

Simple 3-state Markov model in R

with age dependency

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*. 2020 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

```
# no packages required
```

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 <- seq(0.003, 0.01, length.out = n_t) # probabilities to die when sick (age-dependent) - this is :
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)
```

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 array

```

# create the transition probability array
a_P <- array(0,                                     # Create 3-D array
            dim = c(n_states, n_states, n_t),
            dimnames = list(v_n, v_n, 0:(n_t - 1))) # name the dimensions of the array

```

Fill in the transition probability array:

```

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

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

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

```

04.3 Check if transition array and probabilities are valid

```

verbose <- TRUE
# Check if transition probabilities are valid (i.e., in [0, 1])
m_indices_notvalid <- arrayInd(which(a_P < 0 | a_P > 1),
                               dim(a_P))

try(
  if(dim(m_indices_notvalid)[1] != 0){
    v_rows_notval <- rownames(a_P)[m_indices_notvalid[, 1]]
    v_cols_notval <- colnames(a_P)[m_indices_notvalid[, 2]]
    v_cycles_notval <- dimnames(a_P)[[3]][m_indices_notvalid[, 3]]

    df_notvalid <- data.frame(`Transition probabilities not valid:` =
                             matrix(paste0(paste(v_rows_notval, v_cols_notval,
                                                  sep = "->"),
                                           "; at cycle ",
                                           v_cycles_notval), ncol = 1),
                             check.names = FALSE)

    if(verbose){
      message("Not valid transition probabilities")
      stop(print(df_notvalid), call. = FALSE)
    }
  }
)

# Check if transition probability array is valid
valid <- apply(a_P, 3, function(x) all.equal(sum(rowSums(x)), n_states))

```

```

if (!isTRUE(all.equal(as.numeric(sum(valid)), as.numeric(n_t)))) {
  if(verbose){
    stop("This is not a valid transition Matrix")
  }
}

```

05 Run Markov model

```

for (t in 1:n_t){ # t<-1 # loop through the number of cycles
  m_M[t + 1, ] <- m_M[t, ] %*% a_P[, , t] # estimate the Markov trace for cycle t + 1
                                              # using the t-th matrix from the
                                              # probability array
}
head(m_M) # print the first lines of the matrix

```

```

##      Healthy      Sick      Dead
## 0 1.0000000 0.0000000 0.0000000
## 1 0.9470000 0.0500000 0.0030000
## 2 0.8966966 0.0923500 0.01095336
## 3 0.8489589 0.1279498 0.02309122
## 4 0.8036620 0.1576028 0.03873525
## 5 0.7606865 0.1820256 0.05728792

```

06 Compute and Plot Epidemiological Outcomes

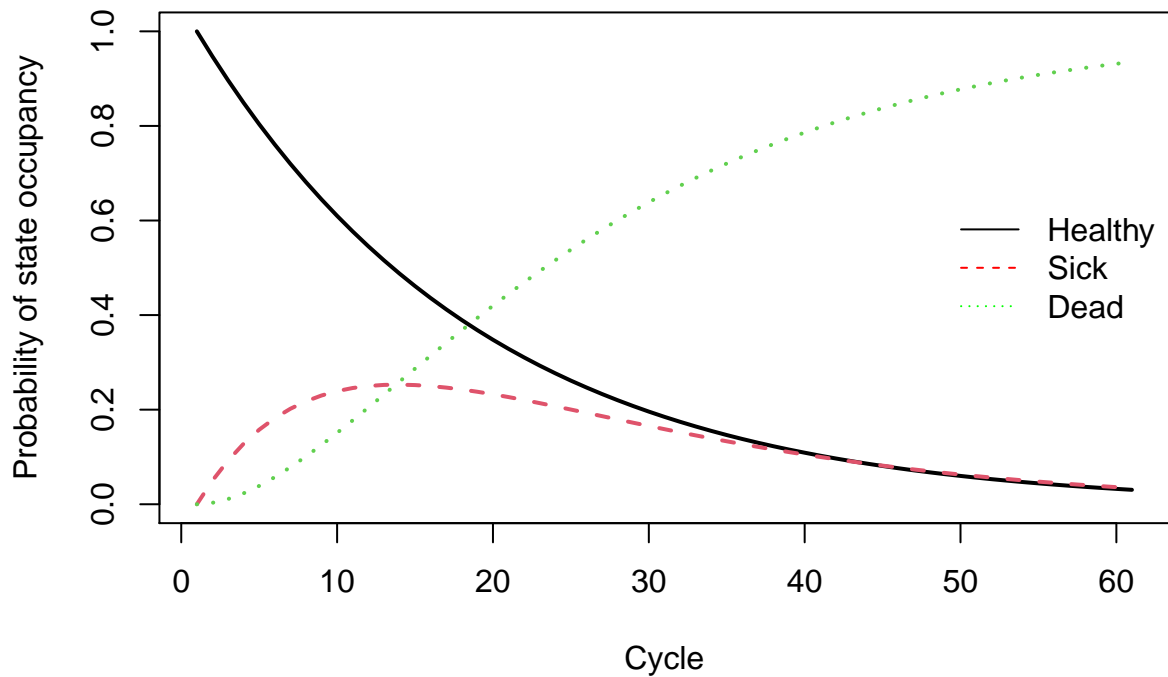
06.1 Cohort trace

```

# create a plot of the data
matplot(m_M, type = 'l',
        ylab = "Probability of state occupancy",
        xlab = "Cycle",
        main = "Cohort Trace", lwd = 2)
# add a legend to the graph
legend("right", v_n, col = c("black", "red", "green"), lty = 1:3, bty = "n")

```

Cohort Trace



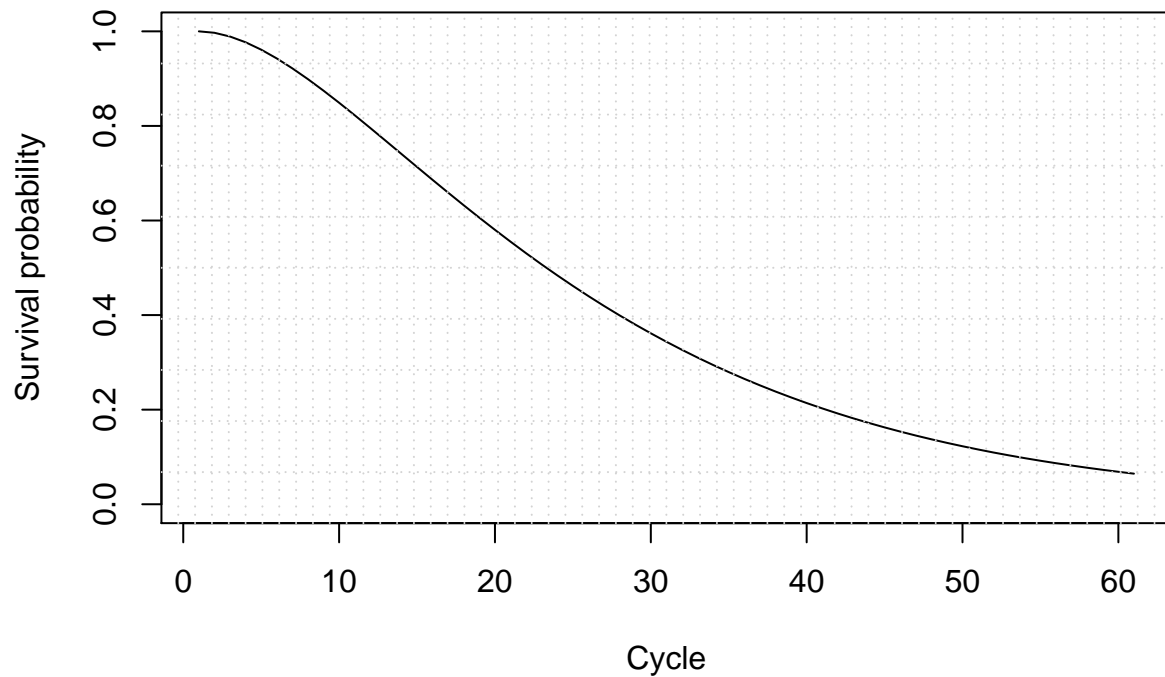
06.2 Overall Survival (OS)

```
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

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

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

Overall Survival

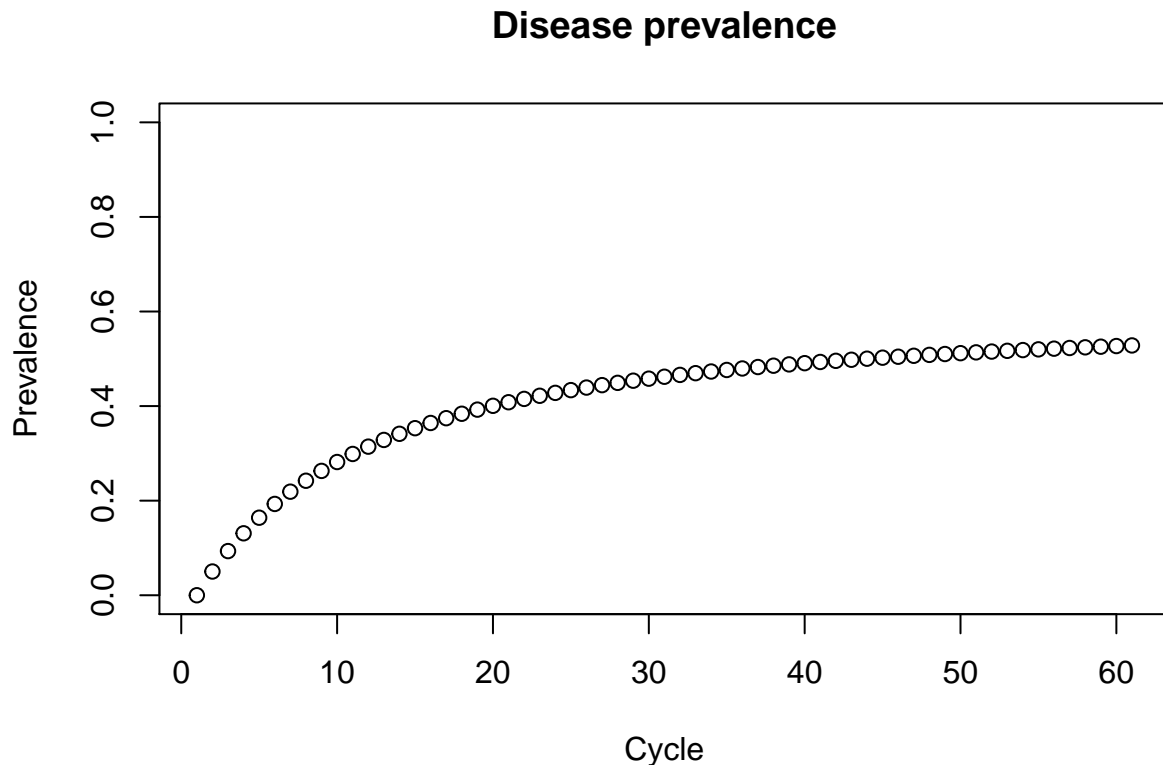


06.2.1 Life Expectancy (LE)

```
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	9501.679	26.27112	12.04905