A MULTIDIMENSIONAL ARRAY REPRESENTATION OF STATE-TRANSITION MODEL DYNAMICS

A PREPRINT

Eline M. Krijkamp[*]
Department of Epidemiology
Erasmus University Medical Center
Rotterdam, The Netherlands

Fernando Alarid-Escudero[*†]
Drug Policy Program
Center for Research and Teaching in Economics
(CIDE)-CONACyT
Aguascalientes, Ags., Mexico, 20313
fernando.alarid@cide.edu

Eva A. Enns
Division of Health Policy and Management
University of Minnesota School of Public Health
Minneapolis, MN, USA 55455

Petros Pechlivanoglou
Child Health Evaluative Sciences
The Hospital for Sick Children, Toronto, ON, Canada
Institute for Health Policy Management and Evaluation
University of Toronto, ON, Canada

M.G. Myriam Hunink
Departments of Epidemiology and Radiology
Erasmus University Medical Center
Rotterdam, The Netherlands
Center of Health Decision Sciences
Harvard T.H. Chan School of Public Health
Boston, MA, USA

Hawre J. Jalal
Graduate School of Public Health
University of Pittsburgh
Pittsburgh, PA, USA

October 15, 2019

ABSTRACT

Cost-effectiveness analyses often rely on cohort state-transition models (cSTMs). The cohort trace is the primary outcome of cSTMs, which captures the proportion of the cohort in each health state over time (state occupancy). However, the cohort trace is an aggregated measure that does not capture the information about the specific transitions among the health states (transition dynamics). In practice, these transition dynamics are crucial in many applications, such as incorporating transition rewards or computing various epidemiological outcomes that could be used for model calibration and validation (e.g. disease incidence and lifetime risk). In this manuscript we propose modifying the transitional cSTMs cohort trace computation to compute and store cSTMs dynamics that capture both state occupancy and transition dynamics. This approach produces a multidimensional matrix from which both the state occupancy and the transition dynamics can be recovered. We highlight the advantages and potential applications of this approach with an example coded in R to facilitate the implementation of our method.

Keywords cost-effectiveness analysis · decision modeling · health economics · matrices · R project · state-transition models · transition dynamics · transition rewards · multidimensional arrays · tensors

[*]Contributed equally to this work.
[†]Corresponding author

State transition models (STM) are decision models commonly used in cost-effectiveness analysis (CEA) to capture economic and health outcomes of different strategies over time in discrete time cycles [1, 2]. In a cohort STM (cSTM), the disease dynamics are captured by distributing a closed cohort among a mutually exclusive and collectively exhaustive set of health states [2, 3, 4]. The cohort trace is the primary outcome of cSTMs, which stores the proportion of the cohort in each health state over time (i.e., it summarizes state occupancy) [5, 1]. A limitation of the cohort trace is that it does not keep track of the transitions among health states over time (i.e., the transition dynamics of the cohort). As a consequence, it can only be used to capture outcomes that involve residing in a state for a full cycle to compute the so-called *state rewards* and does not contain a mechanism to assign transition rewards, which are applied only when specific transitions occur. It also limits the type of epidemiological outcomes that can be obtained from cSTMs. For example, obtaining incidence of a disease requires knowledge of the proportion of the population transitioning from a subset of states without disease to the state(s) representing the disease of interest.

To overcome the limitations of the cohort trace, we propose a multidimensional array-based approach that serves as a full summary of cSTM dynamics that complements the already useful cohort trace. The proposed approach allows modelers to efficiently calculate all measures of interest that rely on transition dynamics and at the same time to aggregate this into a standard cohort trace.

We start by providing a formal definition of cSTMs elements and the cohort trace. We complement this standard notation with a description of the detailed transition dynamics. Then, we introduce the multidimensional-array structure and show how it can be easily generated. Finally, we illustrate its use to compute a measure of interest that depends on transitions among health states. We demonstrate this approach with an illustrative example of a cSTM programmed in R provided in the supplementary material and in GitHub `https://github.com/DARTH-git/state-transition-model-dynamics`)[9, 10].

Traditional cohort trace approach

We denote the distribution of the cohort across n_s health states in a cSTM at cycle t as the state vector $\mathbf{m}_t$ of dimensions $1 \times n_s$. That is, each element in $\mathbf{m}_t$ represents the proportion of the cohort in health state i at time t. Thus, $\mathbf{m}_t$ is written as

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

where the initial state vector $\mathbf{m}_0$ contains the distribution of the cohort across all n_s health states at the start of the simulation. The cohort distribution evolves over time governed by state transition probabilities. The probability of transitioning from health state i to health state j in cycle t is denoted as $p_{[i,j,t]}$. The collection of transition probabilities across the model states over the time horizon forms the time-dependent state transition probability matrix, P_t of dimensions $n_s \times n_s$,

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

For any t, all rows of P_t must sum to one. Note that if P_t is equal for all t times, equation (2) becomes a time-homogeneous transition probability matrix, where $P_t = P$.

The state vector at cycle $t+1$, $\mathbf{m}_{t+1}$, is then obtained by the inner product between the state vector at cycle t, $\mathbf{m}_t$, and the corresponding transition probability matrix P_t, such that

$$\mathbf{m_{t+1}} = \mathbf{m_t} P_t \quad \text{for } t = 0, \ldots, (n_t - 1). \tag{3}$$

Stacking the state vectors by rows for all $t = 0, \ldots, n_t$ results in the full cohort trace matrix, M, of dimensions $(n_t + 1) \times n_s$, where each row is a state vector $(-\mathbf{m}_t-)$, resulting in

$$M = \begin{bmatrix} -\mathbf{m}_0- \\ -\mathbf{m}_1- \\ \vdots \\ -\mathbf{m}_{n_t}- \end{bmatrix}. \tag{4}$$

Together, the state vectors $\mathbf{m}_t$, the transition probability matrices P_t and the cohort trace M in equations (1), (2) and (4), respectively, represent the three main components of a cSTM.

Dynamics-array approach

The trace matrix M aggregates transitions from all the states to a specific state, thus loses details of the transition dynamics. We propose to use a multidimensional array, $\mathbf{A}$, of dimensions $n_s \times n_s \times (n_t + 1)$ to store the proportion of the cohort that transitions between any two health states in each cycle over the time horizon. This array can be thought of as a set of two-dimensional matrices stacked along a third dimension that represents time. Below, we illustrate how to compute $\mathbf{A}$ from the three main components of cSTMs described above.

A_0 represents the first "slice" of $\mathbf{A}$. We compute A_0 as a matrix containing the initial state vector $\mathbf{m}_0$ in its diagonal and 0s in the off-diagonal, such that

$$A_0 = \text{diag}(\mathbf{m}_0) = \begin{bmatrix} m_{[0,1]} & 0 & \cdots & 0 \\ 0 & m_{[0,2]} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & m_{[0,n_s]} \end{bmatrix}. \tag{5}$$

Each subsequent $(t+1)$-th "slice" of $\mathbf{A}$ is obtained by multiplying a diagonal matrix of $\mathbf{m}_t$, denoted as diag($\mathbf{m}_t$), by P_t, such that

$$\mathbf{A}_{t+1} = \text{diag}(\mathbf{m}_t) \cdot P_t \quad \text{for } t = 1, \ldots, (n_t - 1). \tag{6}$$

The resulting elements of the "t-th slice" of $\mathbf{A}$, A_t for $t > 0$, are

$$A_t = \begin{bmatrix} a_{[1,1,t]} & a_{[1,2,t]} & \cdots & a_{[1,n_s,t]} \\ a_{[2,1,t]} & a_{[2,2,t]} & \cdots & a_{[2,n_s,t]} \\ \vdots & \vdots & \ddots & \vdots \\ a_{[n_s,1,t]} & a_{[n_s,2,t]} & \cdots & a_{[n_s,n_s,t]} \end{bmatrix}, \tag{7}$$

where $a_{[i,j,t]}$ is the proportion of the cohort that transitions from state i to state j between cycles $t-1$ and t, generated via

$$a_{[i,j,t]} = m_{[t-1,i]} p_{[i,j,t-1]} \quad \text{for } t = 1, \ldots, (n_t - 1), \tag{8}$$

where $m_{[t-1,i]}$ is the proportion of the cohort in state i at cycle $t-1$ and $p_{[i,j,t-1]}$ the corresponding transition probability at cycle $t-1$ of transitioning from state i to state j. In other words, $\mathbf{A}$ stores the transition dynamics of a simulated cohort in a cSTM.

Figure 1 compares graphically the computation involved in both the traditional cohort-trace approach (a) and the dynamics-array approach (b) and shows the structures of the resulting cohort trace M and dynamics-array $\mathbf{A}$ (c) and how $\mathbf{A}$ recovers the transition dynamics that are being aggregated in M.

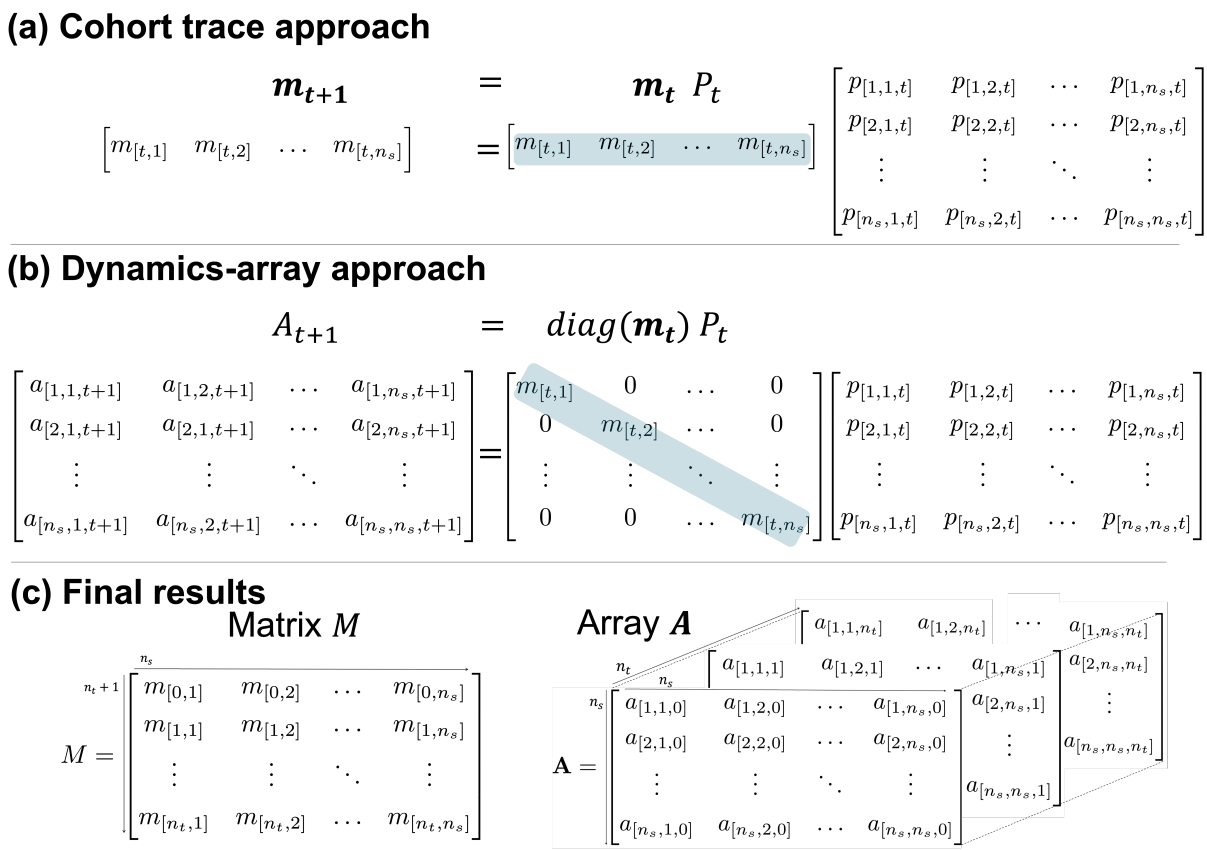

Figure 1: **(a)** The cohort trace approach computes row vector $\mathbf{m}_{t+1}$ of the cohort trace M that describes the distribution of the simulated cohort among the different health states at time $t+1$. $\mathbf{m}_{t+1}$ results from multiplying the state vector $\mathbf{m}_t$ (gray) by the transition probability matrix P_t. **(b)** The dynamics-array approach computes matrix A_{t+1} containing information regarding the transition dynamics of the simulated cohort at time $t+1$. The state vector $\mathbf{m}_t$ is highlighted (gray) to emphasize that the information in both approaches is identical. (**c**) Shows the resulting matrix M and array $\mathbf{A}$ of the approaches (a) and (b), respectively.

In R, it takes only a few lines of code to generate $\mathbf{A}$ complementary to M (Box 1).

```
for(t in 1:n.t){  # loop through the number of cycles
  m.M[t + 1, ]   <- m.M[t, ] %*% m.P          # estimate the state vector for the next cycle (t + 1)
  a.A[, , t + 1] <- diag(m.M[t, ]) %*% m.P    # estimate the transition dynamics at t + 1
}
```

Box 1: R code to iteratively generate the cohort trace M and the dynamics-array $\mathbf{A}$.

The cohort trace M can be computed from $\mathbf{A}$ by obtaining the t-th row of M, $\mathbf{m}_t$, summing each of the columns of $\mathbf{A}_t$ as follows:

$$\mathbf{m}_t = \mathbf{1}^T \mathbf{A}_t = \left[\sum_{i=1}^{n_s} a_{[i,1,t]}, \sum_{i=1}^{n_s} a_{[i,2,t]}, \ldots, \sum_{i=1}^{n_s} a_{[i,n_s,t]} \right], \tag{9}$$

where $\mathbf{1}$ is a vector of ones of dimension $n_s \times 1$. Although M can be obtained from $\mathbf{A}$ (avoiding computing M altogether), we prefer to compute both M and $\mathbf{A}$ simultaneously.

Applying state and transition rewards

One of the main advantages of $\mathbf{A}$ over M is the ability to incorporate transition rewards. Here, we demonstrate how to apply both state and transition rewards (e.g. utilities or cost) to the cSTM by using the dynamics-array, $\mathbf{A}$. Let R_t be a reward matrix of dimensions $n_s \times n_s$ that contains both state and transition rewards:

$$R_t = \begin{bmatrix} r_{[1,1,t]} & r_{[1,2,t]} & \cdots & r_{[1,n_s,t]} \\ r_{[2,1,t]} & r_{[2,2,t]} & \cdots & r_{[2,n_s,t]} \\ \vdots & \vdots & \ddots & \vdots \\ r_{[n_s,1,t]} & r_{[n_s,2,t]} & \cdots & r_{[n_s,n_s,t]} \end{bmatrix}, \tag{10}$$

where $r_{[i,j,t]}$ is the reward associated with transitioning from state i to state j at cycle t. When $j = i$, $r_{[i,i,t]}$ is the reward associated with staying in the i-th health state at cycle t. That is, the off-diagonal entries of R_t store the transition rewards and the diagonal of R_t stores the state rewards for cycle t. The state and transition rewards can be applied to the model dynamics by element-wise multiplication between A_t and R_t, indicated by the $\odot$ sign, which produces the matrix of outputs at cycle t, Y_t. Formally,

$$Y_t = A_t \odot R_t. \tag{11}$$

In R, applying these rewards required one additional line of code compared to Box 1, as shown in Box 2.

```
for(t in 1:n.t){  # loop through the number of cycles
  m.M[t + 1, ]    <- m.M[t, ] %*% m.P            # estimate the state vector for the next cycle (t + 1)
  a.A[, , t + 1] <- diag(m.M[t, ]) %*% m.P       # estimate the transition dynamics at t + 1
  # element-wise-multiplication of array A with the rewards matrices to apply both state and transition rewards
  a.Y[, , t + 1] <- a.A[, , t + 1] * m.R
}
```

Box 2: R code to apply time-invariant state and transition rewards to the model dynamics stored in array $\mathbf{A}$.

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

$$\mathbf{r}_t = \mathbf{1}^T Y_t = \left[\sum_{i=1}^{n_s} Y_{[i,1,t]}, \sum_{i=1}^{n_s} Y_{[i,2,t]}, \ldots, \sum_{i=1}^{n_s} Y_{[i,n_s,t]} \right]. \tag{12}$$

Implementation in R using an illustrative example

To facilitate the implementation of the array approach, we demonstrate its use with a stylistic healthy-sick-dead 3-state time-homogeneous cSTM example coded in R [10]. The model is used to simulate a cohort of 70-year-old individuals to compute their expected costs and quality-adjusted life years (QALYs) accrued over their remaining lifetime accounting for several transition rewards. The explanation of the model and the R code can be found in the supplementary material and more detailed on GitHub (`https://github.com/DARTH-git/state-transition-model-dynamics`).

Estimation of epidemiological measures

By obtaining $\mathbf{A}$, it is possible to compute epidemiological outcomes that otherwise would not have been easily derived from M. For example, obtaining incidence and lifetime risk from M would require creating additional steps, variables

or health states. Epidemiological outcomes could be used as outputs of simulation models for calibration or validation purposes. A full exposition of computing epidemiological measures from $\mathbf{A}$ is case-specific and is beyond the scope of this brief report. However, we illustrate the potential application of our approach by calculating a simple ratio from a generic cSTM.

Consider a cSTM with $n_s > 3$ health states. We are interested in calculating a ratio e_t of those that transition from health state 2 to health state n_s at cycle t to those that make this transition from health states 1, 2 and 3. Using the dynamics-array approach, the ratio e_t can be computed as

$$e_t = \frac{a_{[2,n_s,t]}}{a_{[1,n_s,t]} + a_{[2,n_s,t]} + a_{[3,n_s,t]}} \quad \text{for } t = 1, \ldots, n_t. \tag{13}$$

Discussion

We propose a multidimensional-array approach to overcome a limitation of the cohort trace produced by cSTM in not being able to store transition dynamics. The practical application of our approach involves adding a simple step to the traditional cohort trace approach that stores all transitions among health states over time in multidimensional array $\mathbf{A}$.

Traditionally, researchers have dealt with this limitation of the cSTM cohort trace by creating temporary health states that collect the state-to-state transition information. However, this solution can quickly complicate a model and result in an explosion of the number of health states. Using an individual-based microsimulation STM is another alternative [1], with considerable implications on computational time [11].

Another method that explicitly keeps track of state-to-state transitions is through a discretely integrated condition-event (DICE) simulation [8, 7]. DICE is a modeling technique that can free up some of the Markov restrictions that makes it possible to explicitly include many events occurring at various times. Although DICE simulation is a well-structured method, and the authors of the DICE papers provided very useful supplementary files to apply the method, we see the dynamics-array approach as a relatively simpler method to compute than DICE to overcome the limitation of the cohort trace on applying transition rewards and generating all the epidemiological outcomes of interest.

A potential limitation of the use of $\mathbf{A}$ is the additional computation needed when building the model. However, for many applications this may be a minor limitation given the matrix-based computational efficiency of current computers. Another potential limitation is the additional storage memory required to store $\mathbf{A}$, which could become a limitation in systems with limited memory. This could be an issue for computationally complex models with multiple states. However, the benefit of using $\mathbf{A}$ for large models is that all the complexity in the model dynamics is summarized into a compact structure which makes it relatively simple to extract information or to apply new rewards without re-running the model.

In conclusion, structuring the output of cSTMs using the dynamics-array is an efficient, simple and convenient approach to summarize the model dynamics. This simple structure allows applying state and transition rewards and obtaining epidemiological measures, while still being able to obtain and display the conventional cohort trace.

References

[1] Siebert U, Alagoz O, Bayoumi AM, Jahn B, Owens DK, Cohen DJ and Kuntz KM. State-Transition Modeling: A Report of the ISPOR-SMDM Modeling Good Research Practices Task Force-3. Value Health, Sep-Oct;15(6): 812–20, 2012.

[2] Kuntz KM, Russell BL, Owens DK, Sanders GD, Trikalions TA and Salomon JA. Decision Models in Cost-Effectiveness Analysis. In Cost-Effectiveness in Health and Medicine, Neumann, PJ, Ganiats TG, Russell LB, Sanders GD, Siegel JE, Chapter 5, pages 105–36. Oxford University Press, New York, second edition, 2017.

[3] Beck JR and Pauker SG. The markov process in medical prognosis. Med Decis Making, Dec;3(4):419–58, 1983.

[4] Iskandar R. A theoretical foundation for state-transition cohort models in health decision analysis. PLoS One, Dec;13(12): 1–11, 2018.

[5] Sonnenberg FA and Beck JR. Markov models in medical decision making: A practical guide. Med Decis Making, Oct-Dec;13(4): 322–38, 1993.

[6] Bouter LM, Zielhuis GA and Zeegers MPA. Textbook of Epidemiology Bohn Stafleu van Loghum, 2018.

[7] Caro JJ and Möller J. Adding Events to a Markov Model Using DICE Simulation Med Decis Making, Feb;38(2): 235-45, 2018.

[8] Caro JJ Discretely Integrated Condition Event (DICE) Simulation for Pharmacoeconomics PharmacoEconomics, Jul;34(7): 665-72, 2016.

[9] Jalal HJ, Pechlivanoglou P, Krijkamp EM, Alarid-Escudero F, Enns EA and Hunink MGM. An Overview of R in Health Decision Sciences. Med Decis Making, Oct;37(7): 735–46, 2017.

[10] R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[11] Krijkamp EM, Alarid-Escudero F, Enns EA, Jalal HJ, Hunink MGM and Pechlivanoglou P. Microsimulation Modeling for Health Decision Sciences Using R: A Tutorial. Med Decis Making, Apr;38(3): 400–22, 2018.

1 Supplementary material: R code of the stylistic 3-state model

Model description

We follow a cohort of healthy 70-year-old individuals over their remaining lifetime, using 30 annual cycles. The healthy individuals can transition to the sick health state, they can die or remain healthy. Sick individuals can fully recover, transitioning back to healthy, remain sick or die. Remaining in each of these health states is associated with some utilities and costs (the state rewards). In addition to these state rewards, transition dis-utilities and costs apply. Getting sick is associated with a sudden decrease of quality of life of 0.1. In addition, transitioning to dead incurs a one-time cost of $4,000. Both the state and transition rewards are constant over time. The R code to use the dynamics-array approach for this case example is shown below. All parameters of this model are fictitious, not based on a specific disease. On GitHub, we describe this simple 3-state example in more detail GitHub - (`https://github.com/DARTH-git/state-transition-model-dynamics`).

R code

We recommend downloading this code from GitHub to avoid errors due to copying from the manuscript.

```r
## Illustrative 3-state model showing the dynamics-array approach ##

# Load the packages
library(reshape2) # load reshape2 to transform data
library(ggplot2)  # load ggplot2 for nice looking plots

# initial set up
age          <- 70  # age of starting cohort
n.t          <- 30  # time horizon, number of cycles
v.age.names <- age:(age + n.t - 1) # vector with age names
v.n <- c("H", "S", "D") # vector with the 3 health states of the model:
# Healthy (H), Sick (S), Dead (D)
n.states <- length(v.n) # number of health states

#### Generate initial set of base-case external parameters ####
# Costs
c.H   <- 1000   # cost of remaining one cycle healthy
c.S   <- 3000   # cost of remaining one cycle sick
c.D   <- 0      # cost of being dead (per cycle)
# State utilities
u.H   <- 1      # utility when healthy
u.S   <- 0.60   # utility when sick
u.D   <- 0      # utility when healthy
# Transition probabilities (per cycle)
p.HS  <- 0.30   # probability to become sick when healthy
p.HD  <- 0.05   # probability to die when healthy
p.SH  <- 0.15   # probability to become healthy when sick
p.SD  <- 0.20   # probability to die when sick
# Transition rewards
du.HS  <- 0.10  # one-time utility decrement when becoming sick
ic.D   <- 4000  # one-time cost of dying

#### Transition probability matrix ####
# matrix m.P at the first cycle
m.P <- matrix(NA, nrow = n.states, ncol = n.states, dimnames = list(v.n, v.n))

```

```r
# Fill in matrix
# From Healthy
m.P["H", "H"]  <- 1 - (p.HS + p.HD)
m.P["H", "S"]  <- p.HS
m.P["H", "D"]  <- p.HD
# From Sick
m.P["S", "H"]  <- p.SH
m.P["S", "S"]  <- 1 - (p.SH + p.SD)
m.P["S", "D"]  <- p.SD
# From Death
m.P["D", "H"]  <- 0
m.P["D", "S"]  <- 0
m.P["D", "D"]  <- 1

#### Initial state vector ####
v.m0 <- c(H = 1, S = 0, D = 0) # initiate the vector

## Create the Markov cohort trace matrix m.M that captures the proportion of the
## cohort in each state at each cycle
m.M <- matrix(0,
              nrow = (n.t + 1),
              ncol = n.states,
              dimnames = list(0:n.t, v.n)) # initialize cohort trace matrix
m.M[1, ] <- v.m0   # store the initial state vector

# Initiate the array
a.A <- array(0,
             dim = c(n.states, n.states, n.t + 1),
             dimnames = list(v.n, v.n, 0:n.t)) # initialize array

diag(a.A[, , 1]) <- v.m0 # store the initial state vector in the diagonal of A

m.R.costs <- m.R.effects <- matrix(NA,
                                   nrow = n.states, ncol = n.states,
                                   dimnames = list(v.n, v.n))

# Fill in matrix for costs
# From Healthy
m.R.costs["H", "H"]  <- c.H
m.R.costs["S", "H"]  <- c.H
m.R.costs["D", "H"]  <- c.H
# From Sick
m.R.costs["H", "S"]  <- c.S
m.R.costs["S", "S"]  <- c.S
m.R.costs["D", "S"]  <- c.S
# From Death
m.R.costs["H", "D"]  <- c.D + ic.D
m.R.costs["S", "D"]  <- c.D + ic.D
m.R.costs["D", "D"]  <- c.D

# Fill in matrix for effects
# From Healthy
```

```r
m.R.effects["H", "H"]  <- u.H
m.R.effects["S", "H"]  <- u.H
m.R.effects["D", "H"]  <- u.H
# From Sick
m.R.effects["H", "S"]  <- u.S - du.HS
m.R.effects["S", "S"]  <- u.S
m.R.effects["D", "S"]  <- u.S
# From Death
m.R.effects["H", "D"]  <- u.D
m.R.effects["S", "D"]  <- u.D
m.R.effects["D", "D"]  <- u.D

#### Expected QALYs and Costs per cycle for each strategy ####
a.Y.costs <- a.Y.effects <- array(0,
                                  dim = c(n.states, n.states, n.t + 1),
                                  dimnames = list(v.n, v.n, 0:n.t))
# Initialize arrays
a.Y.costs[, , 1]   <- a.A[, , 1] * m.R.costs
a.Y.effects[, , 1] <- a.A[, , 1] * m.R.effects

### Run the model
for(t in 1:n.t){  # loop through the number of cycles
  # estimate the state vector for the next cycle (t + 1)
  m.M[t + 1, ] <- m.M[t, ] %*% m.P          # estimate the state vector for the next cycle (t + 1)
  a.A[, , t + 1] <- diag(m.M[t, ]) %*% m.P # estimate the transition dynamics at t + 1

  # element-wise-multiplication of array A with the rewards matrices
  a.Y.costs[, , t + 1]   <- a.A[, , t + 1] * m.R.costs
  a.Y.effects[, , t + 1] <- a.A[, , t + 1] * m.R.effects
}

# Perform a cost-effectiveness analysis
v.costs <- rowSums(t(colSums(a.Y.costs)))     # calculate the expected costs per cycle
v.QALYs <- rowSums(t(colSums(a.Y.effects)))  # calculate the expected QALYs per cycle
TC <- sum(v.costs)                            # calculate the total expected costs
TE <- sum(v.QALYs)                            # calculate the total expected QALYS
v.results <- c(TC, TE)                        # combine the total expected costs and QALYs
names(v.results) <- c("Costs", "Effect")      # name the vector
v.results                                     # print the results

##############################################################################
### Ratio of those that transitioned from sick to dead at each cycle to those that transitioned to dead from both healthy and sick.
v.e <- numeric(n.t + 1)   # create the vector v.e
v.e[1] <- 0               # initiate the vector

### calculate the ratio across all cycles starting in cycle 2
v.e[-1] <-  a.A["S", "D", -1] / (a.A["H", "D", -1] +  a.A["S", "D", -1])
```