Cohort state-transition models in R: From conceptualization to implementation

Fernando Alarid-Escudero, PhD[*]
Eline M. Krijkamp, MSc[†]
Eva A. Enns, PhD[‡]
M.G. Myriam Hunink, PhD[†§]
Petros Pechlivanoglou, PhD[¶]
Hawre Jalal, MD, PhD[‖]

2020-01-17

Abstract

Decision models can synthesize evidence from different sources to provide estimates of long-term consequences of a decision with uncertainty. Cohort state-transition models (cSTM) are decision models commonly used in medical decision making because they can simulate hypothetical cohorts' transitions across various health states over time. This tutorial shows how to conceptualize cSTMs in a programming language environment and shows examples of their implementation in R. We illustrate their use in a cost-effectiveness analysis of a treatment using a previously published testbed cSTM. Both time-independent cSTM where transition probabilities are constant over time and time-dependent cSTM where transition probabilities vary over time are represented. For the time-dependent cSTM, we consider transition probabilities dependent on age and state residence. We also illustrate how this setup can facilitate the computation of epidemiological outcomes of interest, such as survival and prevalence. We conclude by demonstrating how to calculate economic outcomes and conducting a cost-effectiveness analysis of a treatment compared to usual care using the testbed model. We provide a link to a public repository with all the R code described in this tutorial that can be used to replicate the example or to be modified to suit different decision modeling needs.

1 Introduction

Healthcare policy makers are frequently faced with decisions about how to allocate healthcare resources under constrained budgets. These decisions are often informed by health economic evaluations that rely on decision models to synthesize evidence from different sources and project long-term outcomes of different alternatives. A commonly used decision model is the cohort state-transition model (cSTM), often called a Markov model.[1]

In a recent review of R modeling packages, we illustrated the increased utilization of the statistical programming framework R in health decision sciences and provided a collection of resources for its application in medical

[*] Drug Policy Program, Center for Research and Teaching in Economics (CIDE) - CONACyT, Aguascalientes, AGS, Mexico
[†] Department of Epidemiology, Erasmus University Medical Center, Rotterdam, The Netherlands
[‡] Division of Health Policy and Management, University of Minnesota School of Public Health, Minneapolis, MN, USA
[§] Center for Health Decision Sciences, Harvard T.H. Chan School of Public Health, Boston, USA
[¶] The Hospital for Sick Children, Toronto and University of Toronto, Toronto, Ontario, Canada
[‖] University of Pittsburgh, Pittsburgh, PA, USA

decision making.[2] Despite its increasing use, there is limited guidance on how to conceptualize and implement cohort-based decision models in a programming language. The purpose of this tutorial is twofold: (1) to conceptualize cSTMs in a programming language, and (2) to provide guidance on how to implement these cSTMs in R.

We first provide a description of each of the components of a cSTM. Then, we illustrate the implementation of these components with an illustrative example. Our general conceptualization should apply to other programming languages (e.g., MATLAB, Python, C++). The full R code used in this tutorial is provided in the supplementary materials so that readers can replicate and modify the example to fit their needs. The most updated code can be found on GitHub (https://github.com/DARTH-git/Cohort-modeling-tutorial). We assume that the reader is familiar with the basics of decision modelling and coding in programming languages. Thus, a gentle introduction to R and to linear algebra for decision modelers is suggested.

2 Cohort state-transition models

Cohort state-transition models (cSTMs) are deterministic mathematical models that simulate hypothetical cohorts over time. If the decision problem can be described with a reasonable number of health states, a cSTM can be used because of its transparency, efficiency, ease of debugging and ability to conduct specific value-of-information analyses.[3] For these reasons, cSTMs have been used to evaluate screening and surveillance programs,[4,5] diagnostic procedures,[6] disease management programs,[7] and interventions.[8,9]

A cSTM consists of a set of n_s mutually exclusive and collectively exhaustive health states. The cohort is assumed to be homogeneous (i.e., they all have the same characteristics at the beginning of the simulation), differing only in their current health state. That is, all persons residing in a particular health state are indistinguishable from one another. The cohort can transition between health states with defined probabilities, called "transition probabilities". A transition probability represents the chance that individuals in the cohort residing in a state in a given cycle, transition to another state or remain in the same state for the next cycle. Transition probabilities only depend on the current health state in a given cycle and do not depend on the history prior to that cycle, which is often referred to as the "Markovian assumption".[10–12] This means that in a cSTM, transition probabilities do not depend on the history of past transitions or time spent in a given state (i.e., state residence).

cSTMs are classified as either time-homogeneous (time-independent) or time-inhomogeneous (time-dependent). Time-independent cSTMs have constant transition probabilities (i.e., the probability of any state transition is independent of time), while time-dependent cSTMs have transition probabilities that vary over time (i.e., the probability of any state transition is dependent of time). Time-independent models are simpler to implement than time-dependent but most problems in healthcare are best modeled with time-dependent cSTMs because they can capture the increasing age-specific background mortality as the cohort ages (age dependency) and dependency on the amount of time spent in a given state (state residence). In this tutorial, we will first cover time-independent models and will expand them to account for both age and state residence dependency.

In the next sections, we will illustrate the conceptualization and implementation of a cSTM in R to conduct a cost-effectiveness analysis (CEA) using an illustrative example described in the next section. Researchers can easily then customize this application to their own needs.

3 Case study: Sick-Sicker model

We illustrate the implementation of a cSTM for a CEA using a previously published 4-state Sick-Sicker model.[13,14] The model is used to quantify the expected costs and quality-adjusted life years (QALYs) for individuals at risk of a hypothetical disease with two different stages, "Sick" and "Sicker". We then evaluate the cost-effectiveness of a hypothetical treatment that improves quality of life (QoL) for those in the Sick state.[8] We use the Sick-Sicker model to illustrate both time-independent and time-dependent cSTMs. For the time-dependent model, we first consider age dependency and expand it to also account for state residence dependency. All the parameters of the Sick-Sicker model and the corresponding R variable names are presented in Table 1 and follow the notation described in the DARTH coding framework.[15] Briefly, we define variables by `<x>_<y>_<var_name>`, where `x` indicates the data type [e.g., scalar (no prefix), vector, matrix, data frame], `y` is the variable type (e.g., probability, rate, relative risk, cost, utility, etc.), and `var_name` is some description of the variable presented separated by underscores. For example, `c_Trt` denotes the cost of the treatment where `c_` refers to cost followed by the variable name `Trt`. In later sections we will define all the naming of all the other parameters.

In the Sick-Sicker model, we simulate a hypothetical cohort of 25-year-old individuals over a lifetime (until a maximum age of 110 years old) who all start in the "Healthy" health state (denoted "H"). This means that we will simulate the cohort for 85 cycles. The total number of cycles is denoted as n_t and defined in R as `n_t`. Healthy individuals are at risk of developing the disease, at which point they transition to the "Sick" health state (denoted by "S1"). Individuals that become sick incur a one-time utility decrement of 0.01 (`du_HS1`, disutility of transitioning from H to S1) and a transition cost of $1,000 (`ic_HS1`) that reflects the acute impacts of developing the illness. Sick individuals are at risk of further progressing to a more severe disease stage, the "Sicker" health state (denoted by "S2"). In the time-independent model, the risk of progressing from Sick to Sicker is constant and in the time-dependent model, this risk is a function of the duration of time spent in the Sick state. Individuals in the Sick state can recover and return to the Healthy state. However, once individuals reach the Sicker state, they cannot recover; that is, the probability of transitioning to the Sick or Healthy health states from the Sicker health state is zero. Individuals in the Healthy state face background mortality that could be either constant for the time-independent model or age-dependent for the time-dependent model. For the age-dependency, Sick and Sicker individuals face an increased mortality compared to Healthy individuals in the form of a hazard ratio (HR) of 3 and 10, respectively, relative to the background mortality rate. Sick and Sicker individuals also experience increased health care costs (in addition to the one-time cost of $1,000) and reduced QoL compared to healthy individuals. Once simulated individuals die, they transition to the absorbing "Dead" state (denoted by "D"), where they remain, and incur a one-time cost of $2,000 (`ic_D`) that reflects the acute care that might be received immediately preceding death. The state-transition diagram of the Sick-Sicker model is shown in Figure 1. The evolution of the cohort is simulated in one-year discrete-time cycles. Both costs and QALYs are discounted at an annual rate of 0.03%.

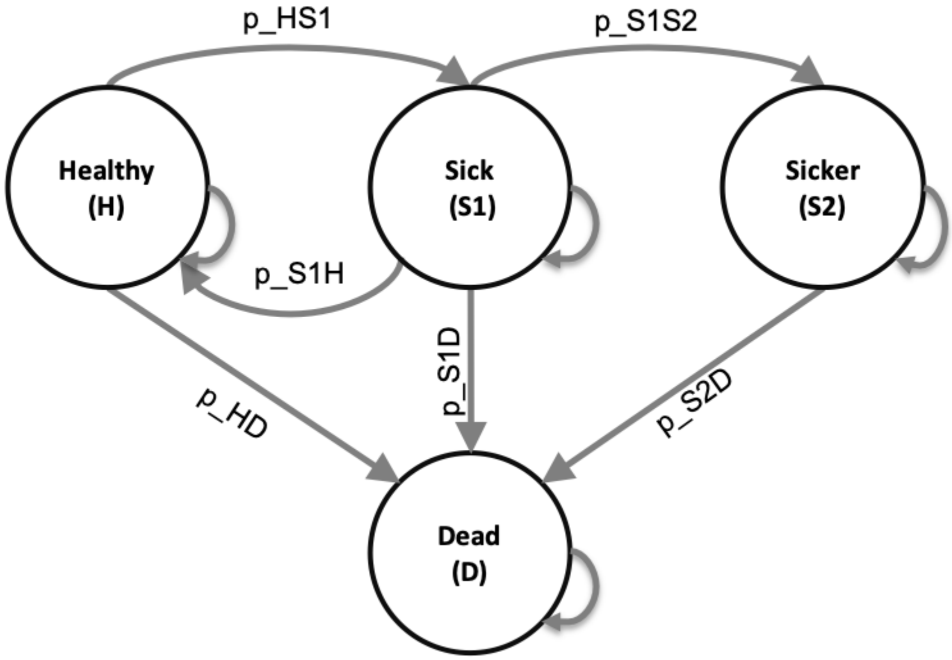


Figure 1: State-transition diagram of the time-independent Sick-Sicker cohort state-transition model with the name of the health states and possible transitions with their corresponding transition probabilities.

We are interested in evaluating the cost-effectiveness of a hypothetical new treatment for individuals affected with the disease compared to usual care (the status quo). The treatment costs $12,000 per year (`c_Trt`). The effect of this new treatment is an increase in the QoL of individuals in the Sick state from 0.75 (utility without treatment, `u_S1`) to 0.95 (utility with treatment, `u_Trt`). Treatment does not have any impact on the QoL of individuals in the Sicker state, nor does it change the risk of becoming sick or progressing through the sick states. We assume that it is not possible to distinguish between Sick and Sicker patients; therefore, individuals in both sick states receive the treatment. Note that the transition probabilities of the cSTM for the new treatment strategy is structurally identical to that of the usual-care strategy. The only difference is the added cost of the treatment for those in the Sick or Sicker state and the increase in QoL for those in the Sick state. After comparing the two strategies in terms of expected QALYs and costs, we calculate the incremental cost per QALY gained of treatment compared to usual care.

Table 1: Description of parameters, the parameters' names in R and their values

	Parameter	
Parameter	**Name**	**Value**
Time horizon (n_t)	`n_t`	85 years
Names of health states (n)	`v_n`	H, S1, S2, D
Annual discount rate for costs	`d_c`	3%
Annual discount rate for QALYs	`d_e`	3%
Annual transition probabilities		
- Disease onset (H to S1)	`p_HS1`	0.15

Parameter	R name	Value
- Recovery (S1 to H)	`p_S1H`	0.5
- Disease progression (S1 to S2) in the time-independent model	`p_S1S2`	0.105
- Time-dependent disease progression (S1 to S2)	`p_S1S2_tunnels`	
Weibull parameters		
Scale (λ)	`l`	0.08
Shape (γ)	`g`	1.10
Annual mortality		
- All-cause mortality (H to D)	`p_HD`	0.002 or age-specific
- Hazard ratio of death in S1 vs H	`hr_S1`	3
- Hazard ratio of death in S2 vs H	`hr_S2`	10
Annual costs		
- Healthy individuals	`c_H`	\$2,000
- Sick individuals in S1	`c_S1`	\$4,000
- Sick individuals in S2	`c_S2`	\$15,000
- Dead individuals	`c_D`	\$0
- Additional costs of sick individuals treated in S1 or S2	`c_Trt`	\$12,000
Utility weights		
- Healthy individuals	`u_H`	1.00
- Sick individuals in S1	`u_S1`	0.75
- Sick individuals in S2	`u_S2`	0.50
- Dead individuals	`u_D`	0.00
Intervention effect		
- Utility for treated individuals in S1	`u_Trt`	0.95
Transition rewards		
- Utility decrement of healthy individuals when transitioning to S1	`du_HS1`	0.01
- Cost of healthy individuals when transitioning to S1	`ic_HS1`	\$1,000
- Cost of dying	`ic_D`	\$2,000

Below, we show how to input the parameters in R. All the code is stored as a GitHub repository at https://github.com/DARTH-git/Cohort-modeling-tutorial. We use the prefixes "`v_`", "`m_`" and "`a_`" to represent a vector, a matrix and a multidimensional array, respectively, consistent with the DARTH framework.[9]

```r
## General setup
n_age_init <- 25  # age at baseline
n_age_max  <- 110 # maximum age of follow up
n_t <- n_age_max - n_age_init # time horizon, number of cycles
v_n <- c("H", "S1", "S2", "D") # the 4 health states of the model:
                               # Healthy (H), Sick (S1), Sicker (S2), Dead (D)
```

```r
n_states <- length(v_n) # number of health states
d_c <- d_e <- 0.03 # equal discount of costs and QALYs by 3%
v_names_str <- c("Usual care", "New treatment") # store the strategy (str) names

## Transition probabilities (per cycle)
p_HD    <- 0.002 # constant probability of dying when Healthy (all-cause mortality)
p_HS1   <- 0.15  # probability of becoming Sick when Healthy
p_S1H   <- 0.5   # probability of becoming Healthy when Sick
p_S1S2  <- 0.105 # probability of becoming Sicker when Sick
hr_S1   <- 3     # hazard ratio of death in Sick vs Healthy
hr_S2   <- 10    # hazard ratio of death in Sicker vs Healthy

## Cost and utility inputs
# State rewards
c_H   <- 2000  # cost of being Healthy for one cycle
c_S1  <- 4000  # cost of being Sick for one cycle
c_S2  <- 15000 # cost of being Sicker for one cycle
c_D   <- 0     # cost of being dead (per cycle)
c_Trt <- 12000 # cost of treatment (per cycle)

u_H   <- 1     # utility of being Healthy for one cycle
u_S1  <- 0.75  # utility of being Sick for one cycle
u_S2  <- 0.5   # utility of being Sicker for one cycle
u_D   <- 0     # utility of being dead (per cycle)
u_Trt <- 0.95  # utility when being treated for one cycle

# Transition rewards
du_HS1 <- 0.01  # one-time utility decrement when transitioning from Healthy to Sick
ic_HS1 <- 1000  # one-time cost when transitioning from Healthy to Sick
ic_D   <- 2000  # one-time cost when dying
```

To compute the mortality risks from the Sick and Sicker states, we transform `p_HD` to a rate assuming a constant exponential rate, `-log(1 - p_HD)`, multiply it by the hazard ratios `hr_S1` and `hr_S2`, respectively, and then convert back. See the R code below for more details on the calculations.

```r
p_S1D  <- 1 - exp(log(1 - p_HD) * hr_S1) # probability of dying when Sick
p_S2D  <- 1 - exp(log(1 - p_HD) * hr_S2) # probability of dying when Sicker
```

4 Conceptualizing a cSTM

In general, a cSTM consists of four core components: (1) a state vector, $\mathbf{m}_t$, that stores the distribution of the cohort across all health states in cycle t where $t = 0, \ldots, n_t$; (2) the cohort trace matrix, M, that *stores* the distribution of the cohort over time and is simply the collection of $\mathbf{m}_t$ for all t; and (3) a transition probability matrix, P_t.[16] If the cSTM is comprised of n_s discrete health states, $\mathbf{m}_t$ is a $1 \times n_s$ vector and P_t is a $n_s \times n_s$ matrix. The i-th element of $\mathbf{m}_t$ represents the proportion of the cohort in the i-th health state in cycle t, referred to as $m_{[t,i]}$. Thus, $\mathbf{m}_t$ is written as

$$\mathbf{m}_t = \begin{bmatrix} m_{[t,1]} & m_{[t,2]} & \cdots & m_{[t,n_s]} \end{bmatrix}.$$

The elements of P_t are the transition probabilities of moving from state i to state j, potentially as a function of time t, $p_{[i,j,t]}$, where $\{i,j\} \in 1,\ldots,n_s$ and $t = 0,\ldots,n_t$

$$P_t = \begin{bmatrix} p_{[1,1,t]} & p_{[1,2,t]} & \cdots & p_{[1,n_s,t]} \\ p_{[2,1,t]} & p_{[2,2,t]} & \cdots & p_{[2,n_s,t]} \\ \vdots & \vdots & \ddots & \vdots \\ p_{[n_s,1,t]} & p_{[n_s,2,t]} & \cdots & p_{[n_s,n_s,t]} \end{bmatrix}.$$

Note that all rows of the transition probability matrix in each cycle t must sum to one, $\sum_{j=1}^{n_s} p_{[i,j,t]} = 1$ for all $i = 1,\ldots,n_s$ and $t = 0,\ldots,n_t$.

The state vector at cycle $t+1$, $\mathbf{m}_{t+1}$ is then calculated by the inner product of the state vector at cycle t, $\mathbf{m}_t$, and the transition probability matrix that the cohort faces in cycle t, P_t, such that,

$$\mathbf{m}_{t+1} = \mathbf{m}_t P_t \text{ for } t = 0,\ldots,(n_t - 1),$$

where $\mathbf{m}_t$ is initialized at $\mathbf{m}_0$, which represents the initial state vector with the proportion of the cohort distributed across all health states at the start of the simulation that begins at cycle 0.

The cohort trace matrix, M, is a matrix of dimensions $(n_t + 1) \times n_s$ where each row is a state vector $(-\mathbf{m}_t-)$

$$M = \begin{bmatrix} -\mathbf{m}_0- \\ -\mathbf{m}_1- \\ \vdots \\ -\mathbf{m}_{n_t}- \end{bmatrix}.$$

M is a commonly used matrix to store the output of cSTM, which could be used to produce epidemiological outcomes, such as prevalence and survival, and economic outcomes, such as cumulative resource use and costs. Note that the initial cycle corresponds to $t = 0$, which is stored on the first row of M. Table 2 shows a description of each of the core components of cSTM and their suggested name in R code.

Table 2: Components of cSTM with their R name

Element	Description	R name
$\mathbf{m}_0$	Initial state vector	`v_s_init`
$\mathbf{m}_t$	State vector in cycle t	`v_mt`
M	Cohort trace matrix	`m_M`
P	Time-independent transition probability matrix	`m_P`
$\mathbf{P}$	Time-dependent transition probability array	`a_P`

Element	Description	R name
A	Transition array	`a_A`

For the Sick-Sicker model, the entire cohort starts in the Healthy state. Therefore, we create the initial state vector `v_s_init` with all of the cohort assigned to the H state.

```r
v_s_init <- c(H = 1, S1 = 0, S2 = 0, D = 0) # initial state vector
v_s_init
```

```
##  H S1 S2  D 
##  1  0  0  0
```

The variable `v_s_init` is used to initialize M represented by `m_M`.

```r
## Initialize cohort trace
m_M <- matrix(0,
              nrow = (n_t + 1), ncol = n_states,
              dimnames = list(0:n_t, v_n))
# Store the initial state vector in the first row of the cohort trace
m_M[1, ] <- v_s_init
```

Note that the initial state vector, `v_s_init`, can be modified to account the distribution of the cohort across the states at the start of the simulation.

4.1 Time-independent cSTM

In a time-independent cSTM, transition probabilities are constant (i.e., do not vary over time). Thus,

$$P_t = P = \begin{bmatrix} p_{[1,1]} & p_{[1,2]} & \cdots & p_{[1,n_s]} \\ p_{[2,1]} & p_{[2,2]} & \cdots & p_{[2,n_s]} \\ \vdots & \vdots & \ddots & \vdots \\ p_{[n_s,1]} & p_{[n_s,2]} & \cdots & p_{[n_s,n_s]} \end{bmatrix} \text{ for } t = 0, \ldots, n_t,$$

and $\mathbf{m}_{t+1} = \mathbf{m}_t P$.

Since the Sick-Sicker model consists of 4 states, we create a 4 × 4 transition probability matrix, `m_P`. We initialize the matrix with default values of zero for all transition probabilities and then populate it with the corresponding transition probabilities. To access an element of `m_P`, we specify first the row number (or name) and then the column number (or name) separated by a comma. For example, the transition probability of going from state Healthy (H) to state Sick (S1) could be accessed by `m_P[1, 2]` or by using the corresponding row/column state names as characters `m_P["H", "S1"]`.

```r
## Initialize transition probability matrix
m_P <- matrix(0,
              nrow = n_states, ncol = n_states,
              dimnames = list(v_n, v_n)) # define row and column names
```

```
## Fill in matrix
# From H
m_P["H", "H"]   <- 1 - (p_HS1 + p_HD)
m_P["H", "S1"]  <- p_HS1
m_P["H", "D"]   <- p_HD
# From S1
m_P["S1", "H"]  <- p_S1H
m_P["S1", "S1"] <- 1 - (p_S1H + p_S1S2 + p_S1D)
m_P["S1", "S2"] <- p_S1S2
m_P["S1", "D"]  <- p_S1D
# From S2
m_P["S2", "S2"] <- 1 - p_S2D
m_P["S2", "D"]  <- p_S2D
# From D
m_P["D", "D"]   <- 1
```

Next, to obtain the cohort distribution across the 4 states over 85 cycles using a time-independent cSTM, we iteratively compute the inner product between `m_M` and `m_P` using the inner product `%*%` symbol in R at each cycle using a `for` loop

```
# Iterative solution of time-independent cSTM
for(t in 1:n_t){
  m_M[t + 1, ] <- m_M[t, ] %*% m_P
}
```

Table 3 shows the cohort trace matrix of the Sick-Sicker model for the first six cycles. The whole cohort starts in the Healthy state and transitions to the rest of the health states over time. Given that the Dead state is an absorbing state, the proportion in this state increases over time. A graphical representation of the cohort trace for all the cycles is shown in Figure 2.

Table 3: First six cycles of the cohort trace of the time-independent Sick-Sicker model

Cycle	H	S1	S2	D
0	1.000	0.000	0.000	0.000
1	0.848	0.150	0.000	0.002
2	0.794	0.186	0.016	0.005
3	0.766	0.191	0.035	0.008
4	0.745	0.189	0.054	0.011
5	0.727	0.185	0.073	0.015

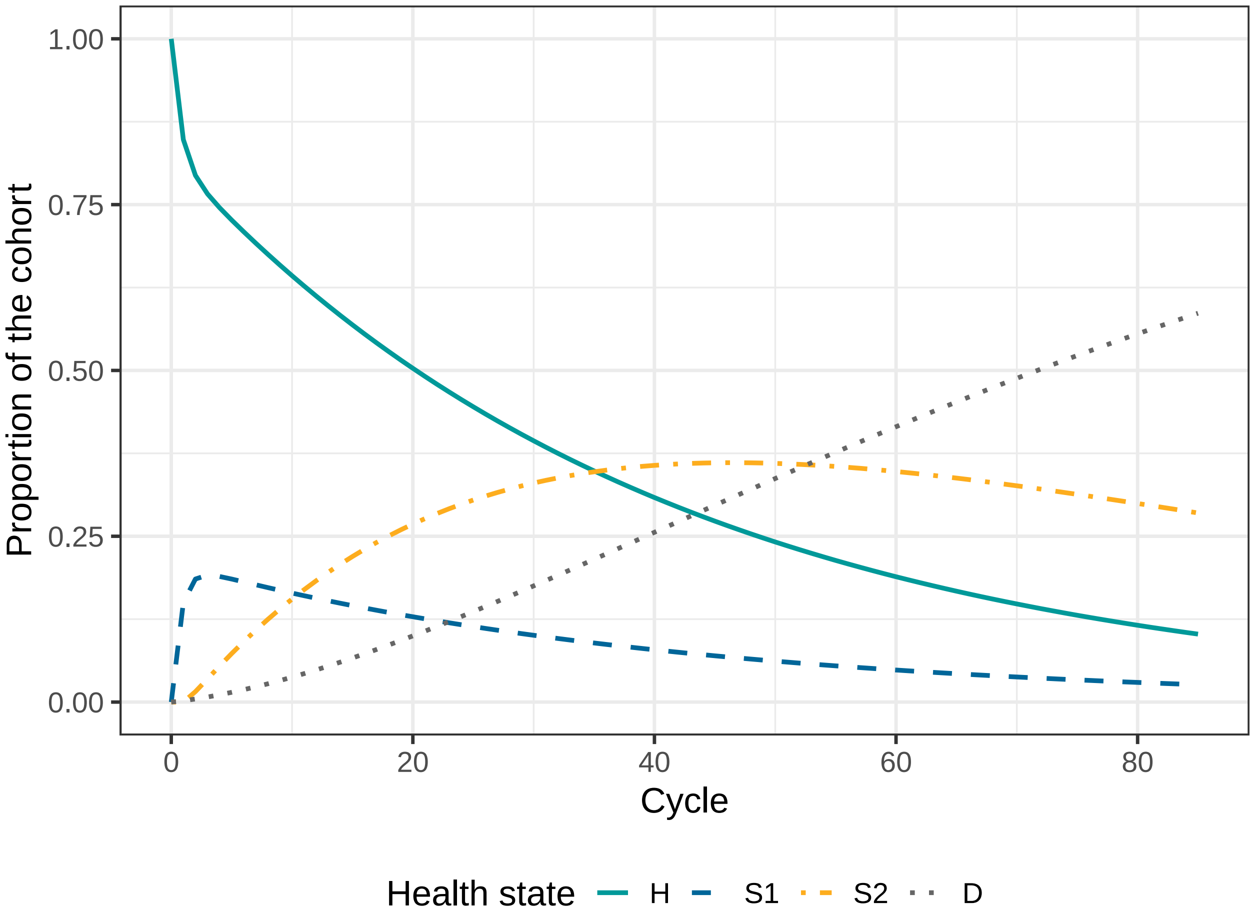


Figure 2: Cohort trace of the time-independent cSTM.

4.2 Time-dependent cSTM

Time-dependency in cSTMs means that transition probabilities vary over time (i.e., the probability of any state transition is dependent of time). Time-dependency in the transition probabilities of cSTMs is a frequently encountered scenario and it can occur in at least two forms: (1) dependence on the time since model start (e.g., age-specific background mortality) in at least one of the health states and (2) dependence on the duration spent in a state (i.e., state-residence). In the next sections, we describe both types of time-dependency and provide a description of their implementation in R.

4.2.1 Time dependency since model start

Transition probabilities that depend on the duration since model start are accounted by the transition probability matrix being a function of time, P_t. To illustrate the implementation of time-dependency cince model start in the Sick-Sicker cSTM, we assume that all-cause mortality is a function of age. It is common to obtain all-cause mortality from life tables in the form of age-specific mortality rates, $\mu(a)$, where a refers to age. For this example, we create a vector `v_r_mort_by_age` of length n_t with the age-specific background mortality rates obtained from the 2015 US life-tables[17], corresponding to the age of the cohort at each cycle.

To compute the transition probability from Healthy to Dead for the Sick-Sicker model, we need to transform $\mu(a)$ to a transition probability assuming constant exponential rate within each year of age

$$p_{[H,D,t]} = 1 - \exp\left[-\mu(a_0 + t)\right],$$

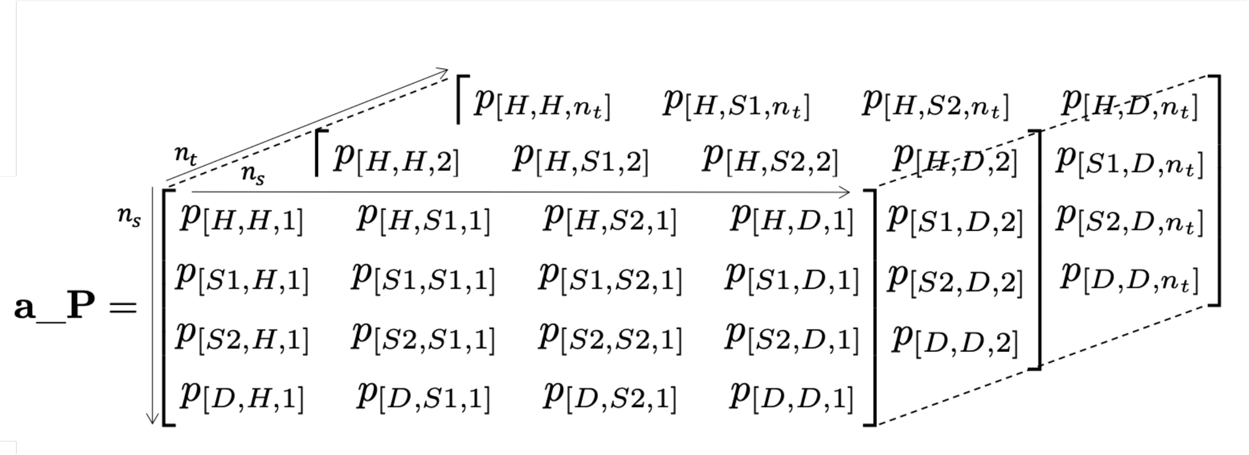
where $a_0 = 25$ is the starting age of the cohort. We will run the Sick-Sicker model for 85 cycles; therefore, we select the background mortality rates for ages 25 to 110 by indexing `v_r_mort_by_age`. Thus, the variable in R is constructed as

```
# Age-specific mortality risk in the Healthy state (background mortality)
v_p_HDage  <- 1 - exp(-v_r_mort_by_age[(n_age_init + 1) + 0:(n_t - 1)])
```

Because mortality in the Sick and Sicker health states are defined relative to the background mortality, adding age dependency on background mortality results in age dependent mortality in all health states. To generate the age-specific mortality for all cycles, we multiply the age-specific background mortality rate, `v_r_mort_by_age`, by the constant hazard ratios `hr_S1` and `hr_S2`, respectively, and then convert the resulting stage-specific mortality rates to probabilities as described previously.

```
# Age-specific mortality risk in the Sick state
v_p_S1Dage <- 1 - exp(-v_r_mort_by_age[(n_age_init + 1) + 0:(n_t - 1)] * hr_S1)
# Age-specific mortality risk in the Sicker state
v_p_S2Dage <- 1 - exp(-v_r_mort_by_age[(n_age_init + 1) + 0:(n_t - 1)] * hr_S2)
```

One way to incorporate age-dependency into the transition probability matrix, P_t, is to expand the dimensions of the matrix and create a 3-dimensional transition probability array, `a_P`, of dimensions $n_s \times n_s \times n_t$, where the first two dimensions correspond to transitions between states and the third dimension to time. That is, the t-th element in the third dimension corresponds to the transition probability matrix at cycle t. A visual representation of `a_P` is shown in Figure 3.



a_P =
n_t
n_s
n_s
$p_{[H,H,n_t]}$ $p_{[H,S1,n_t]}$ $p_{[H,S2,n_t]}$ $p_{[H,D,n_t]}$
$p_{[S1,D,n_t]}$
$p_{[S2,D,n_t]}$
$p_{[D,D,n_t]}$
$p_{[H,H,2]}$ $p_{[H,S1,2]}$ $p_{[H,S2,2]}$ $p_{[H,D,2]}$
$p_{[S1,D,2]}$
$p_{[S2,D,2]}$
$p_{[D,D,2]}$
$p_{[H,H,1]}$ $p_{[H,S1,1]}$ $p_{[H,S2,1]}$ $p_{[H,D,1]}$
$p_{[S1,H,1]}$ $p_{[S1,S1,1]}$ $p_{[S1,S2,1]}$ $p_{[S1,D,1]}$
$p_{[S2,H,1]}$ $p_{[S2,S1,1]}$ $p_{[S2,S2,1]}$ $p_{[S2,D,1]}$
$p_{[D,H,1]}$ $p_{[D,S1,1]}$ $p_{[D,S2,1]}$ $p_{[D,D,1]}$

Figure 3: A 3-dimensional array of the time-dependent Sick-Sicker model where transition probabilities depend on time since model start.

First, let's initialize `a_P` with a default value of zero for all transition probabilities.

```
# Initialize the transition probability array
a_P <- array(0, dim = c(n_states, n_states, n_t),
              dimnames = list(v_n, v_n, 0:(n_t - 1)))
```

Filling `a_P` with the corresponding transition probabilities is comparable with filling `m_P` above, with the difference being that we now refer to all the cycles by leaving the index of the third dimension blank. This operation stores the age-dependent transition probabilities in the third dimension of the array. For those transitions that are constant over time, only one transition probability is provided and R replicates the value of such transition as many times as the length of the dimension of time, for our example $n_t + 1$ times.

```
### Fill in array
## From H
a_P["H", "H", ]   <- 1 - (p_HS1 + v_p_HDage)
a_P["H", "S1", ]  <- p_HS1
a_P["H", "D", ]   <- v_p_HDage
## From S1
a_P["S1", "H", ]  <- p_S1H
a_P["S1", "S1", ] <- 1 - (p_S1H + p_S1S2 + v_p_S1Dage)
a_P["S1", "S2", ] <- p_S1S2
a_P["S1", "D", ]  <- v_p_S1Dage
## From S2
a_P["S2", "S2", ] <- 1 - v_p_S2Dage
a_P["S2", "D", ]  <- v_p_S2Dage
## From D
a_P["D", "D", ]   <- 1
```

As mentioned before, each of the elements on the third dimension of `a_P` correspond to a transition probability matrix. For example, the transition matrix for 25-year-olds in the Sick-Sicker model is shown below

```
a_P[, , 1]
```

```
##             H        S1        S2           D
## H  0.8489865 0.1500000 0.0000000 0.001013486
## S1 0.5000000 0.3919626 0.1050000 0.003037378
## S2 0.0000000 0.0000000 0.9899112 0.010088764
## D  0.0000000 0.0000000 0.0000000 1.000000000
```

To simulate the cohort over the n_t cycles for the age-dependent cSTM, we initialize a new cohort trace matrix `m_M_ad`

```
## Initialize cohort trace for age-dependent (ad) cSTM
m_M_ad <- matrix(0,
                 nrow = (n_t + 1), ncol = n_states,
                 dimnames = list(0:n_t, v_n))
m_M_ad[1, ] <- v_s_init
```

and then use the inner product to get the state vector at cycle t. This is similar to the time-independent model but we now also index the transition probability array `a_P` by t to obtain the cycle-specific transition probability matrix.

```
# Iterative solution of age-dependent cSTM
for(t in 1:n_t){
  m_M_ad[t + 1, ] <- m_M_ad[t, ] %*% a_P[, , t]
}
```

A graphical representation of the cohort trace for all cycles of the age-dependent cSTM is shown in Figure 4.

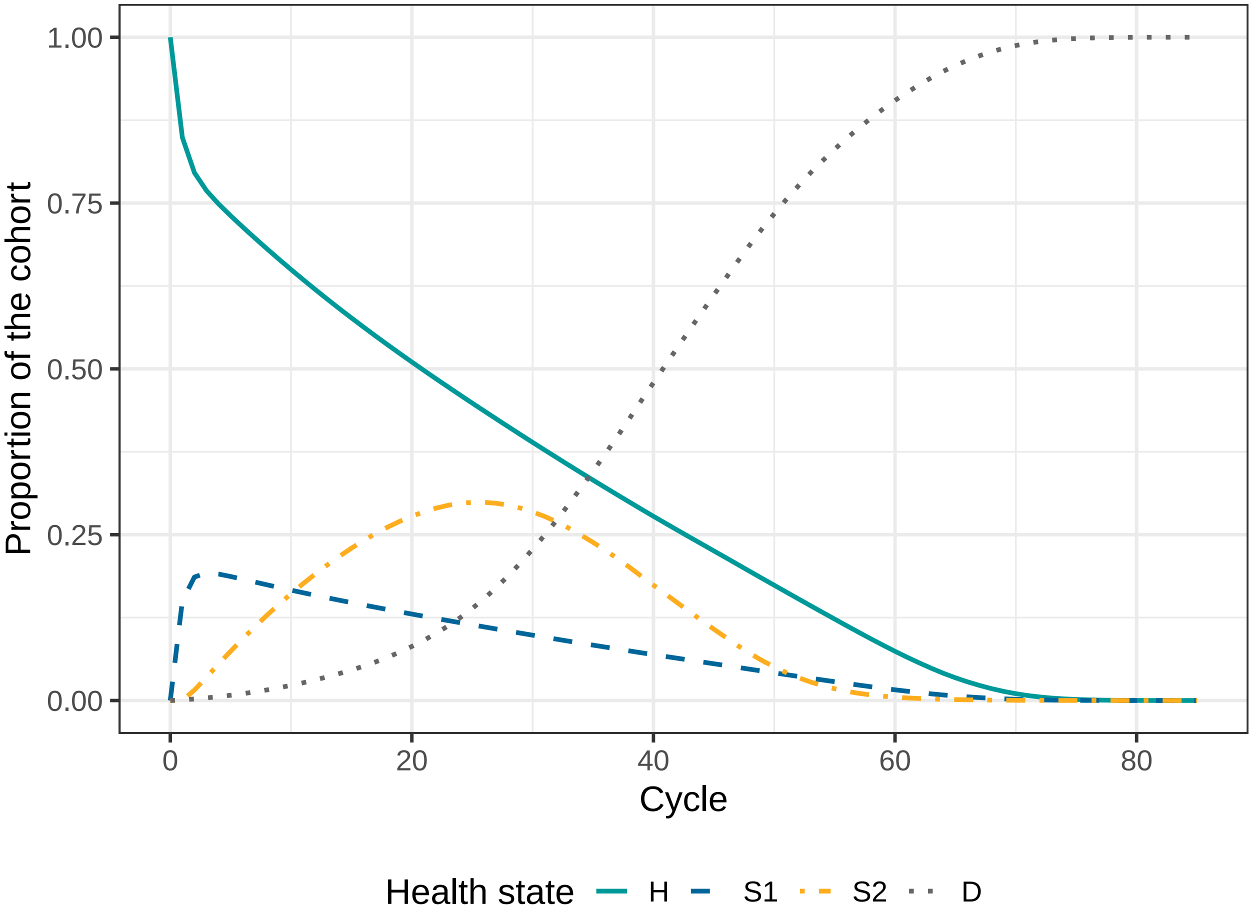


Figure 4: Cohort trace of the age-dependent cSTM.

4.2.2 Time dependency based on state residence

Transition probabilities can vary by the length of time spent in a given state, which we call *state-residency* dependence. If time has an effect on the probability of transition from an initial state to any other state then the solution is similar to the section above. However, if probabilities depend on residence in any of the intermediate states, cSTM cannot easily account for such time dependency without a modification of the state space. One way to account for dependency on the time spent in a state, is to expand the number of states as much as needed. This expansion requires a series of transient states, referred to as *tunnel* states, where the cohort can only stay for a given number of cycles. The set of tunnel states then essentially counts the amount of time that an individual has spent in the tunnel representing a specific health state (up to some pre-defined maximum).

When addressing dependency on state residence with tunnels states, we produce as many states as time-dependency is desired. For example, if dependency in a given state is assumed for T cycles, such state needs to be expanded T times. This will generate an expanded transition probability matrix (or array if time since model start matters too) of dimensions $n_{s_T} \times n_{s_T}$ (or $n_{s_T} \times n_{s_T} \times n_t$), where n_{s_T} is the total number of health states after accounting for the expansion of a given state for T tunnel states. Dependency to state residence could occur for as many as $n_t - 1$ cycles.

Figure 5 shows the state-transition diagram of the Sick-Sicker model that includes state residence dependency with T tunnel states for S1.

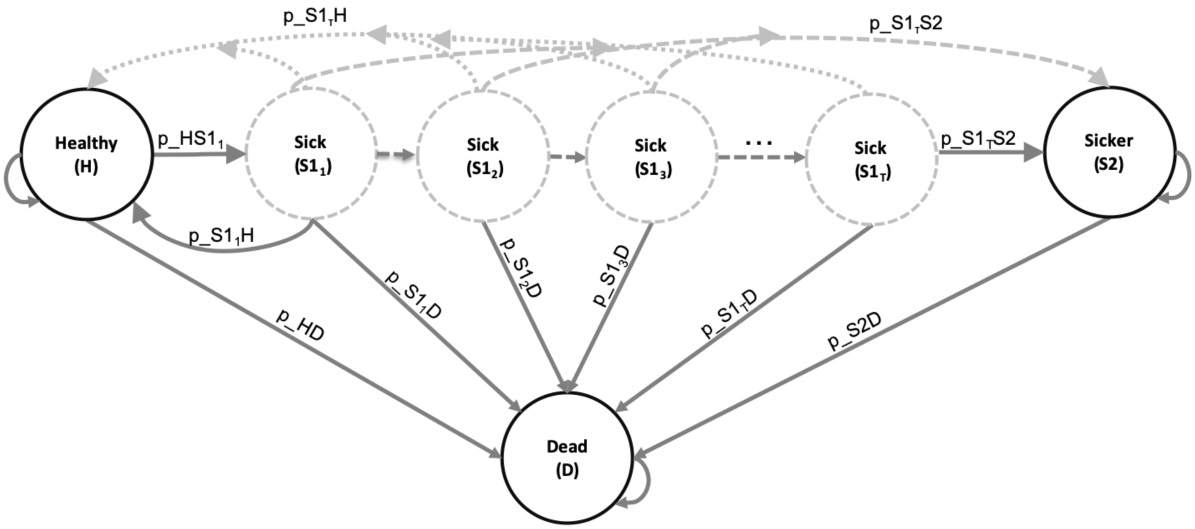


Figure 5: State-transition diagram of the Sick-Sicker model with tunnel states expanding the Sick state $(S1_1, S1_2, ..., S1_\tau)$.

To illustrate state residence dependency in the age-dependent Sick-Sicker model defined above, we assumed that the risk of progression from Sick to Sicker increases the longer a person has been sick. This increase follows a Weibull growth curve, defined as

$$p_{[S1_\tau, S2, t]} = \lambda\gamma\tau^{(\lambda-1)},$$

where $\tau = 1, \ldots, T$ is the τ-th cycle (year) that a person has been in the Sick state, and λ and γ are the scale and shape parameters of the Weibull function, respectively. We assume that state residence dependency occurs throughout the whole simulation (i.e., $T = n_t$) and create a new variable called `n_tunnel_size` with the length of the tunnel equal to `n_t`. Thus, there will be 85 S1 tunnel states plus 3 more states (H, S2, D) resulting in a total of $n_{s_T} = 88$.

To implement the state residence dependency in the Sick-Sicker model using tunnels, we create four new parameters: (1) `v_Sick_tunnel`, a variable with the tunnel names of the Sick state, (2) `v_n_tunnels`, a vector with state names including tunnel states, (3) `n_states_tunnels`, a variable with the number of states including the expanded names for the tunnel states, and (4) `v_s_init_tunnels`, an initial state vector for the model with tunnels that it represents. Table 4 shows the extra R variables for state-residence-dependent Sick-Sicker model with tunnels.

Table 4: Extra R variables for the state-residence-dependent Sick-Sicker model with tunnels

R name	Description
`n_tunnel_size`	number of tunnel states
`v_Sick_tunnel`	vector with the the tunnel names of the Sick state
`v_n_tunnels`	vector with state names including tunnel states
`n_states_tunnels`	number of states including tunnel states
`v_s_init_tunnels`	initial state vector for the model with tunnels

```r
## Number of tunnels
n_tunnel_size <- n_t
## Name for tunnels states of Sick state
v_Sick_tunnel <- paste("S1_", seq(1, n_tunnel_size), "Yr", sep = "")
## Create variables for model with tunnels
v_n_tunnels <- c("H", v_Sick_tunnel, "S2", "D") # state names
n_states_tunnels <- length(v_n_tunnels)                   # number of states
## Initialize first cycle of Markov trace accounting for the tunnels
v_s_init_tunnels <- c(1, rep(0, n_tunnel_size), 0, 0)
```

Based on the updated parameters, the state-residence-dependent transition probability from Sick to Sicker based on a Weibull function, `p_S1S2_tunnels`, is:

```r
# Weibull parameters
l <- 0.08 # scale
g <- 1.1  # shape
# Weibull function
p_S1S2_tunnels <- l * g * (1:n_tunnel_size)^{g-1}
```

To implement the transition probability array for the age- and state-residence-dependent Sick-Sicker model, we create an expanded 3-dimensional array accounting for tunnels `a_P_tunnels` of dimensions $n_{s_T} \times n_{s_T} \times n_t$, where $n_{s_T} = 88$, is the number of states of the Sick-Sicker model with tunnels. A visual representation of `a_P_tunnels` of the Sick-Sicker model with tunnel states expanding the Sick state is shown in Figure 6.

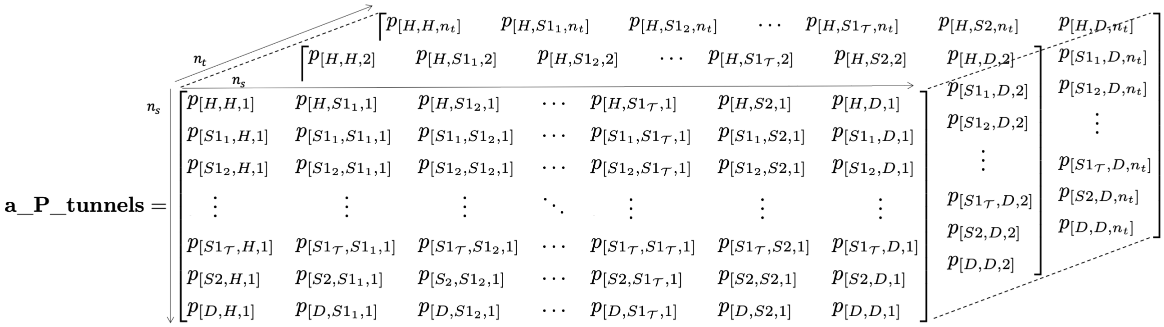


Figure 6: The 3-dimensional array of the age- and state-residence-dependent Sick-Sicker model with tunnel states expanding the Sick state.

```r
# Initialize array
a_P_tunnels <- array(0, dim = c(n_states_tunnels, n_states_tunnels, n_t),
                     dimnames = list(v_n_tunnels, v_n_tunnels, 0:(n_t - 1)))
```

Filling `a_P_tunnels` with the corresponding transition probabilities is similar to the `a_P` above, with the difference being that we now fill the transition probabilities from all tunnel states of the Sick state by iterating through all tunnel states and assigning the corresponding disease progression transition probability.

```r
### Fill in array
## From H
a_P_tunnels["H", "H", ]              <- 1 - (p_HS1 + v_p_HDage)
a_P_tunnels["H", v_Sick_tunnel[1], ] <- p_HS1
a_P_tunnels["H", "D", ]              <- v_p_HDage
## From S1
for(i in 1:(n_tunnel_size - 1)){
  a_P_tunnels[v_Sick_tunnel[i], "H", ]  <- p_S1H
  a_P_tunnels[v_Sick_tunnel[i],
              v_Sick_tunnel[i + 1], ]   <- 1 - (p_S1H + p_S1S2_tunnels[i] + v_p_S1Dage)
  a_P_tunnels[v_Sick_tunnel[i], "S2", ] <- p_S1S2_tunnels[i]
  a_P_tunnels[v_Sick_tunnel[i], "D", ]  <- v_p_S1Dage
}
## repeat code for the last cycle to force the cohort stay in the last tunnel stateof Sick
a_P_tunnels[v_Sick_tunnel[n_tunnel_size], "H", ]  <- p_S1H
a_P_tunnels[v_Sick_tunnel[n_tunnel_size],
            v_Sick_tunnel[n_tunnel_size], ]       <- 1 - (p_S1H +
                                                            p_S1S2_tunnels[n_tunnel_size] +
                                                            v_p_S1Dage)
a_P_tunnels[v_Sick_tunnel[n_tunnel_size], "S2", ] <- p_S1S2_tunnels[n_tunnel_size]
a_P_tunnels[v_Sick_tunnel[n_tunnel_size], "D", ]  <- v_p_S1Dage
### From S2
a_P_tunnels["S2", "S2", ] <- 1 - v_p_S2Dage
```

```
a_P_tunnels["S2", "D", ]  <- v_p_S2Dage
# From D
a_P_tunnels["D", "D", ] <- 1
```

To simulate the cohort over the n_t cycles for the time-dependent cSTM with tunnels, we initialize a new cohort trace matrix m_M_tunnels of dimensions $n_t \times n_{S_T}$

```
# Initialize cohort for state-residence-dependent cSTM
m_M_tunnels <- matrix(0,
                      nrow = (n_t + 1), ncol = n_states_tunnels,
                      dimnames = list(0:n_t, v_n_tunnels))
m_M_tunnels[1, ] <- v_s_init_tunnels
```

and then calculate inner product similar to the age-dependent model to generate the full cohort trace.

```
# Iterative solution of state-residence-dependent cSTM
for(t in 1:n_t){
  m_M_tunnels[t + 1, ] <- m_M_tunnels[t, ] %*% a_P_tunnels[, , t]
}
```

To compute a summarized cohort trace comparable with the time-independent and age-dependent cSTM, we aggregate over the tunnel states of the Sicker state for all cycles (Figure 7).

```
# Create aggregated trace
m_M_tunnels_sum <- cbind(H = m_M_tunnels[, "H"],
                       S1 = rowSums(m_M_tunnels[, 2:(n_tunnel_size +1)]),
                       S2 = m_M_tunnels[, "S2"],
                       D = m_M_tunnels[, "D"])
```

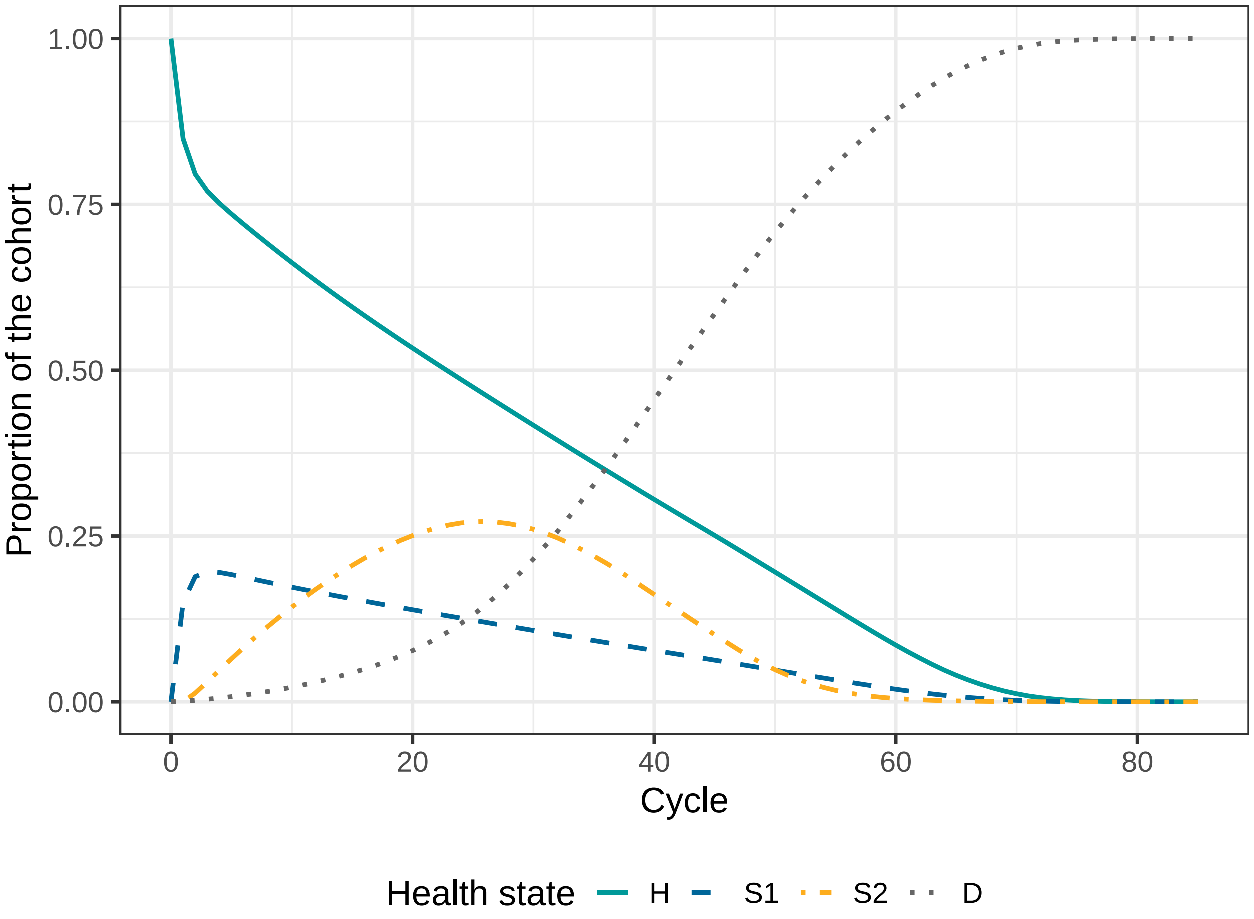


Figure 7: Cohort trace of the age- and state-residence-dependent cSTM.

5 Epidemiological and economic outputs

cSTMs can be used to generate different epidemiological and economic outputs. In CEA, the final outcomes are typically the total expected QALYs and costs accrued in the cohort over the chosen time horizon. However, epidemiological outcomes are often used for different purposes such as calibration and validation. Some common epidemiological outcomes include survival, prevalence, incidence, average number of events and lifetime risk events.[3] Below, we provide the epidemiological definition of some of these outcomes and how they can be calculated from a cSTM using the age-dependent Sick-Sicker cSTM as an example.

5.1 Epidemiological outcomes

5.1.1 Survival curve

The survival curve, $S(t)$, is defined as the proportion of the population alive at cycle t. To estimate $S(t)$ from the simulated cohort of the age-dependent Sick-Sicker model, we sum the proportions of the non-death states for all n_t cycles in `m_M_ad`.

```
v_S_ad <- rowSums(m_M_ad[, -4]) # vector with survival curve
```

and is shown in Figure 8.

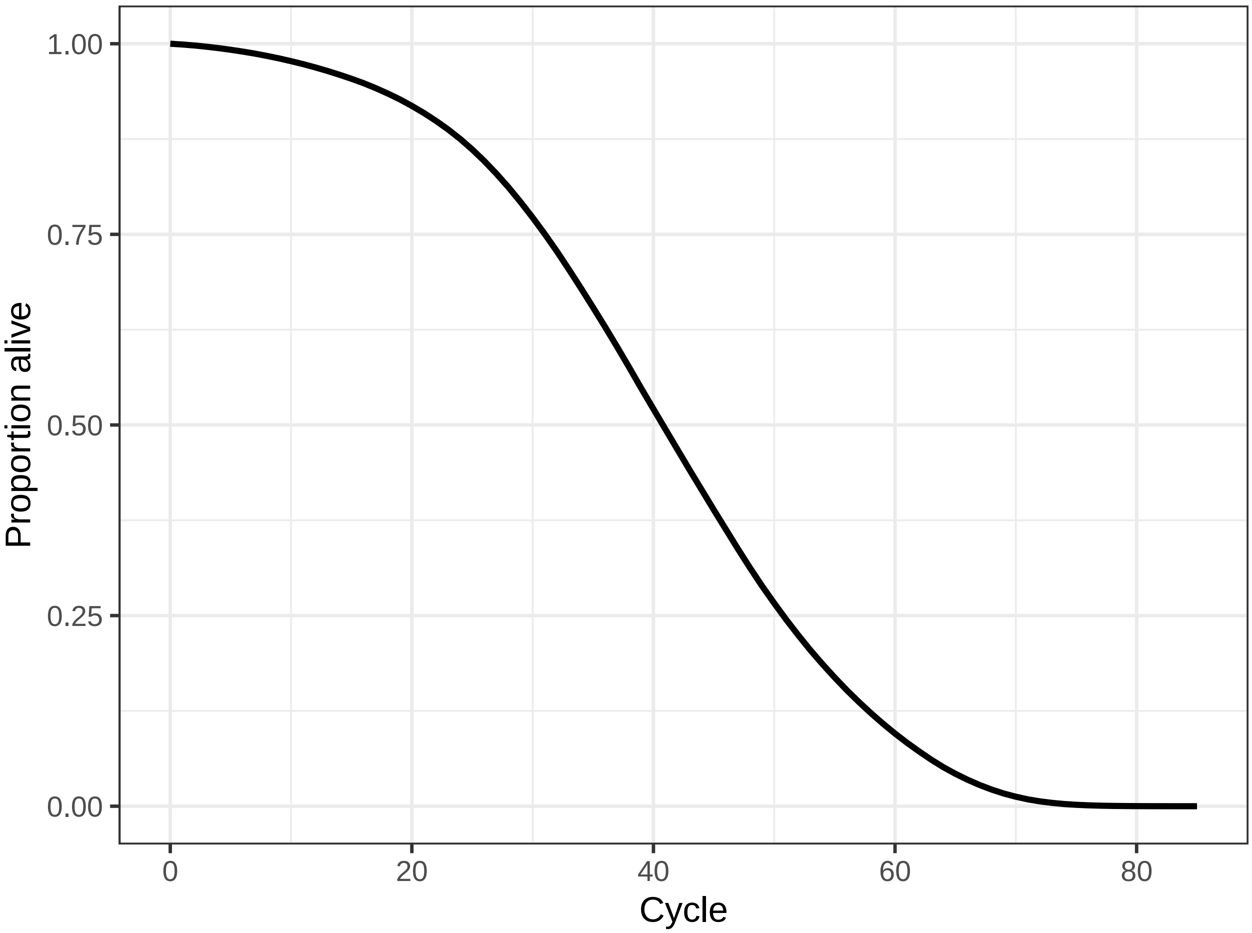


Figure 8: Survival curve of time-dependent cSTM.

5.1.2 Prevalence

Prevalence is defined as the proportion of those having a specific condition (or being in a specific health state) out of those in the living population at risk of experiencing the condition of interest.[18] To calculate the prevalence of S1 at cycle t, $\text{prev}(t)_i$, we compute the ratio between the proportion of the cohort in S1 and the proportion alive at that cycle.[19] The proportion of the cohort alive is given by the survival curve $S(t)$ defined above. The individual prevalence of the S1 and S2 health states, and the overall prevalence of sick individuals (i.e., S1 and S2) of the age-dependent Sick-Sicker cSTM at each cycle t is computed as follows and are shown in Figure 9.

```
v_prev_S1   <- m_M_ad[, "S1"] / v_S_ad          # vector with prevalence of Sick
v_prev_S2   <- m_M_ad[, "S2"] / v_S_ad          # vector with prevalence of Sicker
v_prev_S1S2 <- rowSums(m_M_ad[, c("S1", "S2")])/v_S_ad # prevalence of Sick and Sicker
```

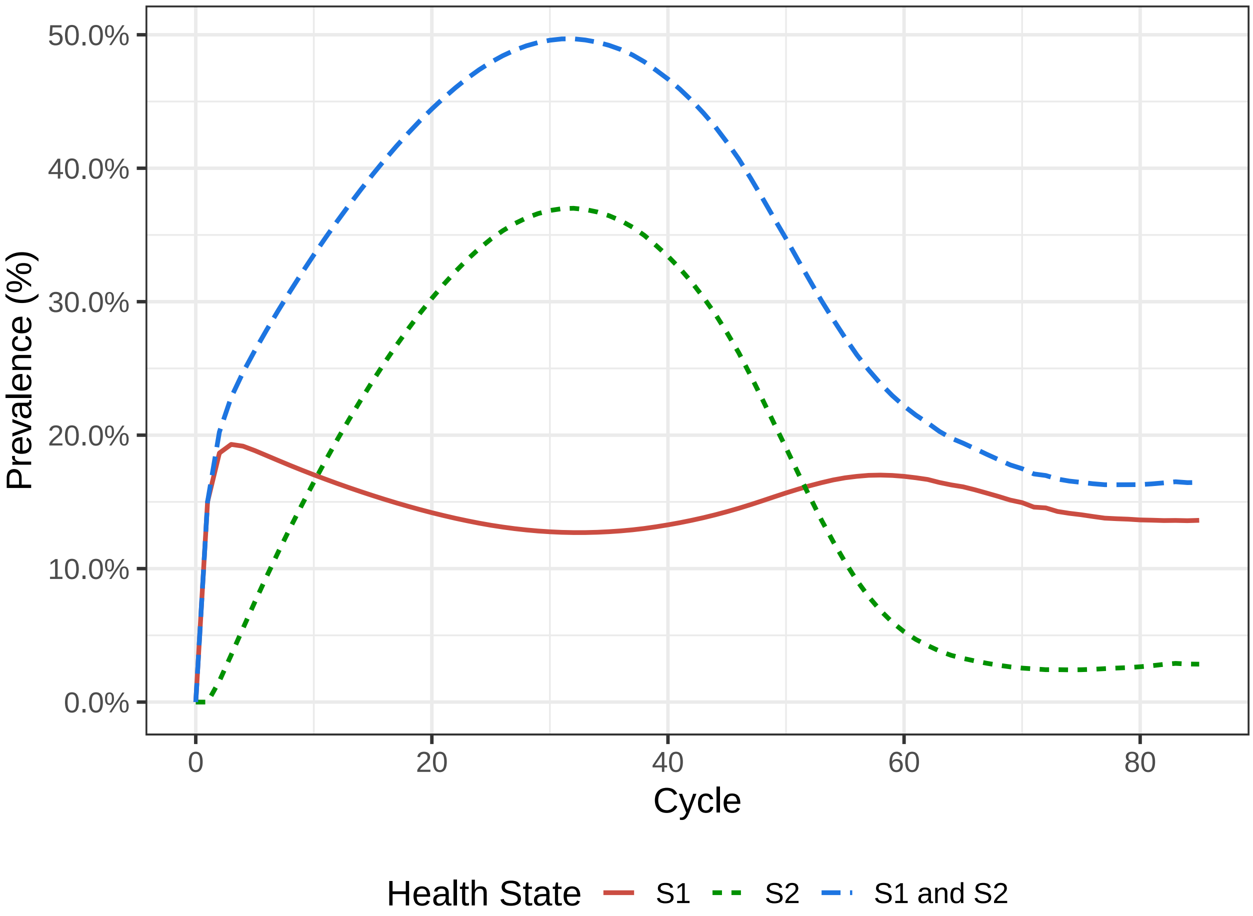


Figure 9: Prevalence of sick states of age-dependent cSTM.

Another epidemiological outcome that could be of interest, is the proportion of Sicker (i.e., S2) among the sick individuals. To compute the proportion of S2 among all sick individuals at each cycle t, we divide $m_{[S2,t]}$ by $\text{prev}(t)_{\{S1,S2\}}$ where $t > 0$, as presented in Figure 10. Note that t does not start at 0 because it takes one cycle for the cohort to get sick.

```
# Vector with proportion of Sicker among sick individuals
v_prop_S2 <- m_M_ad[-1, "S2"] / v_prev_S1S2[-1]
```

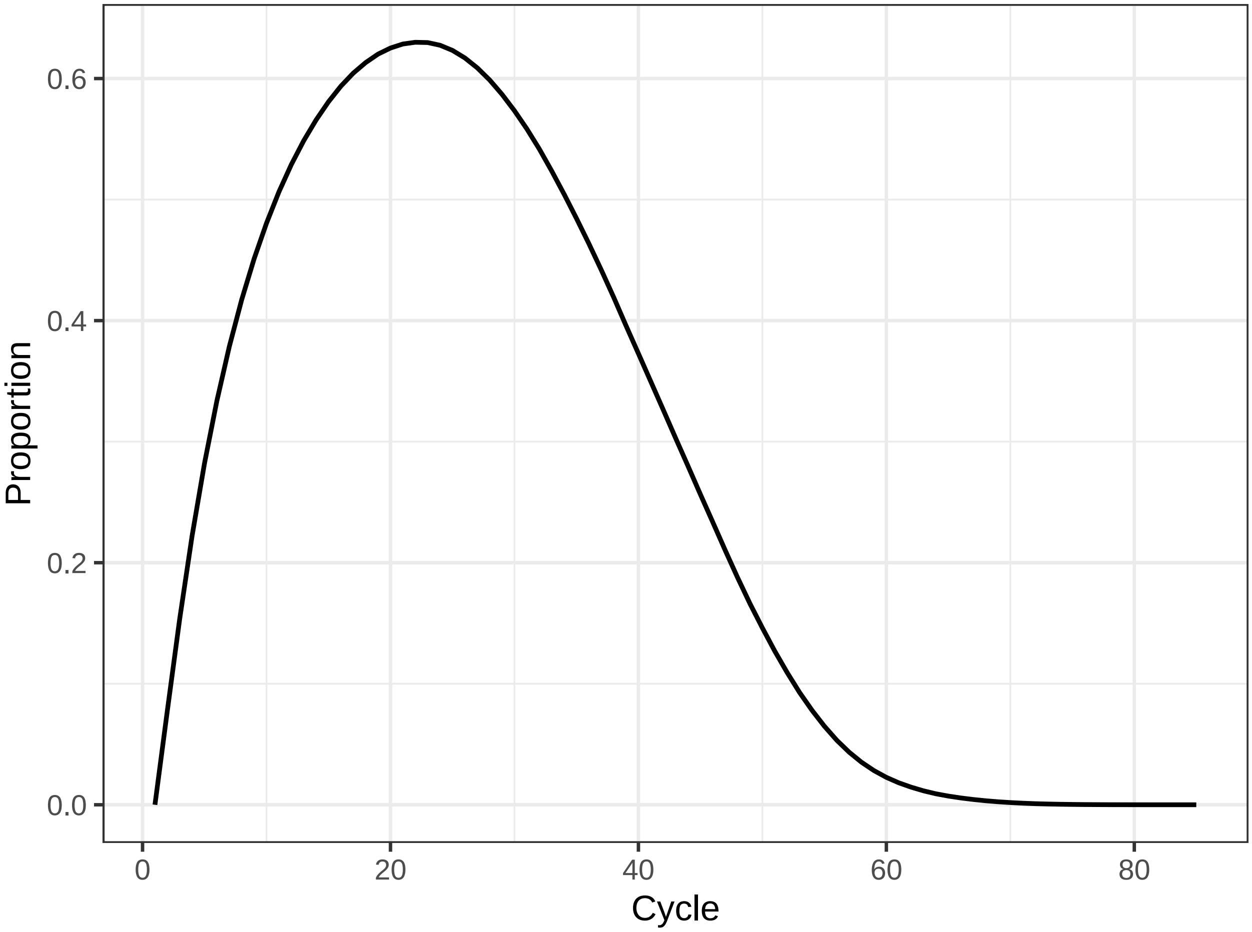


Figure 10: Proportion of Sicker (S2) individuals among all sick patients of age-dependent cSTM.

5.1.3 Life expectancy

Life expectancy (LE) refers to the expected number of years remaining to be lived.[20] LE is the area under the survival curve $S(t)$.[21] In discrete time, LE is calculated as

$$LE = \sum_{t=0}^{\infty} S(t).$$

In the age-dependent Sick-Sicker model, where we simulated a cohort over 85 cycles, life expectancy `le_ad` is 41.1, which is calculated as follows

```
le_ad <- sum(v_S_ad) # life expectancy
```

6 State and transition rewards

6.1 State rewards

A state reward refers to a value assigned to individuals for being in a given state. In a cost-utility context, these could be either utilities or costs associated with remaining in a certain health state for one cycle. The total expected reward of an outcome of interest for the entire cohort at each cycle can be stored in the column vector $\mathbf{y}$ of size $(n_t + 1)$. To calculate $\mathbf{y}$, we simply multiply the cohort trace matrix times a *vector* of state rewards $\mathbf{r}$ of the same dimension as the number of states (n_s)

$$\mathbf{y} = M\mathbf{r}. \tag{1}$$

Note that rewards can also be time- or age-dependent; in such cases, the vector of state rewards will be time-dependent, $\mathbf{r}_t$. For the Sick-Sicker model, we create a vector of utilities and costs under usual care, `v_u_UC` and `v_c_UC`, respectively. Each of these vectors contain the utilities and costs corresponding with being in each of the four health states under usual care, which are shown in Table 1.

```r
# Vector of state utilities under usual care
v_u_UC <- c(H = u_H, S1 = u_S1, S2 = u_S2, D = u_D)
# Vector of state costs under usual care
v_c_UC <- c(H = c_H, S1 = c_S1, S2 = c_S2, D = c_D)
```

To create the state-reward vectors under the new treatment, we account for the benefits and costs of the new treatment. For the vector of utilities under new treatment, `v_u_Tr`, we substitute the utility of being in state S1 under usual care, `u_S1`, with the utility associated to the benefit of the treatment in being in that state, `u_Trt`.

```r
# Vector of state utilities under new treatment
v_u_Tr <- c(H = u_H, S1 = u_Trt, S2 = u_S2, D = u_D)
```

To create the vector of state costs under the new treatment, `v_c_Tr`, we add the cost of treatment, `c_Trt`, to the state costs of S1 and S2.

```r
# Vector of state costs under new treatment
v_c_Tr <- c(H = c_H, S1 = c_S1 + c_Trt, S2 = c_S2 + c_Trt, D = c_D)
```

To compute the expected QALYs and costs for the age-dependent Sick-Sicker model under usual care and new treatment for all cycles, we apply Eq. (1) by multiplying the cohort trace matrix, `m_M_ad`, times these vectors.

```r
# Vector of QALYs under usual care
v_qaly_UC <- m_M_ad %*% v_u_UC
# Vector of costs under usual care
v_cost_UC <- m_M_ad %*% v_c_UC
# Vector of QALYs under new treatment
v_qaly_Trt <- m_M_ad %*% v_u_Tr
# Vector of costs under new treatment
```

```r
v_cost_Trt <- m_M_ad %*% v_c_Tr
```

To account for discounting, the generated rewards per cycle are multiplied by the cycle specific discount weight. The vector of expected rewards, $\mathbf{y}$, is multiplied by a discounting column vector $\mathbf{d}$ of size $n_t + 1$ where each of its t-th entry represents the discounting for cycle t

$$\mathbf{d} = \left[0, \frac{1}{(1+d)^1}, \frac{1}{(1+d)^2}, \cdots, \frac{1}{(1+d)^{n_t}}\right],$$

where d is the cycle-length discount rate. Therefore, the total expected discounted outcome over the n_t cycles, y, is obtained by the inner product between $\mathbf{y}$ transposed, $\mathbf{y}^{'}$, and $\mathbf{d}$,

$$y = \mathbf{y}^{'}\mathbf{d}. \tag{2}$$

The discount vectors for costs and QALYs for the Sick-Sicker model, `v_dwc` and `v_dwe`, respectively, are

```r
# Discount weight for effects
v_dwe <- 1 / ((1 + d_e) ^ (0:(n_t)))
# Discount weight for costs
v_dwc <- 1 / ((1 + d_c) ^ (0:(n_t)))
```

To compute the total expected discounted QALYs and costs under usual care and the new treatment, we apply (2) to the vectors with the rewards and the vectors of discounts

```r
## Expected QALYs under usual care
qaly_UC <- t(v_qaly_UC) %*% v_dwe
## Expected costs under usual care
cost_UC <- t(v_cost_UC) %*% v_dwc
## Expected QALYs under new treatment
qaly_Trt <- t(v_qaly_Trt) %*% v_dwe
## Expected costs under new treatment
cost_Trt <- t(v_cost_Trt) %*% v_dwc
```

Table 5: Total expected discounted QALYs and costs per average individual in the cohort of the age-dependent Sick-Sicker model under usual care and the new treatment

	Costs	QALYs
Usual care	$113,573	19.981
New treatment	$211,025	20.679

The total expected QALYs and costs for the Sick-Sicker model under usual care and the new treatment are shown in Table 5. The total expected QALYs for usual care and the new treatment are 19.981 and 20.679, respectively. The total expected Costs for usual care and the new treatment are $113,573 and $211,025, respectively.

6.2 Transition rewards

In addition to the state rewards (e.g., costs and utilities associated with residing in a given state), a transition from one state to another may in of itself be associated with a one-time cost or utility impact, termed a "transition reward". For example, in the Sick-Sicker model, we previously mentioned that dying (i.e., transitioning to the Dead state) incurs a one-time cost of $2,000 that reflects the acute care that might be received immediately preceding death. This could include emergency services, hospitalization, or other healthcare utilization to address the ultimately fatal health complication. We also include a utility decrement and a cost increment on the transition from the Healthy to Sick states. Whereas the cost and utility of the Sick state reflects the ongoing cost and utility of being chronically sick, transition rewards capture the short-term impact of the acute events of becoming sick, such as hospitalization, stabilization, and so on.

To include transition rewards on the calculation of the total expected reward of an outcome of interest, $\mathbf{y}$, we replace the *vector* of state rewards $\mathbf{r}$ with a *matrix* of rewards R_t of dimensions $n_s \times n_s$, and the cohort trace matrix, M, with a multidimensional array $\mathbf{A}$, of dimensions $n_s \times n_s \times [n_t + 1]$, called the transition-dynamics array.[22] Briefly, $\mathbf{A}$ stores the proportion of the cohort that transitions between any two health states in each cycle over the time horizon, and R_t contains the rewards associated with transitioning between states -including remaining in a state- at cycle t. For a more detailed description of the transition-dynamics array and matrix of rewards, please see Krijkamp et al. (2020)[22].

To compute $\mathbf{A}$ for the age-dependent Sick-Sicker model, we initialize a three-dimensional array `a_A` of dimensions $n_s \times n_s \times [n_t + 1]$ and set the diagonal of the first slice to the initial state vector `v_s_init`.

```r
# Initialize transition array
a_A <- array(0,
             dim = c(n_states, n_states, (n_t + 1)),
             dimnames = list(v_n, v_n, 0:n_t))
# Set first slice of A equal to the initial state vector in its diagonal
diag(a_A[, , 1]) <- v_s_init
```

We then iterate the element by element multiplication between each of the rows of the cohort trace matrix, `m_M_ad`, and the array of transition matrices, `a_P`, for each cycle to estimate all transitions between all states over all n_t cycles.

```r
# Iterative solution to produce the transition array
for (t in 1:n_t){
  a_A[, , t + 1] <- m_M_ad[t, ] * a_P[, , t]
}
```

To create the arrays of rewards for costs and utilities for the Sick-Sicker cSTM, we create strategy-specific arrays of rewards because the new treatment affects both outcomes. We use the `aperm` function as a shortcut to order the rewards such that each of the rows across the third dimension of these arrays are filled with the vector of state rewards.

```r
# Array of state and transition utilities under Usual Care
a_R_u_UC <- aperm(array(v_u_UC,
                        dim = c(n_states, n_states, n_t + 1),
                        dimnames = list(v_n, v_n, 0:n_t)),
                  perm = c(2, 1, 3))
# Array of state and transition costs under Usual Care
a_R_c_UC <- aperm(array(v_c_UC,
                        dim = c(n_states, n_states, n_t + 1),
                        dimnames = list(v_n, v_n, 0:n_t)),
                  perm = c(2, 1, 3))
# Array of utilities under New Treatment
a_R_u_Tr <- aperm(array(v_u_Tr,
                       dim = c(n_states, n_states, n_t + 1),
                       dimnames = list(v_n, v_n, 0:n_t)),
                 perm = c(2, 1, 3))
# Array of costs under New Treatment
a_R_c_Tr <- aperm(array(v_c_Tr,
                       dim = c(n_states, n_states, n_t + 1),
                       dimnames = list(v_n, v_n, 0:n_t)),
                 perm = c(2, 1, 3))
```

To account for the transition rewards, we either add or subtract them in the corresponding location of the reward matrix that represents the transitions of interest. For example, to account for the disutility of transitioning from H to S1 on the Treatment strategy, we subtract the disutility to the entry of the array of rewards corresponding to the transition from H to S1 across all cycles.

```
# Add disutility due to transition from Healthy to Sick
a_R_u_Tr["H", "S1", ] <- a_R_u_Tr["H", "S1", ] - du_HS1
```

In a similar approach, we add the costs of transitioning from H to S1 and the cost of dying for the treatment strategy.

```
# Add transition cost due to transition from Healthy to Sick
a_R_c_Tr["H", "S1", ] <- a_R_c_Tr["H", "S1", ] + ic_HS1
# Add transition cost of dying from all non-dead states
a_R_c_Tr[-n_states, "D", ] <- a_R_c_Tr[-n_states, "D", ] + ic_D
a_R_c_Tr[, , 1]
```

```
##       H    S1    S2    D
## H  2000 17000 27000 2000
## S1 2000 16000 27000 2000
## S2 2000 16000 27000 2000
## D  2000 16000 27000    0
```

Below, we show how the transition rewards are added to the reward matrices corresponding to usual care.

```
# Add disutility due to transition from H to S1
a_R_u_UC["H", "S1", ] <- a_R_u_UC["H", "S1", ] - du_HS1
# Add transition cost due to transition from H to S1
a_R_c_UC["H", "S1", ] <- a_R_c_UC["H", "S1", ] + ic_HS1
# Add transition cost of dying from all non-dead states
```

```
a_R_c_UC[-n_states, "D", ] <- a_R_c_UC[-n_states, "D", ] + ic_D
```

The state and transition rewards are applied to the model dynamics by element-wise multiplication between $\mathbf{A}$ and $\mathbf{R}$, indicated by the $\odot$ sign, which produces the array of outputs for all n_t cycles, $\mathbf{Y}$. Formally,

$$\mathbf{Y} = \mathbf{A} \odot \mathbf{R} \tag{3}$$

To obtain $\mathbf{Y}$ for all the QALYs and costs for both strategies, we apply Equation (3) by multiplying the transition array `a_A` element by element times their corresponding array of rewards

```
# For Usual Care
a_Y_c_UC <- a_A * a_R_c_UC
a_Y_u_UC <- a_A * a_R_u_UC
# For New Treatment
a_Y_c_Tr <- a_A * a_R_c_Tr
a_Y_u_Tr <- a_A * a_R_u_Tr
```

The total rewards for each health state at cycle t, $\mathbf{t}_t$, is obtained by summing the rewards across all $j = 1, ..., n_s$ health states for all n_t cycles.

$$\mathbf{y}_t = \mathbf{1}^T Y_t = \left[\sum_{i=1}^{n_s} Y_{[i,1,t]}, \sum_{i=1}^{n_s} Y_{[i,2,t]}, \dots, \sum_{i=1}^{n_s} Y_{[i,n_s,t]}\right]. \tag{4}$$

To obtain the expected costs and QALYs per cycle for each strategy, $\mathbf{y}$, we apply again Equation (4) across all the matrices of the third dimension of $\mathbf{Y}$ for all the outcomes

```
# Vector of qalys under Usual Care
v_qaly_UC <- rowSums(t(colSums(a_Y_u_UC)))
# Vector of costs under Usual Care
v_cost_UC <- rowSums(t(colSums(a_Y_c_UC)))
# Vector of qalys under New Treatment
v_qaly_Tr <- rowSums(t(colSums(a_Y_u_Tr)))
# Vector of costs under New Treatment
v_cost_Tr <- rowSums(t(colSums(a_Y_c_Tr)))
```

The total expected discounted (TED) costs and QALYs, y, is obtained by applying Equation (1) to the expected outcomes accounting for transition rewards.

```
### For Usual Care
## QALYs
n_totqaly_UC <- t(v_qaly_UC) %*% v_dwe
## Costs
n_totcost_UC <- t(v_cost_UC) %*% v_dwc
### For New Treatment
## QALYs
n_totqaly_Tr <- t(v_qaly_Tr) %*% v_dwe
```

```
## Costs
n_totcost_Tr <- t(v_cost_Tr) %*% v_dwc
```

7 Cost-effectiveness analysis

We combine the total expected discounted costs and QALYs for both strategies into outcome-specific vectors. We use the R package dampack (https://github.com/DARTH-git/dampack) to calculate the incremental costs and effectiveness, and the incremental cost-effectiveness ratio (ICER) of the treatment strategy versus usual care and create the data frame df_cea with this information.

```
### Vector of costs
v_ted_cost <- c(n_totcost_UC, n_totcost_Tr)
### Vector of effectiveness
v_ted_qaly <- c(n_totqaly_UC, n_totqaly_Tr)

### Calculate incremental cost-effectiveness ratios (ICERs)
df_cea <- dampack::calculate_icers(cost = v_ted_cost,
                                   effect = v_ted_qaly,
                                   strategies = v_names_str)
```

The results of the CEA of the time-dependent Sick-Sicker model are presented in Table 6. The total expected QALYs for usual care and the new treatment are 19.96 and 20.657, respectively. The total expected costs for usual care and the new treatment are $116,415 and $213,867, respectively. Under the new treatment, the simulated cohort gets an expected benefit of 0.697 QALYs per individual for an additional expected cost of $97,452. The ICER, defined as the difference in costs divided by the difference in QALYs between the new treatment and usual care, is $139,794/QALY.

Table 6: Cost-effectiveness analysis results for the age-dependent Sick-Sicker model. ND: Non-dominated strategy.

Strategy	Costs ($)	QALYs	Incremental Costs ($)	Incremental QALYs	ICER ($/QALY)	Status
Usual care	116,415	19.960	NA	NA	NA	ND
New treatment	213,867	20.657	97,452	0.697	139,794	ND

Figure 11 shows the cost-effectiveness efficient frontier of the new treatment vs. usual care for the age-dependent Sick-Sicker model.

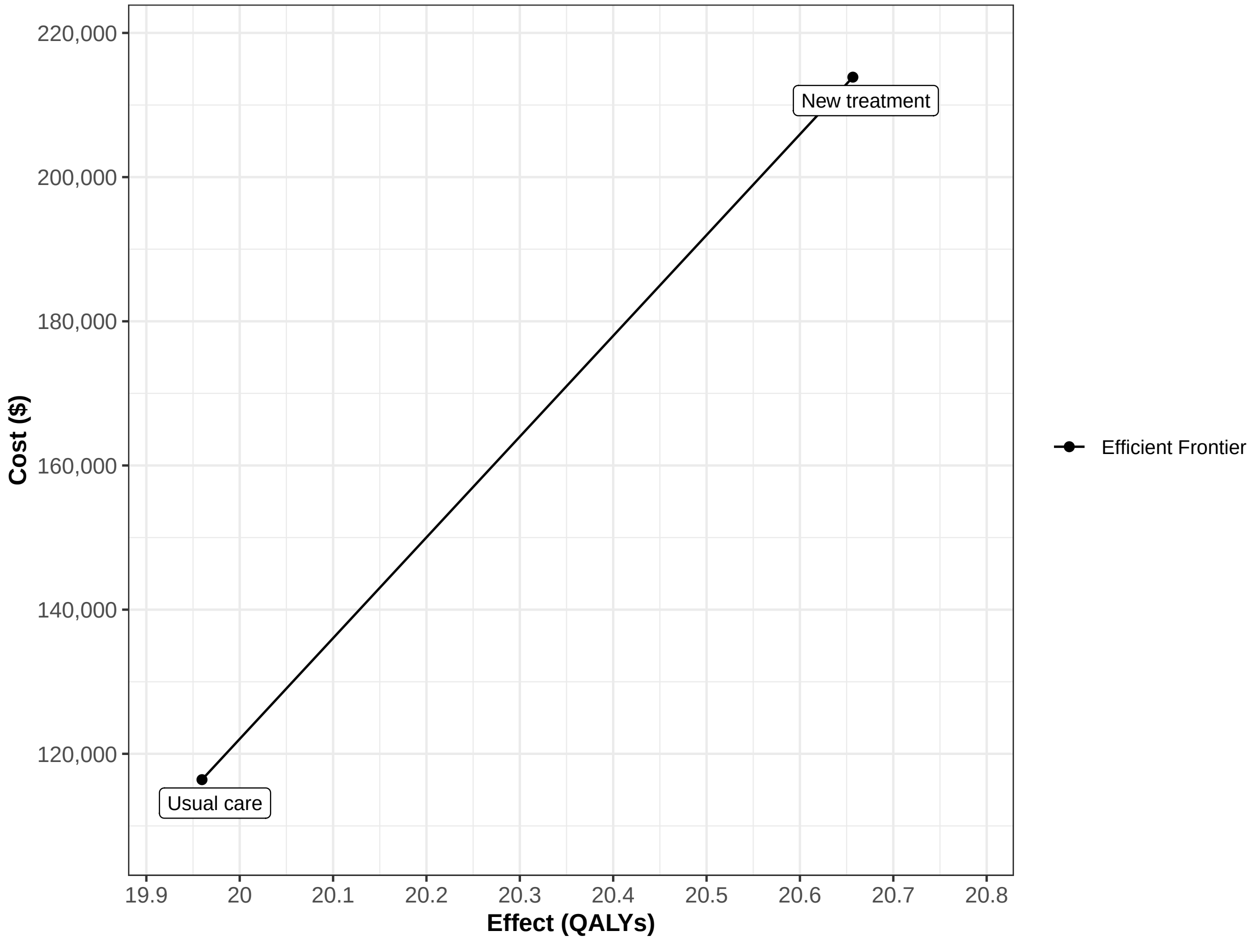


Figure 11: Cost-effectiveness effcicient frontier of the new treatment vs. usual care for the age-dependent Sick-Sicker model.

8 Discussion

In this tutorial, we provided a conceptualization of cSTMs with their mathematical description and a walk-through of their implementation for CEA in R using a previously published example. We used R as the programming language of choice to show the implementation of these models with accompanying code throughout the tutorial. We describe both time-independent and time-dependent models. We showed two different implementations of the time-dependent model, accounting for transition probabilities that depend on time since model start (e.g., age dependence) and on state residence.

We focused on discrete-time matrix-form cSTMs but continuous-time models or implemented via a set of difference equations[23,24] could also be implemented in R. We refer readers interested in learning more on continuous-time cSTMs to previously published papers[25–28] and a tutorial using R.[29]

We should note that a number of alternative approaches can be applied in the incorporation of time-dependency in cSTM in R and in a programming language in general. For example, another approach to incorporate age-dependency involves the updating of the time varying elements of the transition probability matrix P_t

at each time point t. That would alleviate the need for the construction of the array `a_P`. This can reduce computer memory requirements, but at the expense of increasing the number of operations in the update of P_t at every cycle. Another approach to account for state residence dependency, is to also use a 3-dimensional transition probability matrix with dimensions for current state, future state, and time in current state.[30] However, to incorporate age dependence, this multidimensional matrix will have to be expanded by another dimension or the states will have to be replicated for as many age groups are considered in the model. In this tutorial we decided to use the third dimension as the age of the cohort and incorporate state residence dependency by expanding the corresponding health states on the second dimension of the 3-dimensional to account for time spent in current state. A benefit of this approach is that we can use the transition dynamics array to capture all transitions between all states for all cycles.

cSTMs are recommended when the number of states is "not too large".[3] This recommendation arises because as the number of states increase, it becomes more difficult to keep track of their construction but not because of the added computational expense. It is possible to build a fairly complex cSTMs in R as long as the size of the transition probability matrix and outputs of interest can be stored in the RAM memory of the computer running the analysis. However, if the required number of state descriptions gets too large and difficult to manage its coding, it becomes preferable to use a stochastic (Monte Carlo) version of the state-transition model –often called microsimulation models– rather than a cohort simulation model.[14]

In summary, this tutorial provides a conceptualization of cSTMs and a step-by-step guide to implement them in R. We aim to add to the current body of literature and material on building this type of decision models so health decision scientists are able to develop cSTMs in a more flexible, open-source and transparent manner, and encouraging increased transparency and reproducibility.

Acknowledgements

Dr Alarid-Escudero was supported by a grant from the National Cancer Institute (U01-CA-199335) as part of the Cancer Intervention and Surveillance Modeling Network (CISNET). Miss Krijkamp was supported by the Society for Medical Decision Making (SMDM) fellowship through a grant by the Gordon and Betty Moore Foundation (GBMF7853). Dr Enns was supported by a grant from the National Institute of Allergy and Infectious Diseases of the National Institutes of Health under award no. K25AI118476. Dr Jalal was supported by a grant from the National Institutes of Health (KL2 TR0001856). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. The funding agencies had no role in the design of the study, interpretation of results, or writing of the manuscript. The funding agreement ensured the authors' independence in designing the study, interpreting the data, writing, and publishing the report.

References

1. Kuntz KM, Russell LB, Owens DK, et al. Decision Models in Cost-Effectiveness Analysis. In: Neumann PJ, Sanders GD, Russell LB, et al. (eds) *Cost-effectiveness in health and medicine*. New York, NY: Oxford

University Press, 2017, pp. 105–136.

2. Jalal H, Pechlivanoglou P, Krijkamp E, et al. An Overview of R in Health Decision Sciences. *Medical Decision Making*; 37: 735–746, http://journals.sagepub.com/doi/10.1177/0272989X16686559 (2017).

3. Siebert U, Alagoz O, Bayoumi AM, et al. State-Transition Modeling: A Report of the ISPOR-SMDM Modeling Good Research Practices Task Force-3. *Medical Decision Making*; 32: 690–700, http://mdm.sagepub.com/cgi/doi/10.1177/0272989X12455463 (2012).

4. Suijkerbuijk AW, Van Hoek AJ, Koopsen J, et al. Cost-effectiveness of screening for chronic hepatitis B and C among migrant populations in a low endemic country. *PLoS ONE* 2018; 13: 1–16.

5. Sathianathen NJ, Konety BR, Alarid-Escudero F, et al. Cost-effectiveness Analysis of Active Surveillance Strategies for Men with Low-risk Prostate Cancer. *European Urology*; 75: 910–917, https://linkinghub.elsevier.com/retrieve/pii/S0302283818308534 (2019).

6. Lu S, Yu Y, Fu S, et al. Cost-effectiveness of ALK testing and first-line crizotinib therapy for non-small-cell lung cancer in China. *PLoS ONE* 2018; 13: 1–12.

7. Djatche LM, Varga S, Lieberthal RD. Cost-Effectiveness of Aspirin Adherence for Secondary Prevention of Cardiovascular Events. *PharmacoEconomics - Open*; 2: 371–380, https://doi.org/10.1007/s41669-018-0075-2 (2018).

8. Pershing S, Enns EA, Matesic B, et al. Cost-Effectiveness of Treatment of Diabetic Macular Edema. *Annals of Internal Medicine* 2014; 160: 18–29.

9. Smith-Spangler CM, Juusola JL, Enns EA, et al. Population Strategies to Decrease Sodium Intake and the Burden of Cardiovascular Disease: A Cost-Effectiveness Analysis. *Annals of Internal Medicine*; 152: 481–487, http://annals.org/article.aspx?articleid=745729 (2010).

10. Kuntz KM, Weinstein MC. Modelling in economic evaluation. In: Drummond MF, McGuire A (eds) *Economic evaluation in health care: Merging theory with practice.* New York, NY: Oxford University Press, 2001, pp. 141–171.

11. Sonnenberg FA, Beck JR. Markov models in medical decision making: A practical guide. *Medical Decision Making*; 13: 322–338, http://mdm.sagepub.com/cgi/doi/10.1177/0272989X9301300409 (1993).

12. Beck J, Pauker SG. The Markov process in medical prognosis. *Medical Decision Making*; 3: 419–458, http://mdm.sagepub.com/cgi/doi/10.1177/0272989X8300300403 (1983).

13. Enns EA, Cipriano LE, Simons CT, et al. Identifying Best-Fitting Inputs in Health-Economic Model Calibration: A Pareto Frontier Approach. *Medical Decision Making*; 35: 170–182, http://www.ncbi.nlm.nih.gov/pubmed/24799456 (2015).

14. Krijkamp EM, Alarid-Escudero F, Enns EA, et al. Microsimulation Modeling for Health Decision Sciences Using R: A Tutorial. *Medical Decision Making*; 38: 400–422, http://journals.sagepub.com/doi/10.1177/0272989X18754513 (2018).

15. Alarid-Escudero F, Krijkamp E, Pechlivanoglou P, et al. A Need for Change! A Coding Framework for Improving Transparency in Decision Modeling. *PharmacoEconomics*; 37: 1329–1339, https://doi.org/10.

1007/s40273-019-00837-x (2019).

16. Iskandar R. A theoretical foundation of state-transition cohort models in health decision analysis. *PLOS ONE*; 13: e0205543, https://www.biorxiv.org/content/early/2018/09/28/430173 (2018).

17. Arias E, Heron M, Xu J. United States Life Tables, 2014. *National Vital Statistics Reports*; 66: 63, https://www.cdc.gov/nchs/data/nvsr/nvsr66/nvsr66{_}04.pdf (2017).

18. Rothman KJ, Greenland S, Lash TL. *Modern Epidemiology.* 3rd ed. Lippincott Williams & Wilkins, 2008.

19. Keiding N. Age-Specific Incidence and Prevalence: A Statistical Perspective. *Journal of the Royal Statistical Society Series A (Statistics in Society)* 1991; 154: 371–412.

20. Lee ET, Wang JW. *Statistical methods for Survival Data Analysis.* 3rd ed. Hoboken, NJ: Wiley, 2003.

21. Klein JP, Moeschberger ML. *Survival Analysis: Techniques for Censored and Truncated Data.* 2nd ed. Springer-Verlag, http://www.springer.com/statistics/life+sciences,+medicine+{\&}+health/book/978-0-387-95399-1 (2003).

22. Krijkamp EM, Alarid-Escudero F, Enns E, et al. A Multidimensional Array Representation of State-Transition Model Dynamics. *Medical Decision Making* 2019; In Press.

23. Grimmett G, Welsh D. Markov Chains. In: *Probability: An introduction.* Oxford University Press, pp. 203, www.statslab.cam.ac.uk/{~}grg/teaching/chapter12.pdf (2014).

24. Axler S, Gehring FW, Ribet KA. Difference Equations. New York, NY: Springer, http://link.springer.com/10.1007/0-387-27645-9 (2005).

25. Cao Q, Buskens E, Feenstra T, et al. Continuous-Time Semi-Markov Models in Health Economic Decision Making: An Illustrative Example in Heart Failure Disease Management. *Medical Decision Making*; 36: 59–71, http://mdm.sagepub.com/cgi/doi/10.1177/0272989X15593080 (2016).

26. Rosmalen J van, Toy M, O'Mahony JF. A Mathematical Approach for Evaluating Markov Models in Continuous Time without Discrete-Event Simulation. *Medical Decision Making*; 33: 767–779, http://mdm.sagepub.com/cgi/doi/10.1177/0272989X13487947 (2013).

27. Begun A, Icks A, Waldeyer R, et al. Identification of a multistate continuous-time nonhomogeneous Markov chain model for patients with decreased renal function. *Medical Decision Making*; 33: 298–306, http://www.ncbi.nlm.nih.gov/pubmed/23275452 (2013).

28. Soares MO, Canto E Castro L. Continuous time simulation and discretized models for cost-effectiveness analysis. *PharmacoEconomics*; 30: 1101–1117, http://www.ncbi.nlm.nih.gov/pubmed/23116289 (2012).

29. Frederix GWJ, Hasselt JGC van, Severens JL, et al. Development of a framework for cohort simulation in cost-effectiveness analyses using a multistep ordinary differential equation solver algorithm in R. *Medical Decision Making*; 33: 780–92, http://www.ncbi.nlm.nih.gov/pubmed/23515213 (2013).

30. Hawkins N, Sculpher M, Epstein D. Cost-effectiveness analysis of treatments for chronic disease: Using R to incorporate time dependency of treatment response. *Medical Decision Making*; 25: 511–9, http://www.ncbi.nlm.nih.gov/pubmed/16160207 (2005).