

Simple 3-state Markov model in R

The DARTH workgroup

Developed by the Decision Analysis in R for Technologies in Health (DARTH) workgroup:

Fernando Alarid-Escudero, PhD (1)

Eva A. Enns, MS, PhD (2)

M.G. Myriam Hunink, MD, PhD (3,4)

Hawre J. Jalal, MD, PhD (5)

Eline M. Krijkamp, MSc (3)

Petros Pechlivanoglou, PhD (6,7)

Alan Yang, MSc (7)

In collaboration of:

1. Drug Policy Program, Center for Research and Teaching in Economics (CIDE) - CONACyT, Aguascalientes, Mexico
2. University of Minnesota School of Public Health, Minneapolis, MN, USA
3. Erasmus MC, Rotterdam, The Netherlands
4. Harvard T.H. Chan School of Public Health, Boston, USA
5. University of Pittsburgh Graduate School of Public Health, Pittsburgh, PA, USA
6. University of Toronto, Toronto ON, Canada
7. The Hospital for Sick Children, Toronto ON, Canada

Please cite our publications when using this code:

- Jalal H, Pechlivanoglou P, Krijkamp E, Alarid-Escudero F, Enns E, Hunink MG. An Overview of R in Health Decision Sciences. *Med Decis Making*. 2017; 37(3): 735-746. <https://journals.sagepub.com/doi/abs/10.1177/0272989X16686559>
- Krijkamp EM, Alarid-Escudero F, Enns EA, Jalal HJ, Hunink MGM, Pechlivanoglou P. Microsimulation modeling for health decision sciences using R: A tutorial. *Med Decis Making*. 2018;38(3):400–22. <https://journals.sagepub.com/doi/abs/10.1177/0272989X18754513>
- Krijkamp EM, Alarid-Escudero F, Enns E, Pechlivanoglou P, Hunink MM, Jalal H. A Multidimensional Array Representation of State-Transition Model Dynamics. *Med Decis Making*. Online First <https://doi.org/10.1177/0272989X19893973>

Copyright 2017, THE HOSPITAL FOR SICK CHILDREN AND THE COLLABORATING INSTITUTIONS. All rights reserved in Canada, the United States and worldwide. Copyright, trademarks, trade names and any and all associated intellectual property are exclusively owned by THE HOSPITAL FOR SICK CHILDREN and the collaborating institutions. These materials may be used, reproduced, modified, distributed and adapted with proper attribution.

```
rm(list = ls())      # clear memory (removes all the variables from the workspace)
```

01 Load packages

```
if (!require('pacman')) install.packages('pacman'); library(pacman) # use this package to conveniently  
  
## Loading required package: pacman  
# load (install if required) packages from CRAN  
p_load("diagram")
```

02 Load functions

```
# no functions required
```

03 Input model parameters

```
# Strategy names  
v_names_str <- c("Base Case")  
  
# Number of strategies  
n_str <- length(v_names_str)  
  
# Markov model parameters  
v_n <- c("Healthy", "Sick", "Dead") # state names  
n_states <- length(v_n) # number of states  
n_t <- 60 # number of cycles  
  
p_HD <- 0.02 # probability to die when healthy  
p_HS <- 0.05 # probability to become sick when healthy  
p_SD <- 0.1 # probability to die when sick  
  
# Costs and utilities  
c_H <- 400 # cost of remaining one cycle healthy  
c_S <- 1000 # cost of remaining one cycle sick  
c_D <- 0 # cost of remaining one cycle dead  
u_H <- 0.8 # utility when healthy  
u_S <- 0.5 # utility when sick  
u_D <- 0 # utility when dead  
d_e <- d_c <- 0.03 # equal discount of costs and QALYs by 3%  
  
# calculate discount weights for costs for each cycle based on discount rate d_c  
v_dwc <- 1 / (1 + d_e) ^ (0:n_t)  
# calculate discount weights for effectiveness for each cycle based on discount rate d_e  
v_dwe <- 1 / (1 + d_c) ^ (0:n_t)
```

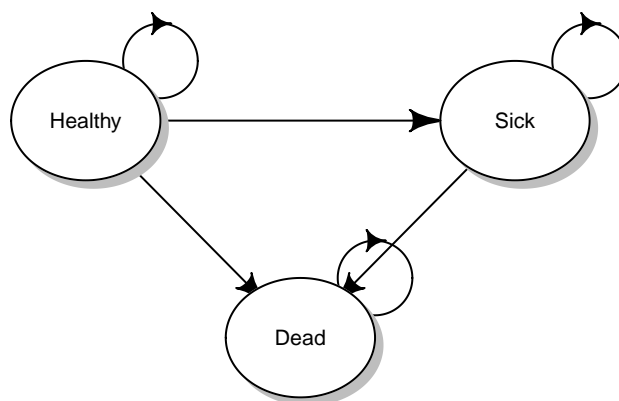
Draw the state-transition cohort model

```
m_P_diag <- matrix(0, nrow = n_states, ncol = n_states, dimnames = list(v_n, v_n))  
m_P_diag["Healthy", "Sick"] = ""
```

```

m_P_diag["Healthy", "Dead" ]      = ""
m_P_diag["Healthy", "Healthy" ]   = ""
m_P_diag["Sick"      , "Dead" ]   = ""
m_P_diag["Sick"      , "Sick" ]   = ""
m_P_diag["Dead"      , "Dead" ]   = ""
layout.fig <- c(2, 1)
plotmat(t(m_P_diag), t(layout.fig), self.cex = 0.5, curve = 0, arr.pos = 0.8,
        latex = T, arr.type = "curved", relsize = 0.85, box.prop = 0.8,
        cex = 0.8, box.cex = 0.7, lwd = 1)

```



04 Define and initialize matrices and vectors

04.1 Cohort trace

```

# create the cohort trace
m_M <- matrix(NA,
              nrow = n_t + 1 , # create Markov trace (n.t + 1 because R doesn't
                               # understand Cycle 0)
              ncol = n_states,
              dimnames = list(0:n_t, v_n))

m_M[1, ] <- c(1, 0, 0) # initialize first cycle of Markov trace

```

04.2 Transition probability matrix

```
# create the transition probability matrix
m_P <- matrix(0,
              nrow = n_states, ncol = n_states,
              dimnames = list(v_n, v_n)) # name the columns and rows of the transition
                                          # probability matrix

m_P

##           Healthy Sick Dead
## Healthy      0      0      0
## Sick         0      0      0
## Dead         0      0      0
```

Fill in the transition probability matrix:

```
# from Healthy
m_P["Healthy", "Healthy"] <- 1 - p_HD - p_HS
m_P["Healthy", "Sick"]    <- p_HS
m_P["Healthy", "Dead"]    <- p_HD

# from Sick
m_P["Sick", "Sick"] <- 1 - p_SD
m_P["Sick", "Dead"] <- p_SD

# from Dead
m_P["Dead", "Dead"] <- 1

# check rows add up to 1
rowSums(m_P)

## Healthy      Sick      Dead
##          1          1          1
```

05 Run Markov model

```
for (t in 1:n_t){
  m_M[t + 1, ] <- m_M[t, ] %*% m_P # loop through the number of cycles
                                     # estimate the state vector for the next cycle (t + 1)
}
```

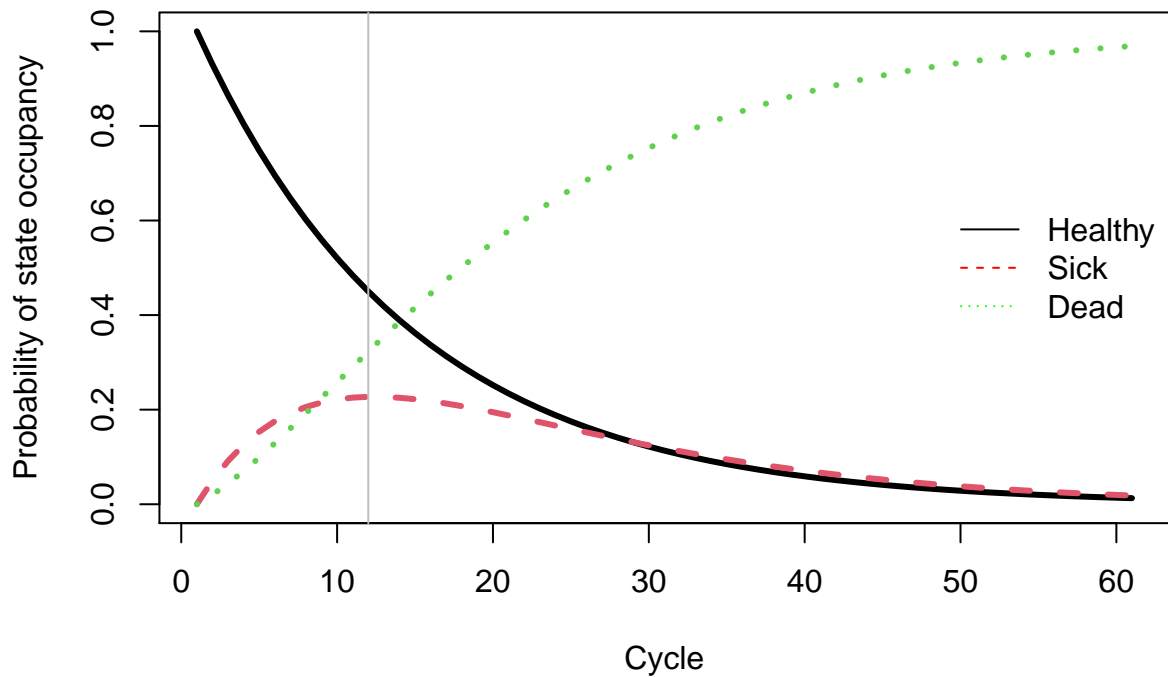
06 Compute and Plot Epidemiological Outcomes

06.1 Cohort trace

```
matplot(m_M, type = 'l',
        ylab = "Probability of state occupancy",
        xlab = "Cycle",
        main = "Cohort Trace", lwd = 3) # create a plot of the data
legend("right", v_n, col = c("black", "red", "green"),
      lty = 1:3, bty = "n") # add a legend to the graph

abline(v = which.max(m_M[, "Sick"]), col = "gray") # plot a vertical line that helps identifying at whi
```

Cohort Trace



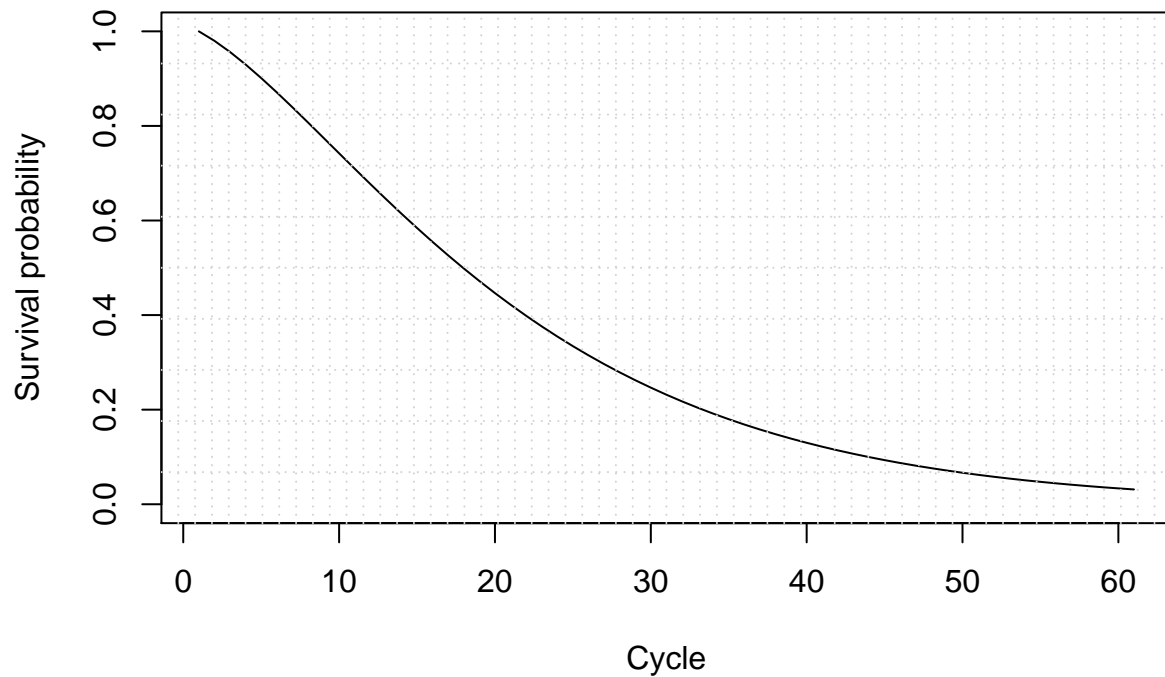
06.2 Overall Survival (OS)

```
v_os <- 1 - m_M[, "Dead"]           # calculate the overall survival (OS) probability
v_os <- rowSums(m_M[, 1:2])         # alternative way of calculating the OS probability

plot(v_os, type = 'l',
     ylim = c(0, 1),
     ylab = "Survival probability",
     xlab = "Cycle",
     main = "Overall Survival")      # create a simple plot showing the OS

# add grid
grid(nx = n_t, ny = 10, col = "lightgray", lty = "dotted", lwd = par("lwd"),
     equilogs = TRUE)
```

Overall Survival

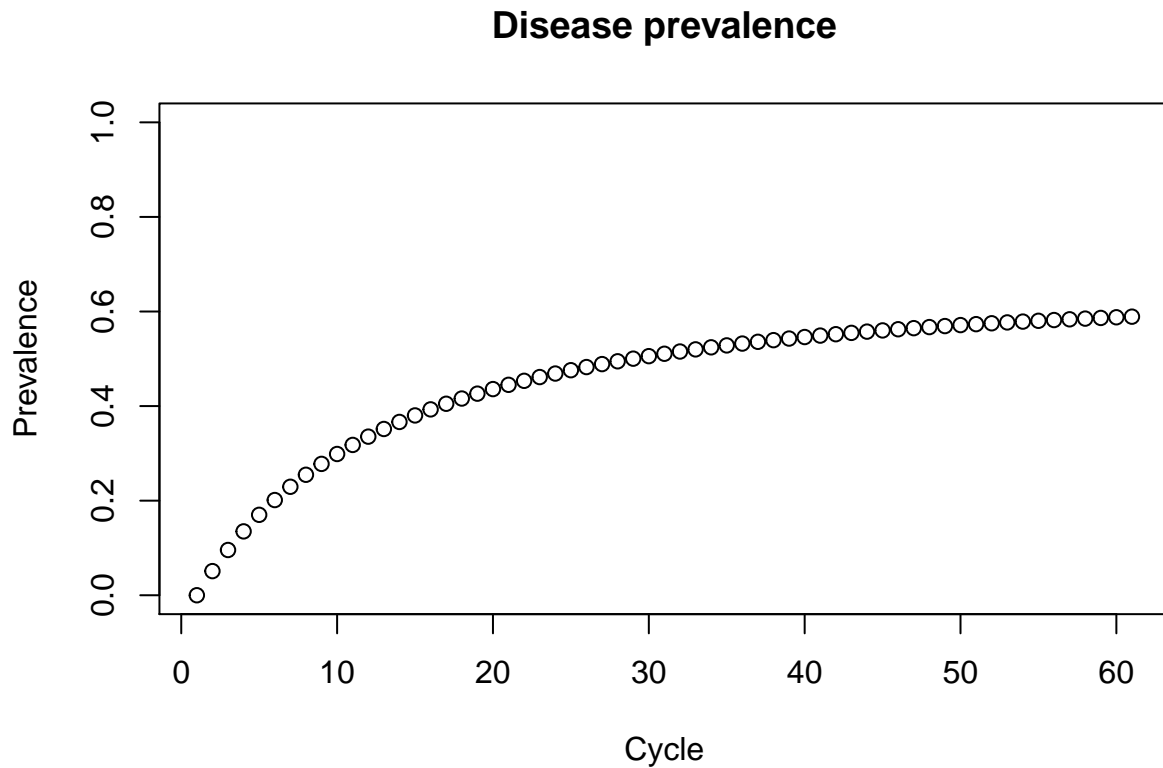


06.2.1 Life Expectancy (LE)

```
v_le <- sum(v_os)           # summing probability of OS over time (i.e. life expectancy)
```

06.3 Disease prevalence

```
v_prev <- m_M[, "Sick"]/v_os
plot(v_prev,
     ylim = c(0, 1),
     ylab = "Prevalence",
     xlab = "Cycle",
     main = "Disease prevalence")
```



07 Compute Cost-Effectiveness Outcomes

07.1 Mean Costs and QALYs

```
# per cycle
# calculate expected costs by multiplying m_M with the cost vector for the different
# health states
v_tc <- m_M %*% c(c_H, c_S, c_D)
# calculate expected QALYs by multiplying m_M with the utilities for the different
# health states
v_tu <- m_M %*% c(u_H, u_S, u_D)
```

07.2 Discounted Mean Costs and QALYs

```
# Discount costs by multiplying the cost vector with discount weights (v_dw)
v_tc_d <- t(v_tc) %*% v_dw
# Discount QALYS by multiplying the QALYs vector with discount weights (v_dw)
v_te_d <- t(v_tu) %*% v_dwe
```

07.3 Results

```
results <- data.frame( "Total Discounted Cost" = v_tc_d,
  "Life Expectancy" = v_le,
  "Total Discounted QALYs" = v_te_d,
```

```
check.names = F)
results
```

##	Total Discounted Cost	Life Expectancy	Total Discounted QALYs
## 1	8043.131	21.00019	10.18939