</tr>
<tr>
<td>4 2721.140</td>
<td>1757.443</td>
</tr>
<tr>
<td>5 2839.541</td>
<td>1699.850</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>medical</th>
<th>treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 2000.000</td>
<td>12000.00</td>
</tr>
<tr>
<td>1 2171.650</td>
<td>11631.84</td>
</tr>
<tr>
<td>2 2293.695</td>
<td>11270.64</td>
</tr>
<tr>
<td>3 2391.402</td>
<td>10917.83</td>
</tr>
<tr>
<td>4 2473.004</td>
<td>10573.78</td>
</tr>
<tr>
<td>5 2541.624</td>
<td>10238.54</td>
</tr>
</tbody>
</table>
Cost-effectiveness

\[
\frac{(\text{sum}(\text{dcosts\_new}[-1,\ ])) - \text{sum}(\text{dcosts\_soc}[-1,\ ]))}{(\text{sum}(\text{dqalys\_new}[-1,\ ])) - \text{sum}(\text{dqalys\_soc}[-1,\ ]))}
\]

\[
[1] \ 122946.8
\]

```r
format\_costs \leftarrow \text{function}(x) \ \text{formatC}(x, \text{format} = "d", \text{big.mark} = ",")
format\_qalys \leftarrow \text{function}(x) \ \text{formatC}(x, \text{format} = "f", \text{digits} = 2)

\text{make\_icer\_tbl} \leftarrow \text{function}(\text{costs0}, \text{costs1}, \text{qalys0}, \text{qalys1}){
    \# \text{Computations}
    \text{total\_costs0} \leftarrow \text{sum}(\text{costs0})
    \text{total\_costs1} \leftarrow \text{sum}(\text{costs1})
    \text{total\_qalys0} \leftarrow \text{sum}(\text{qalys0})
    \text{total\_qalys1} \leftarrow \text{sum}(\text{qalys1})
    \text{incr\_total\_costs} \leftarrow \text{total\_costs1} - \text{total\_costs0}
    \text{inc\_total\_qalys} \leftarrow \text{total\_qalys1} - \text{total\_qalys0}
    \text{icer} \leftarrow \frac{\text{incr\_total\_costs}}{\text{inc\_total\_qalys}}

    \# \text{Make table}
    \text{tibble}(\text{\`Costs\'} = \text{c}(\text{total\_costs0}, \text{total\_costs1}) \text{\%\%}
    \text{\`Strategy\'} = \text{c}(\"SOC", \"New\"),
    \text{\`format\_costs\'),
    \text{\`QALYS\'} = \text{c}(\text{total\_qalys0}, \text{total\_qalys1}) \text{\%\%}
    \text{\`format\_qalys\'),
    \text{\`Incremental\_costs\'} = \text{c}(\"--", \text{incr\_total\_costs} \text{\%}\% \text{\`format\_costs\'),
    \text{\`Incremental\_QALYS\'} = \text{c}(\"--", \text{inc\_total\_qalys} \text{\%}\% \text{\`format\_qalys\'),
    \text{\`ICER\'} = \text{c}(\"--", \text{icer} \text{\%}\% \text{\`format\_costs\')}) \text{\%\%}
    \text{\kable}\text{\%\%
    \text{\kable\_styling}\text{\%\%
    \text{footnote}(\text{general} = \"\text{Costs and QALYS are discounted at 3\% per annum.\"}, \text{footnote\_as\_chunk} = \text{TRUE})
}
```
## Cost-effectiveness

```make_icer_tbl(costs0 = dcosts_soc[-1, ], costs1 = dcosts_new[-1, ], qalys0 = dqalys_soc[-1, ], qalys1 = dqalys_new[-1, ])```

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Costs</th>
<th>QALYs</th>
<th>Incremental costs</th>
<th>Incremental QALYs</th>
<th>ICER</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOC</td>
<td>204,123</td>
<td>21.08</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>New</td>
<td>464,390</td>
<td>23.19</td>
<td>260,266</td>
<td>2.12</td>
<td>122,946</td>
</tr>
</tbody>
</table>

*Note: Costs and QALYs are discounted at 3% per annum.*
Steps

- Define transition matrix SOC
- Define transition matrix New treatment $\text{apply\_rr}(p_{soc}, \text{rr})$
- Define utility and cost values by health state
- Calculate health state probabilities over time $\text{sim\_markov\_chain}(x0, p, n\_cycles)$
- Calculate expected QALYs and costs
  - $\text{pv}(z, \text{dr}, t)$
  - $\text{compute\_qalys}(x, \text{utility}, \text{dr})$
  - $\text{compute\_costs}(x, \text{costs\_medical}, \text{costs\_treat}, \text{dr})$
- Cost-effectiveness analysis $\text{make\_icer\_tbl}(\text{costs0}, \text{costs1}, \text{qalys0}, \text{qalys1})$
Complete R script

```r
# -- Overview
library("rceoa")
library("knitr")
library("kableExtra")
library("magrittr")
library("tibble")

# -- Model parameters ---------------------------------------------
# @knitr transition-probabilities
p_hd <- 0.002 # constant probability of dying when Healthy (all-cause mortality)
p_hs1 <- 0.15 # probability of becoming Sick when Healthy
p_slh <- 0.5 # probability of becoming Healthy when Sick
p_sls2 <- 0.105 # probability of becoming Sicker when Sick
p_sid <- 0.006 # constant probability of dying when Sick
p_s2d <- 0.02 # constant probability of dying when Sicker

# @knitr transition-probability-complements
p_hh <- 1 - p_hs1 - p_hd
p_sls1 <- 1 - p_slh - p_sls2 - p_sid
p_s2s2 <- 1 - p_s2d

# @knitr Gmatrix
p_soc <- matrix(c(p_hh, p_hs1, 0, p_hd, 
p_slh, p_sls1, p_sls2, p_sid, 
0, 0, p_s2s2, p_s2d, 
0, 0, 0, 1),
byrow = TRUE,
nrow = 4, ncol = 4)

state_names <- c("H", "S1", "S2", "D")
colnames(p_soc) <- rownames(p_soc) <- state_names
print(p_soc)

# @knitr apply_rr
apply_rr <- function(p, rr = .8){
  p["H", "S1"] <- p["H", "S1"] * rr
  p["H", "S2"] <- p["H", "S2"] * rr
  p["H", "D"] <- p["H", "D"] * rr
  p["H", "H"] <- 1 - sum(p["H", -2])
  p["S1", "S2"] <- p["S1", "S2"] * rr
  p["S1", "D"] <- p["S1", "D"] * rr
  p["S1", "S1"] <- 1 - sum(p["S1", -2])
}
Simple Markov Cohort Model

2021-07-26

Source: vignettes/01-markov-cohort.Rmd

https://hesim-dev.github.io/rcea/articles/01-markov-cohort.html

referred to as a Markov cohort model. In this tutorial we demonstrate implementation with R of the simplest of models, a time-homogeneous model with transition probabilities that are constant over time. The entire analysis can be run using Base R (i.e., without installing any packages). However, we will use the following packages to create a nice looking cost-effectiveness table.

```r
library("rcea")
library("knitr")
library("kableExtra")
library("magrittr")
library("tibble")
```

As an example, we will consider the 4-state sick-sicker model that has been described in more detail by Alarid-Escudero et al. The model will be used to compare two treatment strategies, a “New” treatment and the existing “standard of care (SOC)”. The model consists 4 health states. Ordered from worst to best: they are: Healthy (H), Sick (S1), Sicker (S2), and Death (D). Possible transitions from each state are displayed in the figure below.
Exercise 1: Simple Markov Cohort model

- Modify R-script “01-markov-cohort-R”
  - Change relative risk
  - Change drug costs
  - Change utilities
  - Change follow-up time (i.e. number of cycles)

```
sim_markov_chain(x0, p, n_cycles)
```

- Run modified script
Simple Markov cohort model – Incorporating probabilistic sensitivity analysis
Probabilistic sensitivity analysis

- The standard methodology for quantifying the impact of parameter uncertainty is probabilistic sensitivity analysis (PSA)
- Propagating uncertainty in the input parameters throughout the model by randomly sampling sets of input values from suitable probability distributions
- Probability distributions are determined according to the distributional properties of the statistical estimates, which, in turn, depend on the statistical techniques used and the distributions of the underlying data
- R is a natural programming language for performing PSA
- Random samples can be drawn from almost any probability distribution
We assume that summary level data is available on transitions from the Healthy state (n = 1000), Sick state (n = 1000), and Sicker state (n = 800).

The transitions from each state to the other 4 states can be modeled using a Dirichlet distribution.

```r
transitions_soc <- matrix(
  c(848, 150, 0, 2,
    500, 389, 105, 6,
    0, 0, 784, 16,
    0, 0, 0, 23),
  nrow = 4, byrow = TRUE)

state_names <- c("H", "S1", "S2", "D")
colnames(transitions_soc) <- rownames(transitions_soc) <- tolower(state_names)
```

The transitions from each state to the other 4 states can be modeled using a Dirichlet distribution.
Combining all model parameters

```r
params <- list(
  alpha_soc = transitions_soc,

  lrr_mean = log(.8),
  lrr_lower = log(.71),
  lrr_upper = log(.9),

  c_medical = c(H = 2000, S1 = 4000, S2 = 15000, D = 0),
  c_soc = 2000,
  c_new = 12000,

  u_mean = c(H = 1, S1 = .75, S2 = 0.5, D = 0),
  u_se = c(H = 0, S1 = 0.03, S2 = 0.05, D = 0.0)
)
```
Simulation

- The simulation proceeds by
  1. randomly sampling the parameters from the probability distributions specified
  2. running the Markov model for each draw of the parameters

- The result is a draw from the probability distribution of each of the model outputs of interest (i.e., state probabilities, QALYs, and costs).
Sampling parameters

- While Base R can be used to draw samples of parameters, the functions `hesim::define_rng()` and `hesim::eval_rng()` simplify this process.

- Any random number generation function can be used inside the `define_rng()` block

```r
rng_def <- define_rng({
  lrr_se <- (lrr_upper - lrr_lower)/(2 * qnorm(.975))
  list(
    # Parameters to return
    p_soc = dirichlet_rng(alpha_soc),
    rr_new = lognormal_rng(lrr_mean, lrr_se),
    c_medical = gamma_rng(mean = c_medical, sd = c_medical),
    c_soc = c_soc,
    c_new = c_new,
    u = beta_rng(mean = u_mean, sd = u_se)
  ), n = 1000)
})

params_rng <- eval_rng(rng_def, params = params)
```
### Sampling parameters

```r
names(params_rng)
```

```r
[1] "p_soc"    "rr_new"    "c_medical" "c_soc"    "c_new"  "u"
```

```r
head(as.matrix(params_rng$p_soc))
```

```
       h_h  h_s1  h_s2    h_d    s1_h    s1_s1    s1_s2    s1_d   s2_h  s2_s1
[1,] 0.8686001 0.1293872    0 0.002012649 0.3847703 0.10868135 0.006244105    0     0     0
[2,] 0.8474698 0.1499868    0 0.002543368 0.3964675 0.08375492 0.005155768    0     0     0
[3,] 0.8559237 0.1428611    0 0.001215126 0.3775261 0.10868021 0.002036291    0     0     0
[4,] 0.8550586 0.1429657    0 0.001975648 0.3775261 0.10868021 0.002036291    0     0     0
[5,] 0.8678962 0.1304462    0 0.001657694 0.3815376 0.09725638 0.004771722    0     0     0
[6,] 0.8530231 0.1459388    0 0.001038023 0.4991514 0.3750200 0.11846736 0.007361245    0     0
```

```
s2_s2 s2_d    d_h    d_s1    d_s2    d_d
[1,] 0.9786377 0.02136235    0     0     0     1
[2,] 0.9871702 0.01282980    0     0     0     1
[3,] 0.9786291 0.02137091    0     0     0     1
[4,] 0.9809938 0.01900624    0     0     0     1
[5,] 0.9793886 0.02061141    0     0     0     1
[6,] 0.9750211 0.02497889    0     0     0     1
```

Simulating the Markov model

- One way that a Markov simulation can be generalized is to store “input data” in an object, i.e. data frame.

- Input data might consist of
  - treatment strategies
  - patients and subgroups
    - For instance, if we were simulating different subgroups we might store the age and sex associated with the subgroup which could, in turn, be used as covariates in a statistical model.

```r
data <- data.frame(
  strategy = c("New", "SOC")
)
```

<table>
<thead>
<tr>
<th>strategy</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>1</td>
</tr>
<tr>
<td>SOC</td>
<td>2</td>
</tr>
</tbody>
</table>
Simulating the Markov model – Create the function

- Set up a `sim_model()` function that runs the entire simulation.
  - Comprised of three smaller functions:
    - `sim_stateprobs()`
    - `compute_qalys()`
    - `compute_costs()`
Simulating the Markov model – Create the function

```r
sim_stateprobs <- function(p0, rr, strategy, n_cycles){
  rr <- ifelse(strategy == "New", rr, 1)
  p <- tpmatrix(
    C, p0$h_s1 * rr, p0$h_s2 * rr, p0$h_d * rr,
    p0$s1_h, C, p0$s1_s2 * rr, p0$s1_d * rr,
    p0$s2_h, p0$s2_s1, C, p0$s2_d * rr,
    0, 0, 0, 1
  )
  x <- sim_markov_chain(x0 = c(1, 0, 0, 0),
                        p = matrix(as.matrix(p), ncol = 4, byrow = TRUE),
                        n_cycles = n_cycles)
  return(x)
}
```

hesim::tpmatrix() makes it easy to define a transition probability matrix.

C denotes that a given element is the complement of all other elements in that row, ensuring that the probabilities sum to 1.

sim_markov_chain() is the function we created previously
Simulating the Markov model – Create the function

```r
# QALYs
compute_qalys <- function(x, utility, dr = .03){
  n_cycles <- nrow(x) - 1
  pv(x %*% utility, dr, 0:n_cycles)
}

# Costs
compute_costs <- function(x, costs_medical, costs_treat, dr = .03){
  n_cycles <- nrow(x) - 1
  costs_treat <- c(rep(costs_treat, 3), 0)
  costs <- cbind(
    pv(x %*% costs_medical, dr, 0:n_cycles),
    pv(x %*% costs_treat, dr, 0:n_cycles)
  )
  colnames(costs) <- c("dcost_med", "dcost_treat")
  return(costs)
}
```
```r
sim_model <- function(params_rng, data, n_cycles = 85, dr_qalys = .03, dr_costs = .03)
{
    # Initialize array of matrices
    n_samples <- attr(params_rng, "n")
    n_strategies <- nrow(data)
    out <- array(NA, dim = c(n_cycles + 1, 7, n_samples * n_strategies))
    dimnames(out) <- list(NULL, c("H", "S1", "S2", "D", "dqalys", "dcosts_med", "dcosts_treat"), NULL)

    # Run the simulation
    i <- 1
    for (s in 1:n_samples)
    {
        for (k in 1:n_strategies)
        {
            x <- sim_stateprobs(p0 = params_rng$p_soc[s, ],
                                 rr = params_rng$rr_new[s],
                                 strategy = data$strategy[k],
                                 n_cycles = n_cycles)
            dqalys <- compute_qalys(x, utility = unlist(params_rng$u[s]), dr = dr_qalys)
            dcosts <- compute_costs(x,
                                     costs_medical = unlist(params_rng$c_medical[s]),
                                     costs_treat = ifelse(data$strategy[k] == "SOC",
                                                          params_rng$c_soc,
                                                          params_rng$c_new),
                                     dr = dr_costs)
            out[, , i] <- cbind(x, dqalys, dcosts)
            i <- i + 1
        }  # End treatment strategy loop
    }  # End PSA loop

    # Store metadata and return
    attr(out, "n_samples") <- n_samples
    attr(out, "strategies") <- data$strategy
    return(out)
}
```

Simulates for each parameter sample and treatment strategy:
- state probabilities (with `sim_stateprobs()`)
- QALYs (with `compute_qalys()`)
- costs (with `compute_costs()`)

An array to store the output.
A series of matrices each with `n_cycles` rows and columns for each output.
There is one matrix for each parameter sample for the PSA and treatment strategy.

The number of parameter samples and the names of the treatment strategies are saved as attributes (i.e., metadata) to the array.
Simulating the Markov model

```r
sim_out <- sim_model(params_rng, data = data)
head(sim_out[, , 1])
```

<table>
<thead>
<tr>
<th></th>
<th>H</th>
<th>S1</th>
<th>S2</th>
<th>D</th>
<th>dqalys</th>
<th>dcosts_med</th>
<th>dcosts_treat</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>1.0000000</td>
<td>0.0000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>1.0000000</td>
<td>639.2051</td>
<td>12000.00</td>
</tr>
<tr>
<td>[2,]</td>
<td>0.8927749</td>
<td>0.1055827</td>
<td>0.000000000</td>
<td>0.001642364</td>
<td>0.9440327</td>
<td>968.8714</td>
<td>11631.35</td>
</tr>
<tr>
<td>[3,]</td>
<td>0.8498706</td>
<td>0.1371191</td>
<td>0.009363733</td>
<td>0.003646603</td>
<td>0.9027066</td>
<td>1072.4091</td>
<td>11269.90</td>
</tr>
<tr>
<td>[4,]</td>
<td>0.8273444</td>
<td>0.1453902</td>
<td>0.021361082</td>
<td>0.005904295</td>
<td>0.8667374</td>
<td>1105.0422</td>
<td>10916.86</td>
</tr>
<tr>
<td>[5,]</td>
<td>0.8113717</td>
<td>0.1463692</td>
<td>0.033882825</td>
<td>0.008376274</td>
<td>0.8332591</td>
<td>1114.3372</td>
<td>10572.54</td>
</tr>
<tr>
<td>[6,]</td>
<td>0.7976015</td>
<td>0.1450801</td>
<td>0.046273111</td>
<td>0.011045289</td>
<td>0.8013670</td>
<td>1114.9710</td>
<td>10236.97</td>
</tr>
</tbody>
</table>
### Reorganize output

```r
sim_out <- array_to_dt(sim_out)
head(sim_out)
```

Convert a 3D array (faster to store data) to a data.table so we can summarize outcomes for each parameter sample and treatment strategy very quickly.

<table>
<thead>
<tr>
<th>cycle</th>
<th>strategy</th>
<th>sample</th>
<th>H</th>
<th>S1</th>
<th>S2</th>
<th>D</th>
<th>dqalys</th>
<th>dcosts_med</th>
<th>dcosts_treat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>0</td>
<td>New</td>
<td>1</td>
<td>1.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>1.000000</td>
<td>639.2051</td>
<td>12000.00</td>
</tr>
<tr>
<td>2:</td>
<td>1</td>
<td>New</td>
<td>1</td>
<td>0.892775</td>
<td>0.105583</td>
<td>0.000000</td>
<td>0.001642</td>
<td>968.8714</td>
<td>11631.35</td>
</tr>
<tr>
<td>3:</td>
<td>2</td>
<td>New</td>
<td>1</td>
<td>0.849871</td>
<td>0.137119</td>
<td>0.009364</td>
<td>0.003644</td>
<td>1072.4091</td>
<td>11269.90</td>
</tr>
<tr>
<td>4:</td>
<td>3</td>
<td>New</td>
<td>1</td>
<td>0.827344</td>
<td>0.145390</td>
<td>0.021361</td>
<td>0.005905</td>
<td>1105.0422</td>
<td>10916.86</td>
</tr>
<tr>
<td>5:</td>
<td>4</td>
<td>New</td>
<td>1</td>
<td>0.811372</td>
<td>0.146369</td>
<td>0.033883</td>
<td>0.008371</td>
<td>1114.3372</td>
<td>10572.54</td>
</tr>
<tr>
<td>6:</td>
<td>5</td>
<td>New</td>
<td>1</td>
<td>0.797601</td>
<td>0.145080</td>
<td>0.046273</td>
<td>0.011045</td>
<td>1114.9710</td>
<td>10236.97</td>
</tr>
</tbody>
</table>
**Cost-effectiveness output**

```r
ce_sim <- sim_out[cycle != 0,
      .(dqalys = sum(dqalys),
       dcosts = sum(dcosts_med) + sum(dcosts_treat)),
      by = c("sample", "strategy")]

ce_sim
```

<table>
<thead>
<tr>
<th>sample</th>
<th>strategy</th>
<th>dqalys</th>
<th>dcosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>1</td>
<td>New</td>
<td>23.30567</td>
</tr>
<tr>
<td>2:</td>
<td>1</td>
<td>SOC</td>
<td>21.36614</td>
</tr>
<tr>
<td>3:</td>
<td>2</td>
<td>New</td>
<td>23.49382</td>
</tr>
<tr>
<td>4:</td>
<td>2</td>
<td>SOC</td>
<td>22.71959</td>
</tr>
<tr>
<td>5:</td>
<td>3</td>
<td>New</td>
<td>23.35243</td>
</tr>
</tbody>
</table>

---

1996: 998 SOC 20.63295 166641.1
1997: 999 New 21.82111 356686.6
1998: 999 SOC 19.58578 104643.0
1999: 1000 New 23.03594 559476.0
2000: 1000 SOC 21.79923 320811.6

*Save for later*

```r
saveRDS(ce_sim, file = "markov-cohort-ce_sim.rds")
```
## Cost-effectiveness output

```r
ce_sim_wider <- dcast(ce_sim, sample ~ strategy,
value.var = c("dqalys", "dcosts"))
```

<table>
<thead>
<tr>
<th>sample</th>
<th>dqalys_New</th>
<th>dqalys_SOC</th>
<th>dcosts_New</th>
<th>dcosts_SOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23.30567</td>
<td>21.36614</td>
<td>372636.3</td>
<td>103030.72</td>
</tr>
<tr>
<td>2</td>
<td>23.49382</td>
<td>22.71959</td>
<td>380159.8</td>
<td>104017.07</td>
</tr>
<tr>
<td>3</td>
<td>23.35243</td>
<td>21.81066</td>
<td>463270.5</td>
<td>207518.50</td>
</tr>
<tr>
<td>4</td>
<td>23.49622</td>
<td>21.45324</td>
<td>680491.2</td>
<td>474308.21</td>
</tr>
<tr>
<td>5</td>
<td>24.11580</td>
<td>22.73777</td>
<td>492869.3</td>
<td>239480.34</td>
</tr>
<tr>
<td>996</td>
<td>23.05054</td>
<td>20.86090</td>
<td>578855.4</td>
<td>336586.31</td>
</tr>
<tr>
<td>997</td>
<td>24.22340</td>
<td>22.48457</td>
<td>379255.5</td>
<td>93578.69</td>
</tr>
<tr>
<td>998</td>
<td>22.22956</td>
<td>20.63295</td>
<td>452278.4</td>
<td>166641.12</td>
</tr>
<tr>
<td>999</td>
<td>21.82111</td>
<td>19.58578</td>
<td>356686.6</td>
<td>104642.99</td>
</tr>
<tr>
<td>1000</td>
<td>23.03594</td>
<td>21.79923</td>
<td>559476.0</td>
<td>320811.62</td>
</tr>
</tbody>
</table>
## Cost-effectiveness output

```r
ce_sim_wider[, idcosts := dcosts_New - dcosts_SOC]
ce_sim_wider[, idqalys := dqalys_New - dqalys_SOC]
```

<table>
<thead>
<tr>
<th>sample</th>
<th>idcosts</th>
<th>idqalys</th>
<th>dcosts_New</th>
<th>dqalys_New</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>23.30567</td>
<td>21.36614</td>
<td>372636.3</td>
<td>103030.72</td>
</tr>
<tr>
<td>2:</td>
<td>23.49382</td>
<td>22.71959</td>
<td>380159.8</td>
<td>104017.07</td>
</tr>
<tr>
<td>3:</td>
<td>23.35243</td>
<td>21.81066</td>
<td>463270.5</td>
<td>207518.50</td>
</tr>
<tr>
<td>4:</td>
<td>23.49622</td>
<td>21.45324</td>
<td>680491.2</td>
<td>474308.21</td>
</tr>
<tr>
<td>5:</td>
<td>24.11580</td>
<td>22.73777</td>
<td>492869.3</td>
<td>239480.34</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>sample</th>
<th>idcosts</th>
<th>idqalys</th>
<th>dcosts_New</th>
<th>dqalys_New</th>
</tr>
</thead>
<tbody>
<tr>
<td>996:</td>
<td>23.05054</td>
<td>20.86090</td>
<td>578855.4</td>
<td>336586.31</td>
</tr>
<tr>
<td>997:</td>
<td>24.22340</td>
<td>22.48457</td>
<td>379255.5</td>
<td>93578.69</td>
</tr>
<tr>
<td>998:</td>
<td>22.22956</td>
<td>20.63295</td>
<td>452278.4</td>
<td>166461.12</td>
</tr>
<tr>
<td>999:</td>
<td>21.82111</td>
<td>19.58578</td>
<td>356686.6</td>
<td>104642.99</td>
</tr>
<tr>
<td>1000:</td>
<td>23.03594</td>
<td>21.79923</td>
<td>559476.0</td>
<td>320811.62</td>
</tr>
</tbody>
</table>

```r
ce_sim_wider[, .(icer = mean(idcosts)/mean(idqalys))]
```

<table>
<thead>
<tr>
<th>icer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: 125028.7</td>
</tr>
</tbody>
</table>
```
Cost-effectiveness plane
Steps

- Define uncertainty for all model input parameters
- Sampling parameter values `hesim::define_rng()`, `hesim::eval_rng()`
- Define treatment strategies and population (data)
- Simulating the Markov model
  ```r
  sim_model(params_rng, data, n_cycles, dr_qalys, dr_costs)
  sim_stateprobs(po, rr, strategy, n_cycles)
  compute_qalys(x, utility, dr)
  compute_costs(x, cost_medical, costs_treat, dr)
  ```
- Reorganize output `rbind_array()`, `array_to_dt()`
- Cost-effectiveness analysis
```
# --- Overview -----------------------------------------------
# @knitr R-packages
library("rceoa")
library("hesim")
library("data.table")
library("magrittr")
library("ggplot2")

# --- Model parameters -------------------------------------
# @knitr tmatrix
transitions_soc <- matrix(c(848, 150, 0, 2,
500, 389, 165, 6,
0, 0, 784, 16,
0, 0, 23),
nrow = 4, byrow = TRUE)
state_names <- c("H", "S1", "S2", "D")
colnames(transitions_soc) <- rownames(transitions_soc) <- tolower(state_names)

# @knitr all-parameters
params <- list(
alp_soc = transitions_soc,
lrr_mean = log(.8),
lrr_lower = log(.71),
lrr_upper = log(.9),
c_medical = c(H = 2000, S1 = 4000, S2 = 15000, D = 0),
c_soc = 2000,
c_new = 12000,
u_mean = c(H = 1, S1 = .75, S2 = .5, D = 0),
u_se = c(H = 0, S1 = .03, S2 = .05, D = 0)
)

# --- Simulation -------------------------------------------
# @knitr sample-parameters
rng.def <- define_rng(
lrr.se <- (lrr.upper - lrr.lower)/(2 * qnorm(.975)) # local object
# not returned
list( # Parameters to return
p_soc <- dirichlet_rng(alp_soc),
rr_new <- lognormal_rng(lrr_mean, lrr.se),
c_medical = gamma_rng(mean = c_medical, sd = c_medical),
c_soc = c_soc,
c_new = c_new,
u = beta_rng(mean = u_mean, sd = u_se)
)
)
```

Tutorial

Overview

Probabilistic sensitivity analysis (PSA) is used to quantify the impact of parameter uncertainty on the uncertainty of model outputs. PSA is typically performed via a simulation approach whereby the model parameters are randomly sampled from suitable probability distributions.

```r
library("rcea")
library("hesim")
library("data.table")
library("magrittr")
library("ggplot2")
```

Model parameters

Transition probabilities for SOC

The probability distribution used for transition probabilities will depend on the underlying data. In this case, we assume that summary level data is available on transitions from the Healthy state \( (n = 900) \), Sick state \( (n = 900) \), and Sicker state \( (n = 800) \). The transitions from each state to the other 4 states can be modeled using a Dirichlet distribution (see Appendix).

```r
transitions_soc <- matrix(
c(848, 150, 0, 2,
200, 600, 0, 0),
byrow = TRUE, nrow = 2)
```
Exercise 2: Incorporating probabilistic sensitivity analysis

- Modify R-script “02-markov-cohort-psa.R”
  - Reduce sample size of data for transition matrix by 50%
  - Increase confidence interval for relative risk

- Run modified script
Simple Markov cohort model with hesim
What is hesim?

- A modular and computationally efficient R package for building simulation models for economic evaluation
- Supports both cohort and individual-level models, encompassing Markov (time-homogeneous and time-inhomogeneous) and semi-Markov processes
  - cohort discrete time state transition models (cDTSTM)
  - individual-level continuous time state transition models (iCTSTM)
  - n-state partitioned survival models (PSM)
- Parameterization by fitting a statistical model in R or by using estimates from external sources
- Nearly all simulation code written in C++ under the hood, but you don’t need to know C++ to use it!
hesim modeling process

1. Parameterization
   - Disease model(s)
   - Utility model
   - Cost model(s)

2. Simulation
   - Disease model(s)
   - Utility model
   - Cost model(s)

3. Decision analysis
   - CEA
   - MCDA
Economic models with *hesim*

1. Model set-up
2. Parameters
3. Simulation
   a. Construction of model
   b. Simulation of outcomes
4. Cost-effectiveness analysis
Model setup

- Define target population and intervention strategies

```r
strategies <- data.frame(
  strategy_id = 1:2,
  strategy_name = c("SOC", "New")
)
patients <- data.frame(
  patient_id = 1,
  age = 25
)
hesim_dat <- hesim_data(
  strategies = strategies,
  patients = patients
)
print(hesim_dat)
```

```r
labs <- get_labels(hesim_dat)
```

```r

$strategies
  strategy_id strategy_name
  1 1 SOC
  2 2 New

$patients
  patient_id age
  1 1 25

attr("class")
[1] "hesim_data"
```
Model parameters

- Same list of parameters as used before

```r
params <- list(
    alpha_soc = transitions_soc,
    lrr_mean = log(.8),
    lrr_lower = log(.71),
    lrr_upper = log(.9),
    c_medical = c(H = 2000, S1 = 4000, S2 = 15000),
    c_soc = 2000,
    c_new = 12000,
    u_mean = c(H = 1, S1 = .75, S2 = 0.5),
    u_se = c(H = 0, S1 = 0.03, S2 = 0.05)
)
```
Model parameters – Random number generation (for PSA)

```r
rng_def <- define_rng(
  lrr_se <- (lrr_upper - lrr_lower)/(2 * qnorm(.975))

  list( # Parameters to return
    p_soc = dirichlet_rng(alpha_soc),
    rr_new = lognormal_rng(lrr_mean, lrr_se),
    c_medical = gamma_rng(mean = c_medical, sd = c_medical),
    c_soc = c_soc,
    c_new = c_new,
    u = beta_rng(mean = u_mean, sd = u_se)
  ), n = 1000)
```

Model parameters – Transformed parameters

- Typically underlying parameters \((params)\) are transformed \((tparams)\) into more relevant parameters for the simulation
  - e.g., predicting an element of a transition probability matrix as a function of the treatment strategy

- We previously did this in the base R PSA example…
```r
sim_model <- function(params_rng, data, n_cycles = 85, dr_qalys = .03, dr_costs = .03){

  # Initialize array of matrices
  n_samples <- attr(params_rng, "n")
  n_strategies <- nrow(data)
  out <- array(NA, dim = c(n_cycles + 1, 7, n_samples * n_strategies))
  dimnames(out) <- list(NULL, c("H", "S1", "S2", "D", "dqalys", "dcosts_med", "dcosts_treat"), NULL)

  # Run the simulation
  i <- 1
  for (s in 1:n_samples){
    # Start PSA loop
    for (k in 1:n_strategies){
      # Start treatment strategy loop
      x <- sim_stateprobs(p0 = params_rng$p_soc[s, ],
                         rr = params_rng$rr_new[s],
                         strategy = data$strategy[k],
                         n_cycles = n_cycles)

      dqalys <- compute_qalys(x, utility = unlist(params_rng$u[s]), dr = dr_qalys)
      dcosts <- compute_costs(x,
                              costs_medical = unlist(params_rng$c_medical[s]),
                              costs_treat = ifelse(data$strategy[k] == "SOC",
                                                   params_rng$c_soc,
                                                   params_rng$c_new),
                              dr = dr_costs)

      out[, , i] <- cbind(x, dqalys, dcosts)
      i <- i + 1
    } # End treatment strategy loop
  } # End PSA loop

  # Store metadata and return
  attr(out, "n_samples") <- n_samples
  attr(out, "strategies") <- data$strategy
  return(out)
}

sim_out <- sim_model(params_rng, data = data)
sim_out <- array_to_dt(sim_out)
```
```r
sim_stateprobs <- function(p0, rr, strategy, n_cycles) {

  rr <- ifelse(strategy == "New", rr, 1)

  p <- tpmatrix(
    C, p0$h_s1 * rr, p0$h_s2 * rr, p0$h_d * rr,
    p0$s1_h, C, p0$s1_s2 * rr, p0$s1_d * rr,
    p0$s2_h, p0$s2_s1, C, p0$s2_d * rr,
    0, 0, 0, 1
  )

  x <- sim_markov_chain(x0 = c(1, 0, 0, 0),
    p = matrix(as.matrix(p), ncol = 4, byrow = TRUE),
    n_cycles = n_cycles)

  return(x)
}
```
Model parameters – Transformed parameters

- A `define_tparams()` block in `hesim` does the same thing, but most of the implementation is done for you (efficiently)

- A `define_tparams()` block returns:
  - `tpmatrix`: The transition probability matrix
  - `utility`: Utility assigned to each health state
  - `costs`: Costs assigned to each health state for each cost category

- All parameters are “transformed” using:
  1. Columns of input data
  2. Parameters returned by `define_rng()`
Model parameters – Transformed parameters

- *Input data* (treatment strategies and patients) can be generated using `expand()`\(^1\)

```r
input_data <- expand(hesim_dat, by = c("strategies", "patients"))
head(input_data)
```

<table>
<thead>
<tr>
<th>strategy_id</th>
<th>patient_id</th>
<th>strategy_name</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>SOC</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>New</td>
<td>25</td>
</tr>
</tbody>
</table>

\(^1\)Could also be expanded by time intervals in a time-inhomogeneous model
Model parameters – Transformed parameters

- You write mathematical expressions
- Vectorized over PSA iterations and input data rows

```r
tparams_def <- define_tparams({
  # The treatment effect (relative risk) varies by
  # strategies (SOC is the reference strategy)
  rr <- ifelse(strategy_name == "SOC", 1, rr_new)
  list(
    tpmatrix = tpmatrix(c,
      p_soc$h_s1 * rr, p_soc$h_s2 * rr, p_soc$h_d * rr,
      p_soc$s1_h, c, p_soc$s1_s2 * rr, p_soc$s1_d * rr,
      p_soc$s2_h, p_soc$s2_s1, 0, c, p_soc$s2_d * rr, 0),
      utility = u,
      costs = list(
        treatment = ifelse(strategy_name == "SOC", c_soc, c_new),
        medical = c_medical
      )
    )
})
```

*tpmatrix*() is a powerful function for creating/storing transition probabilities that vary across PSA samples, strategies, subgroups, and time intervals.

- *C* denotes the "complement", ensuring that the probabilities in a row sum to 1.
- You write mathematical expressions vectorized over PSA iterations and input data rows.
Simulation - Construct the model

- Combine the underlying parameters with the expressions for random number generation and parameter transformation

```
mod_def <- define_model(tparams_def = tparams_def,
rng_def = rng_def,
params = params)
```

- A economic model (of class CohortDtstm) can be created from a defined model (of class model_def) and data using the generic function `create_CohortDtstm()`

```
econmod <- create_CohortDtstm(mod_def, input_data)
```

- This object consists of a
  - transition model for simulating transition probabilities with `$sim_stateprobs()$
  - a utility model for simulating quality-adjusted life-years with `$sim_qalys()$
  - a set of cost models (for each cost category) for simulating costs with `$sim_costs()`
Simulation – Simulating outcomes

- Health state probabilities

```
econmod$sim_stateprobs(n_cycles = 85)
```

- QALYs

```
econmod$sim_qalys(
  dr = 0.03, lys = TRUE,
  integrate_method = "riemann_right"
)
```

- Costs

```
econmod$sim_costs(
  dr = 0.03,
  integrate_method = "riemann_right"
)
```
Cost-effectiveness analysis

- CEAs can be performed directly from the simulation output with hesim.

- First we need to aggregate (i.e., "summarize") costs and QALYs across health states

```r
ce_sim <- econmod$summarize()
```

### Costs

<table>
<thead>
<tr>
<th>category</th>
<th>dr</th>
<th>sample</th>
<th>strategy_id</th>
<th>costs</th>
<th>grp_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: treatment 0.03</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>51194.20</td>
<td>1</td>
</tr>
<tr>
<td>2: treatment 0.03</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>324773.08</td>
<td>1</td>
</tr>
<tr>
<td>3: treatment 0.03</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>48720.41</td>
<td>1</td>
</tr>
<tr>
<td>4: treatment 0.03</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>316828.55</td>
<td>1</td>
</tr>
<tr>
<td>5: treatment 0.03</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>51943.80</td>
<td>1</td>
</tr>
</tbody>
</table>

---

| 5996: total 0.03 | 998 | 2 | 402500.90 | 1 |
| 5997: total 0.03 | 999 | 1 | 197991.68 | 1 |
| 5998: total 0.03 | 999 | 2 | 461755.98 | 1 |
| 5999: total 0.03 | 1000 | 1 | 203541.38 | 1 |
| 6000: total 0.03 | 1000 | 2 | 461558.77 | 1 |

### QALYs

<table>
<thead>
<tr>
<th>dr</th>
<th>sample</th>
<th>strategy_id</th>
<th>qalys</th>
<th>grp_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: 0.03</td>
<td>1</td>
<td>1</td>
<td>19.91854</td>
<td>1</td>
</tr>
<tr>
<td>2: 0.03</td>
<td>1</td>
<td>2</td>
<td>22.34925</td>
<td>1</td>
</tr>
<tr>
<td>3: 0.03</td>
<td>2</td>
<td>1</td>
<td>19.87522</td>
<td>1</td>
</tr>
<tr>
<td>4: 0.03</td>
<td>2</td>
<td>2</td>
<td>22.71959</td>
<td>1</td>
</tr>
<tr>
<td>5: 0.03</td>
<td>3</td>
<td>1</td>
<td>21.39403</td>
<td>1</td>
</tr>
</tbody>
</table>

---

| 1996: 0.03 | 998 | 2 | 23.08819 | 1 |
| 1997: 0.03 | 999 | 1 | 21.25388 | 1 |
| 1998: 0.03 | 999 | 2 | 22.98019 | 1 |
| 1999: 0.03 | 1000 | 1 | 20.75811 | 1 |
| 2000: 0.03 | 1000 | 2 | 23.84501 | 1 |

attr("class")
[1] "ce"

Save for later

```r
saveRDS(ce_sim, file = "markov-cohort-hesim-ce_sim.rds")
saveRDS(hesim_dat, file = "markov-cohort-hesim_data.rds")
```
Cost-effectiveness analysis

- Here, we will consider a pairwise comparison between the new treatment and SOC with the `cea_pw()` function

```r
cea_pw_out <- cea_pw(ce_sim, comparator = 1,
                     dr_qalys = 0.03, dr_costs = 0.03,
                     k = seq(0, 25000, 500))
```

- Although `cea_pw()` allows users to summarize output from a PSA we will just create an ICER table using means for now

```r
format(icer(cea_pw_out, k = 50000, labels = labs))
```

<table>
<thead>
<tr>
<th>Outcome</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Incremental QALYs</td>
<td>2.07 (1.05, 3.13)</td>
</tr>
<tr>
<td>2: Incremental costs</td>
<td>257,297 (203,094, 284,985)</td>
</tr>
<tr>
<td>3: Incremental NMB</td>
<td>-153,833 (-210,941, -77,824)</td>
</tr>
<tr>
<td>4: ICER</td>
<td>124,342</td>
</tr>
</tbody>
</table>
Steps with **hesim**

1. **Model set-up**
   - Specify the treatment strategies and target population(s)
     - `hesim_data()`

2. **Parameters**
   - Estimate or define the parameters of the economic model
     - `define_rng()`, `define_tparams()`

3. **Simulation**
   a. **Construction of model**
      - Create an economic model—consisting of separate statistical models for disease progression, costs, and utilities—that simulate outcomes as a function of *input data* (derived from Step 1) and *parameters* (from Step 2)
      - `define_model()`, `create_CohortDtstm()`
   b. **Simulation of outcomes**
      - Simulate outcomes (disease progression, costs, and quality-adjusted life-years (QALYs)) using the model constructed in Step 3
      - `$sim_stateprobs()`, `$sim_qalys()`, `$sim_costs()`

4. **Cost-effectiveness analysis**
Complete R script

```r
# Complete R script

library("hesim")

strategies <- data.frame(
  strategy_id = 1:2,
  strategy_name = c("SOC", "New")
)

patients <- data.frame(
  patient_id = 1,
  age = 25
)

hesim_dat <- hesim.data(
  strategies = strategies,
  patients = patients
)

print(hesim_dat)

labs <- get_labels(hesim_dat)

transitions_soc <- matrix(
  c(848, 150, 0, 2,
    500, 389, 105, 6,
    0, 0, 784, 16,
    0, 0, 0, 23),
  nrow = 4, byrow = TRUE)

state_names <- c("H", "S1", "S2", "D")
colnames(transitions_soc) <- rownames(transitions_soc) <- tolower(state_names)

params <- list(
  alpha_soc = transitions_soc,
  lrr_mean = log(8),
  lrr_lower = log(.71),
  lrr_upper = log(.9),
  c_medical = c(H = 2000, S1 = 4000, S2 = 15000),
  c_soc = 2000,
  c_new = 12000,
  u_mean = c(H = 1, S1 = .75, S2 = .5),
  u_se = c(H = 8, S1 = .03, S2 = .05)
)
```
Markov Cohort Model with hesim

2021-07-26

https://hesim-dev.github.io/rcea/articles/03-markov-cohort-hesim.html

Overview

This tutorial repeats the probabilistic sensitivity analysis (PSA) of the Markov cohort model simulation performed in the previous tutorial using hesim. We utilize the cohort discrete time state transition model (cDTSTM) class, which is another name for a (time-homogeneous or time-inhomogeneous) Markov cohort model.

More information about hesim can be found by visiting the package website. We recommend reading the “Articles”—starting with the Introduction to hesim—to learn more. Economic models can, in general, be simulated with the following steps:

1. Model setup: Specify the treatment strategies, target population(s), and model structure.
2. Parameters: Estimate or define the parameters of the economic model.
3. Simulation:
   a. Construction of model: Create an economic model—consisting of separate statistical models for disease progression, costs, and utilities—that simulate outcomes as a function of input data (derived from Step 1) and parameters (from Step 2).
   b. Simulation of outcomes: Simulate outcomes (disease progression, costs, and quality-adjusted life-years (QALYs)) using the model constructed in Step 3.

This analysis can be performed using the hesim package alone.

```r
library("hesim")
```

Model setup

Before beginning an analysis, it is necessary to define the treatment strategies of interest and the target population of interest. We
Exercise 3: Markov cohort model with hesim

- **Modify R-script “03-markov-cohort-hesim.R”**
  - Increase confidence interval for relative risk
  - Modify the mean health state utility value
  - Remove impact of the intervention on transitions from “healthy” to “sick”, “sicker”, and “death” (row 72)

- **Run modified script**

```r
tparams_def <- define_tparams(
  ## The treatment effect (relative risk) is transformed so that it varies by strategies (SOC is the reference strategy)
  rr_new <- ifelse(strategy_name == "SOC", 1, rr_new)
  list(
    tpmatrix = tpmatrix(
      c, p_soc$h_s1 * rr, p_soc$h_s2 * rr, p_soc$h_d * rr,
      p_soc$s1_h, c, p_soc$s1_s2 * rr, p_soc$s1_d * rr,
      p_soc$s2_h, p_soc$s2_s1, c, p_soc$s2_d * rr,
      0, 0, 0, 1
    ),
    utility = u,
    costs = list(
      treatment = ifelse(strategy_name == "SOC", c_soc, c_new),
      medical = c_medical
    )
  ))
```
Semi-Markov multi-state model
Semi-Markov multi-state model

- Transition rates can depend on time in intermediate health states (unlike in a Markov model)
- Can only be simulated in a general manner using individual patient simulation (IPS)
- IPS is performed most efficiently using a continuous time state transition model (CTSTM)
- Ideally parameterizing by fitting a multi-state model
Semi-Markov multi-state model

Clock reset
Economic models with \texttt{hesim}

1. Model set-up
2. Parameters
3. Simulation
   a. Construction of model
   b. Simulation of outcomes
4. Cost-effectiveness analysis
Model setup

- The transitions of a multi-state model in hesim must be characterized by a matrix where each element denotes a transition from a row to a column

```r
rmat <- rbind(
  c(NA, 1, 2),
  c(NA, NA, 3),
  c(NA, NA, NA)
)
rownames(rmat) <- c("Stable", "Progression", "Dead")
print(rmat)

<table>
<thead>
<tr>
<th></th>
<th>Stable</th>
<th>Progression</th>
<th>Dead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stable</td>
<td>NA</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Progression</td>
<td>NA</td>
<td>NA</td>
<td>3</td>
</tr>
<tr>
<td>Dead</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>
```
Model setup

```r
n_patients <- 1000

patients <- data.table(
  patient_id = 1:n_patients,
  age = rnorm(n_patients, mean = 45, sd = 7),
  female = rbinom(n_patients, size = 1, prob = .51)
)

strategies <- data.frame(
  strategy_id = 1:2,
  strategy_name = c("SOC", "New")
)
n_strategies <- nrow(strategies)

states <- data.table(
  state_id = c(1, 2),
  state_name = c("Stable", "Progression")
)
n_states <- nrow(states)

hesim_dat <- hesim_data(
  strategies = strategies,
  patients = patients,
  states = states
)
```

As in the cohort model, we must specify the target population and treatment strategies of interest

In an IPS we simulate many patients and then average outcomes across the simulated patients. 1,000 simulated patients should produce reasonably stable results

We also explicitly define the (non-death) health states, which we will use to model utility and costs

We always combine this information into one object
Model setup

```
$strategies
  strategy_id strategy_name
1           1           SOC
2           2           New

$patients
  patient_id age  female
1:          1 42.71774      0
2:          2 48.86723      0
3:          3 40.27539      1
4:          4 46.50052      1
5:          5 47.17538      1
---
996:        996 43.22279      0
997:        997 54.22166      0
998:        998 37.27172      0
999:        999 45.20616      0
1000:       1000 39.15981      1

$states
  state_id state_name
1:        1      Stable
2:        2 Progression

attr("class")
[1] "hesim_data"
```

```
labs <- get_labels(hesim_dat)
```
Parameter estimation

- In the cohort examples, we used parameter estimates from the literature
  - We will continue to do this for utility and costs

- However, in an ideal scenario, we would estimate parameters ourselves using patient-level data
  - We will fit a multi-state model in this manner by estimating transition specific hazards using the R package \texttt{flexsurv}
Parameter estimation – Multi-state model

- Multi-state models can be fit by:
  - Estimating a joint survival model with interaction terms for different transition
  - Fitting separate survival models for each transition
    (Method used here)
## Parameter estimation – Multi-state model

### Dataset

<table>
<thead>
<tr>
<th>1:</th>
<th>from</th>
<th>to strategy_name</th>
<th>female</th>
<th>age</th>
<th>patient_id</th>
<th>time_start</th>
<th>time_stop</th>
<th>status</th>
<th>transition_id</th>
<th>strategy_id</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2:</td>
<td>Stable Progression</td>
<td>New</td>
<td>0</td>
<td>59.85813</td>
<td>1</td>
<td>0.000000</td>
<td>2.420226</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2.420226</td>
</tr>
<tr>
<td>3:</td>
<td>Stable Death</td>
<td>New</td>
<td>0</td>
<td>59.85813</td>
<td>1</td>
<td>0.000000</td>
<td>2.420226</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>2.420226</td>
</tr>
<tr>
<td>4:</td>
<td>Progression Death</td>
<td>New</td>
<td>0</td>
<td>59.85813</td>
<td>1</td>
<td>2.420226</td>
<td>14.620258</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>12.200032</td>
</tr>
<tr>
<td>5:</td>
<td>Stable Progression</td>
<td>New</td>
<td>0</td>
<td>62.57282</td>
<td>2</td>
<td>0.000000</td>
<td>7.497464</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>7.497464</td>
</tr>
</tbody>
</table>

### Estimate parameters

```r
wei_fits <- vector(length = 3, mode = "list")
for (i in 1:3){ # 3 possible transitions
  wei_fits[[i]] <- flexsurvreg(
    Surv(time, status) ~ strategy_name + female,
    data = data,
    subset = (transition_id == i),
    dist = "weibull"
  )
}
wei_fits <- flexsurvreg_list(wei_fits)
```
Parameter estimation – Multi-state model

Stable -> Progression

[[1]]

Call:
flexsurvreg(formula = Surv(time, status) ~ strategy_name + female,
            data = data, subset = (transition_id == i), dist = "weibull")

Estimates:

<table>
<thead>
<tr>
<th></th>
<th>data mean</th>
<th>est</th>
<th>L95%</th>
<th>U95%</th>
<th>se</th>
<th>exp(est)</th>
<th>L95%</th>
<th>U95%</th>
</tr>
</thead>
<tbody>
<tr>
<td>shape</td>
<td>NA</td>
<td>2.0152</td>
<td>1.9226</td>
<td>2.1124</td>
<td>0.0484</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>scale</td>
<td>NA</td>
<td>7.1668</td>
<td>6.7796</td>
<td>7.5762</td>
<td>0.2031</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>strategy_nameNew</td>
<td>0.5281</td>
<td>0.2745</td>
<td>0.2131</td>
<td>0.3360</td>
<td>0.0314</td>
<td>1.3159</td>
<td>1.2375</td>
<td>1.3994</td>
</tr>
<tr>
<td>female</td>
<td>0.4947</td>
<td>-0.1902</td>
<td>-0.2518</td>
<td>-0.1286</td>
<td>0.0314</td>
<td>0.8268</td>
<td>0.7774</td>
<td>0.8793</td>
</tr>
</tbody>
</table>

N = 1975, Events: 1006, Censored: 969
1) Total time at risk: 9192.058
Log-likelihood = -2906.248, df = 4
AIC = 5820.497

New treatment increases time to progression (AFT model)

Shape parameter: whether the hazard is increasing (>1), decreasing (<1), or constant (=1)
Scale: whether the hazard is lower/higher at given time point
## Parameter estimation – Multi-state model

### Stable -> Death

[[2]]

Call:
`flexsurvreg(formula = Surv(time, status) ~ strategy_name + female, data = data, subset = (transition_id == i), dist = "weibull")`

<table>
<thead>
<tr>
<th></th>
<th>data mean</th>
<th>est</th>
<th>L95%</th>
<th>U95%</th>
<th>se</th>
<th>exp(est)</th>
<th>L95%</th>
<th>U95%</th>
<th>shape</th>
<th>NA</th>
<th>2.482459</th>
<th>2.303776</th>
<th>2.675000</th>
<th>0.094614</th>
<th>NA</th>
<th>NA</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>scale</td>
<td>NA</td>
<td>10.406898</td>
<td>9.612760</td>
<td>11.266641</td>
<td>0.421473</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>1.273134</td>
<td>1.170810</td>
<td>1.384402</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>strategy_nameNew</td>
<td>0.528101</td>
<td>0.241482</td>
<td>0.157695</td>
<td>0.325268</td>
<td>0.042749</td>
<td>1.273134</td>
<td>1.170810</td>
<td>1.384402</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>female</td>
<td>0.494684</td>
<td>-0.083337</td>
<td>-0.167300</td>
<td>0.000626</td>
<td>0.042839</td>
<td>0.920041</td>
<td>0.845945</td>
<td>1.000626</td>
<td>0.494684</td>
<td>-0.083337</td>
<td>-0.167300</td>
<td>0.000626</td>
<td>0.042839</td>
<td>0.920041</td>
<td>0.845945</td>
<td>1.000626</td>
<td></td>
</tr>
</tbody>
</table>

N = 1975, Events: 358, Censored: 1617
Total time at risk: 9192.058
Log-likelihood = -1333.968, df = 4
AIC = 2675.936

### Shape parameter: whether the hazard is increasing, decreasing, or constant

### Scale: whether the hazard is lower/higher at given time point
Parameter estimation – Multi-state model

Progression -> Death

[[3]]

Call:
`flexsurvreg(formula = Surv(time, status) ~ strategy_name + female, 
data = data, subset = (transition_id == i), dist = "weibull")`

Estimates:

<table>
<thead>
<tr>
<th></th>
<th>data</th>
<th>mean</th>
<th>est</th>
<th>L95%</th>
<th>U95%</th>
<th>se</th>
<th>exp(est)</th>
<th>L95%</th>
<th>U95%</th>
</tr>
</thead>
<tbody>
<tr>
<td>shape</td>
<td>NA</td>
<td>3.48340</td>
<td>3.25461</td>
<td>3.72826</td>
<td>0.12074</td>
<td>NA</td>
<td>1.00926</td>
<td>0.95780</td>
<td>1.06348</td>
</tr>
<tr>
<td>scale</td>
<td>NA</td>
<td>8.96768</td>
<td>8.55835</td>
<td>9.39658</td>
<td>0.21376</td>
<td>NA</td>
<td>0.50398</td>
<td>0.45930</td>
<td>0.55248</td>
</tr>
<tr>
<td>strategy_nameNew</td>
<td>0.50398</td>
<td>0.00922</td>
<td>-0.04311</td>
<td>0.06155</td>
<td>0.02670</td>
<td>1.00926</td>
<td>0.95780</td>
<td>1.06348</td>
<td></td>
</tr>
<tr>
<td>female</td>
<td>0.52386</td>
<td>-0.11650</td>
<td>-0.16914</td>
<td>-0.06385</td>
<td>0.02686</td>
<td>0.89003</td>
<td>0.84439</td>
<td>0.93815</td>
<td></td>
</tr>
</tbody>
</table>

N = 1006, Events: 468, Censored: 538
Total time at risk: 5479.46
Log-likelihood = -1237.573, df = 4
AIC = 2483.147

Shape parameter: whether the hazard is increasing, decreasing, or constant
Scale: whether the hazard is lower/higher at given time point
Parameters – Utility

```r
utility_tbl <- stateval_tbl(
  data.table(state_id = states$state_id,
              mean = c(.8, .6),
              se = c(0.02, .05)
            ),
  dist = "beta"
)
```

```r
state_id mean se
1: 1 0.8 0.02
2: 2 0.6 0.05
```
Parameters – Medical cost

```r
medcost_tbl <- stateval_tbl(
  data.table(state_id = states$state_id,
              mean = c(2000, 9500),
              se = c(2000, 9500)
  ),
  dist = "gamma")
```

<table>
<thead>
<tr>
<th>state_id</th>
<th>mean</th>
<th>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>2000</td>
<td>2000</td>
</tr>
<tr>
<td>2:</td>
<td>9500</td>
<td>9500</td>
</tr>
</tbody>
</table>
Parameters – Drug cost

n_times <- 2

drugcost_tbl <- stateval_tbl(
  data.table(
    strategy_id = rep(strategies$strategy_id, each = n_states * n_times),
    state_id = rep(rep(states$state_id, each = n_strategies), n_times),
    time_start = rep(c(0, 3/12), n_states * n_strategies),
    est = c(rep(2000, 4), # Costs are always the same with SOC
              12000, 12000, 12000, 10000) # Costs with New drop after 3 months in progression state
  ),
  dist = "fixed"
)

When using an IPS, "state values" (like transition rates) can depend on time in an intermediate health state.

We illustrate by assuming that costs for the new treatment are $12,000 for the first 3 months in the progression state and then $10,000 thereafter.

(Would not be possible in a cohort model without creating tunnel states)
Simulation – Construct the model

Disease model

- The transition model is constructed as a function of the fitted multi-state model and input data (treatment strategy and patients)

```r
transmod_data <- expand(hesim_dat,
                         by = c("strategies", "patients"))

strategy_id patient_id strategy_name age female
1:  1          1          SOC 42.71774      0
2:  1          2          SOC 48.86723      0
3:  1          3          SOC 40.27539      1
4:  1          4          SOC 46.50052      1
5:  1          5          SOC 47.17538      1
6:  1          6          SOC 53.21776      0

transmod <- create_IndivCtstmTrans(wei_fits, transmod_data,
                                      trans_mat = tmat, n = 500,
                                      clock = "reset",
                                      start_age = patients$age)
```
Simulation – Construct the model
Utility and cost models

# Utility
utilitymod <- create_StateVals(utility_tbl, n = 500, hesim_data = hesim_dat)

# Costs
drugcostmod <- create_StateVals(drugcost_tbl, n = 500, time_reset = TRUE, hesim_data = hesim_dat)
medcostmod <- create_StateVals(medcost_tbl, n = 500, hesim_data = hesim_dat)

costmods <- list(Drug = drugcostmod, Medical = medcostmod)

So that costs depend on time in intermediate state
Simulation – Construct the model
Combining the disease progression, cost, and utility models

```r
econmod <- IndivCtstm$new(trans_model = transmod,
                          utility_model = utilitymod,
                          cost_models = costmods)
```
Simulation - Simulating outcomes
Disease progression

\texttt{econmod$sim\_disease(max\_age = 100)}

\texttt{head(econmod$disprog\_)}

\texttt{econmod$sim\_stateprobs(t = seq(0, 30, 1/12))}
\texttt{autoplot(econmod$stateprobs\_, labels = labs)}
Simulation - Simulating outcomes
QALYs and costs

```
$sim_qalys(dr = c(0,.03))

   sample strategy_id grp_id state_id dr     qalys     lys
 1:      1           1      1        1  0 3.795199 5.009791
 2:      1           1      1        2  0 3.433499 5.368864
 3:      1           2      1        1  0 4.898627 6.466354
 4:      1           2      1        2  0 3.491527 5.459601
 5:      2           1      1        1  0 3.882274 4.793997
 6:      2           1      1        2  0 3.220405 5.746082
```

```
$sim_costs(dr = 0.03)

   sample strategy_id grp_id state_id dr category     costs
 1:      1           1      1        1 0.03     Drug  9124.834
 2:      1           1      1        2 0.03     Drug  8261.846
 3:      1           2      1        1 0.03     Drug 69025.893
 4:      1           2      1        2 0.03     Drug 40511.631
 5:      2           1      1        1 0.03     Drug  8778.825
 6:      2           1      1        2 0.03     Drug  8863.685
```
Cost-effectiveness analysis

```r
ce_sim <- econmod$summarize()

cea_pw_out <- cea_pw(ce_sim, comparator = 1,
                       dr_qalys = .03, dr_costs = .03)

format(icer(cea_pw_out, labels = labs))
```

<table>
<thead>
<tr>
<th>Outcome</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Incremental QALYs</td>
<td>0.86 (0.56, 1.17)</td>
</tr>
<tr>
<td>2: Incremental costs</td>
<td>92,981 (81,657, 101,697)</td>
</tr>
<tr>
<td>3: Incremental NMB</td>
<td>-50,160 (-61,751, -38,254)</td>
</tr>
<tr>
<td>4: ICER</td>
<td>108,571</td>
</tr>
</tbody>
</table>
Steps with *hesim*

1. **Model set-up**
   - Specify the treatment strategies, target population(s), and model structure
     - *hesim_data()*

2. **Parameters**
   - Estimate or define the parameters of the economic model
     - *flexsurvreg_list(), stateval_tbl()*

3. **Simulation**
   a. **Construction of model**
      - Create an economic model—consisting of separate statistical models for disease progression, costs, and utilities—that simulate outcomes as a function of *input data* (derived from Step 1) and *parameters* (from Step 2)
      - *create_IndivCtstmTrans(), create_StateVals(), IndivCtstm$new()*
   b. **Simulation of outcomes**
      - Simulate outcomes (disease progression, costs, and quality-adjusted life-years (QALYs)) using the model constructed in Step 3
      - *sim_disease(), sim_stateprobs(), sim_qalys(), sim_costs()*

4. **Cost-effectiveness analysis**
Complete R script

```r
## --- Overview ------------------------------------------
## @knitr R-setup
library("rceo")
library("hesim")
library("data.table")
library("ggplot2")
library("flexsurv")
theme_set(theme_wide())

set.seed(101) # Make random number generation reproducible

## --- Model setup ---------------------------------------
## @knitr tmat
tmat <- rbind(c(NA, 1, 2),
              c(NA, NA, 3),
              c(NA, NA, NA))

rownames(tmat) <- c("Stable", "Progression", "Dead")
print(tmat)

## @knitr hesim_data
n_patients <- 1000
patients <- data.table(
  patient_id = 1:n_patients,
  age = rnorm(n_patients, mean = 45, sd = 7),
  female = rbinom(n_patients, size = 1, prob = .51))

states <- data.table(
  state_id = c(1, 2),
  state_name = c("Stable", "Progression")
  )

n_states <- nrow(states)

strategies <- data.frame(
  strategy_id = 1:2,
  strategy_name = c("SOC", "New")
  )

n_strategies <- nrow(strategies)

hesim.data <- hesim.data(
  strategies = strategies,
  patients = patients,
  states = states)
```

Semi-Markov Multi-state Model

Overview

In this tutorial we use a continuous time state transition model (CTSTM) and relax many of the assumptions made in cohort discrete time state transition models (DTSTMs). First, since the model is in continuous time we do not require model cycles. Second, we estimate the parameters of the health state transitions using a multi-state model so that the simulation model is completely integrated with an underlying statistical model. Third, we use individual patient simulation (IPS) to simulate a semi-Markov model, meaning that (unlike in a Markov model) transitions cannot depend on prior history.

To illustrate, we simplify the sick-sicker model so that it only contains three health states and modify the states—Stable, Progression, and Dead—to mimic an oncology application where patients transition from stable disease to progression to death. There are three transitions: (1) Stable to Progression, (2) Stable to Dead, and (3) Progression to Dead.

1. Stable  
2. Progression  
3. Death

h_{12}(t)  
h_{13}(t)  
h_{23}(t)

The following packages and settings will be used for the analysis. Note that while individual-level simulations can be computationally intensive, they run very quickly in hesim because they are implemented fully in C++ under the hood. You can learn more by looking at the hesim multi-state modeling vignette.
Exercise 4: Semi-Markov multi-state model

- **Modify R-script “04-mstate.R”**
  - Simulate homogeneous patient cohort
  - Use a generalized gamma distribution for transitions (hint: type `?flexsurvreg` into R and look at the options for the `dist` argument)

- **Run modified script**
Partitioned survival model
Steps with **hesim**

1. **Model set-up**
   - Specify the treatment strategies, target population(s), and states
     - `hesim_data()`

2. **Parameters**
   - Estimate or define the parameters of the economic model
     - `flexsurvreg_list()`, `stateval_tbl()`

3. **Simulation**
   a. **Construction of model**
      - Create an economic model—consisting of separate statistical models for disease progression (survival models), costs, and utilities—that simulate outcomes as a function of *input data* (derived from Step 1) and *parameters* (from Step 2)
      - `create_PsmCurves()`, `create_StateVals()`, `Psm$new()`
   b. **Simulation of outcomes**
      - Simulate outcomes using the model constructed in Step 3
      - `$sim_survival()`, `$sim_stateprobs()`, `$sim_qalys()`, `$sim_costs()`

4. **Cost-effectiveness analysis**
Complete R script

```R
# --- Overview ---
library("rms")
library("hesim")
library("data.table")
library("ggplot2")
library("flexsurv")
theme_set(theme_bw())

set.seed(101) # Make PSA reproducible

## --- Model setup --------------------------------------------
## @knitr hesim_data
patients <- data.table(
  patient_id = 1:4,
  patient_wt = rep(1/4, 4), # Each patient has same weight
  age = c(45, 45, 65, 65),
  female = c(0, 1, 0, 1)
)

states <- data.table(
  state_id = c(1, 2),
  state_name = c("Stable", "Progression") # Non-death health states
)

strategies <- data.frame(
  strategy_id = 1:2,
  strategy_name = c("SOC", "New")
)

hesim.dat <- hesim.data(
  patients = patients,
  strategies = strategies,
  states = states
)

print(hesim.dat)

## @knitr labels
labs <- get_labels(hesim_dat)

## --- Parameter estimation -----------------------------------
## @knitr pfs_os_data
onc3 <- hesim::onc3[strategy_name !="New 1"]
onc3[, strategy_name := droplevels(strategy_name)]
levels(onc3$strategy_name) <- c("SOC", "New")
```
Partitioned Survival Model

2021-07-26

Overview

While multi-state models can be used to estimate the parameters of a state transition model (STM) in a very flexible manner, data availability can make it difficult (or infeasible) to fit such a model. This is often the case when an evidence synthesis model based on summary level data is used to parameterize the STM. For example, in oncology, published articles of clinical trials often provide survival curves of progression-free survival (PFS) and overall survival (OS), but do not release information on time to event (and censoring) for each transition. In this setting partitioned survival analysis may consequently be a simpler approach.

We will use the same packages as in the *Semi-Markov Multi-state Model* tutorial.

```r
library("rcea")
library("hesim")
library("data.table")
library("ggplot2")
library("flexsurv")
theme_set(theme_bw())
set.seed(101) # Make PSA reproducible
```

Theory

An 3-state partitioned survival model (PSM) simulates the probability that a patient is in each of 3 distinct health states at a given point of time when treated by a particular therapy. State membership is estimated from 2 survival curves (e.g., PFS and OS) using an *area under the
Cost-effectiveness analysis
Cost-effectiveness analysis

- The optimal treatment strategy is the one that maximizes the expected net monetary benefit (NMB)

\[
NMB(j, \theta) = e_j(\theta) \cdot k - c_j(\theta)
\]

- Can also assess optimal strategy using incremental cost-effectiveness ratio (ICER). Treatment 1 is preferred to treatment 0 if

\[
k > \frac{E_\theta[c_1(\theta) - c_0(\theta)]}{E_\theta[e_1(\theta) - e_0(\theta)]} = ICER
\]

\[1\text{Only true if both incremental costs (numerator) and incremental effectiveness (denominator) are both positive. Treatment 1 dominates treatment 0 if it is more effective and less costly. Treatment 1 is dominated by treatment 0 if it is less effective and more costly. Treatment 1 is preferred to treatment 0 if it is less costly and less effective when } k < ICER.\]
Value of perfect information

- The expected value of perfect information (EVPI) combines the probability of being most effective with the *magnitude* of the expected NMB.

- Intuitively, EVPI is the amount that a decision maker would be willing to pay to collect additional data and completely eliminate uncertainty.

- Mathematically, the EVPI is defined as the difference between the maximum expected NMB given perfect information and the maximum expected NMB given current information.

$$ EVPI = E_\theta [\max_j NMB(j, \theta)] - \max_j E_\theta [NMB(j, \theta)] $$

- *NMB for optimal treatment at each random draw of the parameters*
- *NMB of treatment that is optimal when averaged across all parameter draws*
Economic models with *hesim*

1. Model set-up
2. Parameters
3. Simulation
   a. Construction of model
   b. Simulation of outcomes
4. Cost-effectiveness analysis
Cost-effectiveness analysis with **hesim**

- **hesim** can be used to perform CEA and summarize decision uncertainty
  - Other R packages such as BCEA, dampack, SAVI, and EVSI could also be considered, especially for more advanced value of information analysis

- Implementation via the `cea()` and `cea_pw()` functions
  - `cea()` summarizes results by taking into account each treatment strategy in the analysis
  - `cea_pw()` summarizes “pairwise” results in which each treatment is compared to a comparator

- Both are ”generic” functions that work with (i) data frame like objects or (ii) “ce” objects
Cost-effectiveness analysis with `hesim`

```r
markov_hesim_ce <- readRDS("markov-cohort-hesim-ce_sim.rds")

markov_hesim_ce

$c$ costs

```r
$\begin{array}{cccc}
\text{category} & \text{dr sample} & \text{strategy_id} & \text{costs} & \text{ grp_id} \\
1: & \text{treatment} & 0.03 & 1 & 52301.47 & 1 \\
2: & \text{treatment} & 0.03 & 1 & 323892.01 & 1 \\
3: & \text{treatment} & 0.03 & 2 & 49767.92 & 1 \\
4: & \text{treatment} & 0.03 & 2 & 322401.36 & 1 \\
5: & \text{treatment} & 0.03 & 3 & 52661.20 & 1 \\
5996: & \text{total} & 0.03 & 998 & 434677.65 & 1 \\
5997: & \text{total} & 0.03 & 999 & 165176.48 & 1 \\
5998: & \text{total} & 0.03 & 999 & 446988.07 & 1 \\
5999: & \text{total} & 0.03 & 1000 & 79522.29 & 1 \\
6000: & \text{total} & 0.03 & 1000 & 360039.77 & 1 \\
\end{array}$
```

```r
wtp <- seq(0, 250000, 500) #Willingness to pay per QALY

cea_pw_out <- cea_pw(markov_hesim_ce, comparator = "SOC", 
  sample = "sample", strategy = "strategy",
  e = "dqalys", c = "dcosts",
  k = wtp 
)

cea_out <- cea(markov_hesim_ce, 
  sample = "sample", strategy = "strategy",
  e = "dqalys", c = "dcosts",
  k = wtp 
)
```

$q$ qalys

```r
$\begin{array}{cccc}
\text{dr sample} & \text{strategy_id} & \text{qalys} & \text{ grp_id} \\
1: & 0.03 & 1 & 20.90995 & 1 \\
2: & 0.03 & 1 & 22.24340 & 1 \\
3: & 0.03 & 2 & 20.77562 & 1 \\
4: & 0.03 & 2 & 23.51086 & 1 \\
5: & 0.03 & 3 & 20.89219 & 1 \\
1996: & 0.03 & 998 & 24.15825 & 1 \\
1997: & 0.03 & 999 & 20.88165 & 1 \\
1998: & 0.03 & 999 & 23.63537 & 1 \\
1999: & 0.03 & 1000 & 20.77720 & 1 \\
2000: & 0.03 & 1000 & 23.02164 & 1 \\
\end{array}$
```
Incremental cost-effectiveness ratio with `hesim`

```r
icer(cea_pw_out, wtp = 50000) %>%
format()
```

<table>
<thead>
<tr>
<th>Outcome</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Incremental QALYs</td>
<td>2.09 (1.04, 3.14)</td>
</tr>
<tr>
<td>2: Incremental costs</td>
<td>261,594 (212,238, 291,654)</td>
</tr>
<tr>
<td>3: Incremental NMB</td>
<td>-156,981 (-215,676, -83,866)</td>
</tr>
<tr>
<td>4: ICER</td>
<td>125,029</td>
</tr>
</tbody>
</table>
Representing decision uncertainty with **hesim**

**plot_ceplane(cea_pw_out)**

**plot_ceac(cea_pw_out)**

**plot_ceaf(cea_out)**

**plot_evpi(cea_out)**
Complete R script

```r
# --- Overview --
library("hesim")
library("ggplot2")
library("magrittr")
theme_set(theme_minimal()) # Set ggplot2 theme

# --- Application -----------------------------
markov_ce <- readRDS("markov-cohort-ce_sim.rds")
markov_ce

# @knitr load-ce
markov_ce <- readRDS("markov-cohort-ce_sim.rds")
markov_ce

# @knitr load-hesim-ce
hesim_dat <- readRDS("markov-cohort-hesim_ce_data.rds")
hesim_dat

# @knitr load-ce-out
markov_hesim_ce <- readRDS("markov-cohort-hesim-ce_out_100000.rds")
markov_hesim_ce

# @knitr conduct-ce
wtp <- seq(0, 250000, 500) # Willingness to pay per QALY
cea_pw_out <- cee_pw(markov_hesim_ce,
                      comparator = 1, # Comparator is SOC (ID = 1)
                      dr_qalys = 0.03, dr_costs = 0.03,
                      wtp)

# @knitr conduct-ce-default
cea_pw_out2 <- cee_pw(markov_ce, comparator = "SOC",
                      k = wtp,
                      sample = "sample", strategy = "strategy",
                      e = "dqlays", c = "dcosts")

# --- Incremental cost-effectiveness ratio -----------------------------
# @knitr icer
icer(cea_pw_out, wtp = 50000, labels = labs) %>%
  format()

# --- Cost-effectiveness plane ---------------------------------
# @knitr ceplane-plot
plot_ceplane(cea_pw_out, k = 100000, labels = labs)
```
Cost-effectiveness Analysis

2021-07-26

https://hesim-dev.github.io/rcea/articles/06-cea.html

Overview

The prior tutorials have focused on constructing economic models to simulate disease progression, costs, and quality-adjusted life-years (QALYs). While incremental cost-effectiveness ratios (ICERs) have been computed and probabilistic sensitivity analysis (PSA) has been employed, we have not yet formalized cost-effectiveness analysis (CEA) or represented decision uncertainty.

In this analysis we will perform a CEA given the output of model from the “Semi-Markov Multi-state Model” tutorial. We will use the CEA functions from hesim to summarize decision uncertainty and ggrepplot2 for visualization. The CEA will be performed for a single target population, but you can review the hesim tutorial on CEA and the references therein for an example of CEA in the context of multiple subgroups.

```r
library("hesim")
library("ggplot2")
library("magrittr")
theme_set(theme_minimal()) # Set ggrepplot2 theme
```

Theory

CEA is based on estimating the net monetary benefit (NMB). For a given parameter set θ, the NMB with treatment j is computed as the difference between the monetized health gains from an intervention less costs, or,

\[ NMB(j, \theta) = e_j(\theta) \cdot k - c_j(\theta), \]

where \( e_j \) and \( c_j \) are measures of health outcomes (e.g., QALYs) and costs using treatment \( j \) respectively, and \( k \) is a decision makers willingness to pay for an additional QALY.
Summary
So why R for CE modeling?

- One platform to do everything
  - parameter estimation
  - simulation
- More complex analysis and individual patient simulation
- Your problems are rarely unique
  - But even if they are, R facilitates development of custom models
- Easier to share and review
- Reproducible
Cost-effectiveness analysis with R

- BCEA
- heemod
- hesim
- ...

146
Why *hesim*?

- Focus on setting up and interpreting a model rather than implementation
  - Burdensome programming tasks have already been implemented and optimized

- Flexible enough to cover many problems
  - May need to write custom code for very complex or unique problems (beyond scope of course)

- Very fast!
  - IPS in *hesim* with 100 PSA iterations ran in .44 seconds; equivalent simulation in *mstate* package took 34 minutes (see [here](#))
  - Cohort model in *hesim* with 1,000 PSA iterations ran in 1 second (IPS in 9 seconds), while equivalent cohort model in *heemod* took 85 seconds (see [here](#))
User interfaces with R Shiny

Lesson 1
Welcome to Shiny

Shiny is an R package that makes it easy to build interactive web applications (apps) straight from R. This lesson will get you started building Shiny apps right away.

install.packages("shiny")

Examples

https://shiny.rstudio.com/

Making Markov Models Shiny: A Tutorial

Robert Smith and Paul Schneider, SchARR, University of Sheffield

Sick Sicker Model in Shiny

https://r-hta.org/tutorial/markov_models_shiny/
<table>
<thead>
<tr>
<th>Line of therapy</th>
<th>Treatment regimen</th>
<th>Sequence number</th>
</tr>
</thead>
<tbody>
<tr>
<td>T790M Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>gefitinib</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>erlotinib</td>
<td>II</td>
</tr>
<tr>
<td></td>
<td>afatinib</td>
<td>III</td>
</tr>
<tr>
<td></td>
<td>dacomitinib</td>
<td>IV</td>
</tr>
<tr>
<td></td>
<td>osimertinib</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>pbdc</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pbdc + bevaczumab</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pbdc + nivolumab</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pbdc + pembrolizumab</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pbdc + atezolizumab</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pbdc + atezolizumab + pembrolizumab</td>
<td></td>
</tr>
</tbody>
</table>

**Cost-effectiveness plane**

- Expected outcomes
- ICER
- CE PLANE
- CEAC

- **Sequence comparator**
  - I
  - II
  - III
  - IV

- **Graph**
  - X-axis: Incremental costs
  - Y-axis: Incremental QALYs
  - Colors indicate sequence: Orange = Sequence II, Blue = Sequence III, Pink = Sequence IV
pats <- create_patients(n = 100)

txseq1 <- txseq(
  first = c("gefitinib"),
  second = c("osimertinib", "PBDC"),
  second_plus = c("PBDC + atezolizumab", "PBDC + atezolizumab")
)

txseq2 <- txseq(
  first = c("erlotinib"),
  second = c("osimertinib", "PBDC"),
  second_plus = c("PBDC + atezolizumab", "PBDC + atezolizumab")
)

txseq3 <- txseq(
  first = c("afatinib"),
  second = c("osimertinib", "PBDC"),
  second_plus = c("PBDC + atezolizumab", "PBDC + atezolizumab")
)

txseq4 <- txseq(
  first = c("osimertinib"),
  second = c("PBDC", "PBDC"),
  second_plus = c("PBDC + atezolizumab", "PBDC + atezolizumab")
)
Thank you

devin.incerti@gmail.com
jeroen.jansen@ucsf.edu
Appendix – Building a model with hesim
### hesim overview

<table>
<thead>
<tr>
<th>Economic model</th>
<th>Object</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cohort discrete time state transition model (cDTSTM)</td>
<td>CohortDtstm</td>
<td><code>create_CohortDtstm()</code> <code>create_CohortDtstm$new()</code></td>
</tr>
<tr>
<td>Individual-level continuous time state transition model (iCTSTM)</td>
<td>IndivCtstm</td>
<td><code>IndivCtstm$new()</code></td>
</tr>
<tr>
<td>N-state partitioned survival model (PSM)</td>
<td>Psm</td>
<td><code>Psm$new()</code></td>
</tr>
</tbody>
</table>
### hesim - Parameterization

<table>
<thead>
<tr>
<th>Economic model</th>
<th>Disease progression</th>
<th>Statistical model</th>
<th>Parameter Object</th>
<th>Model fit Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>cDTSTM</td>
<td>CohortDtstm</td>
<td>Custom</td>
<td>tparams_transprobs</td>
<td>define_model()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Multinomial logistic regression</td>
<td>params_mlogit_list</td>
<td>multinom_list()</td>
</tr>
<tr>
<td>iCTSTM</td>
<td>IndivCtstm</td>
<td>Multi-state model</td>
<td>params_surv</td>
<td>flexsurv::flexsurvreg()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(joint likelihood)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Multi-state model</td>
<td>params_surv_list</td>
<td>flexsurvreg_list()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(transition-specific)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSM</td>
<td>Psm</td>
<td>Independent survival models</td>
<td>params_surv_list</td>
<td>flexsurvreg_list()</td>
</tr>
</tbody>
</table>
hesim - Parameterization

Cost and utility

<table>
<thead>
<tr>
<th>Statistical model</th>
<th>Parameter Object</th>
<th>Model fit Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted means</td>
<td>tparams_mean</td>
<td>stateval_tbl()</td>
</tr>
<tr>
<td></td>
<td>tparams_mean</td>
<td>define_model()</td>
</tr>
<tr>
<td>Linear model</td>
<td>params_lm</td>
<td>Stats::lm()</td>
</tr>
</tbody>
</table>
# hesim - Simulation

## Constructing an economic model

<table>
<thead>
<tr>
<th>Economic model</th>
<th>Disease model</th>
<th>Utility model</th>
<th>Cost model(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cDTSTM</td>
<td>CohortDtstmTrans</td>
<td>StateVals</td>
<td>StateVals</td>
</tr>
<tr>
<td></td>
<td>create_CohortDtstmTrans()</td>
<td>create_StateVals()</td>
<td>create_StateVals()</td>
</tr>
<tr>
<td>iCTSTM</td>
<td>IndivCtstmTrans</td>
<td>StateVals</td>
<td>StateVals</td>
</tr>
<tr>
<td></td>
<td>create_IndivCtstmTrans()</td>
<td>create_StateVals()</td>
<td>create_StateVals()</td>
</tr>
<tr>
<td>PSM</td>
<td>PsmCurves</td>
<td>StateVals</td>
<td>StateVals</td>
</tr>
<tr>
<td></td>
<td>create_PsmCurves()</td>
<td>create_StateVals()</td>
<td>create_StateVals()</td>
</tr>
</tbody>
</table>
# hesim - Simulation

Simulating outcomes

<table>
<thead>
<tr>
<th>Economic model</th>
<th>Disease progression</th>
<th>QALYs</th>
<th>Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>cDTSTM</td>
<td>CohortDtstm</td>
<td>sim_stateprobs()</td>
<td>sim_qalys()</td>
</tr>
<tr>
<td>iCTSTM</td>
<td>IndivCtstm</td>
<td>sim_disease()</td>
<td>sim_qalys()</td>
</tr>
<tr>
<td>PSM</td>
<td>Psm</td>
<td>sim_survival()</td>
<td>sim_qalys()</td>
</tr>
</tbody>
</table>