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. Division of Public Administration, Center for Research and Teaching in Economics (CIDE), 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

- Alarid-Escudero F, Krijkamp EM, Enns EA, Yang A, Hunink MGM Pechlivanoglou P, Jalal H. Cohort State-Transition Models in R: A Tutorial. arXiv:200107824v2. 2020:1-48. http://arxiv.org/abs/2001.07824

- 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 Feb;40(2):242-248. https://journals.sagepub.com/doi/10.1177/0272989X19893973

Change `eval` to `TRUE` if you want to knit this document.

```
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
# load (install if required) packages from CRAN
p_load( "dplyr", "devtools", "scales", "ellipse", "ggplot2", "lazyeval", "igraph", "ggraph", "reshape2"
# install_github("DARTH-git/dampack", force = TRUE) Uncomment if there is a newer version
# install_github("DARTH-git/darthtools", force = TRUE) Uncomment if there is a newer version
p_load_gh("DARTH-git/dampack", "DARTH-git/darthtools")
```

02 Load functions

```
# all functions are in the darthtools package
```

03 Input model parameters

```
# Strategy names
v_names_str <- c("Standard of Care", "Treatment")

# Markov model parameters
v_n  <- c("Healthy", "Sick", "Dead")  # state names
n_t  <- 60                            # number of cycles

v_init <- c("Healthy" = 1,
            "Sick"    = 0,
            "Dead"    = 0)            # initial cohort distribution (everyone allocated to the
                                      # "healthy" state)

# Transition probabilities
p_HD      <- 0.02                     # probability of dying when healthy
p_HS      <- 0.05                     # probability of becoming sick when healthy, under standard of ca
p_HS_trt  <- 0.03                     # probability of becoming sick when healthy, under treatment
p_SD      <- 0.1                      # probability of dying when sick

# Costs and utilities
c_H       <- 400                      # cost of one cycle in healthy state
c_S       <- 1000                     # cost of one cycle in sick state
c_D       <- 0                        # cost of one cycle in dead state
c_trt     <- 800                      # cost of treatment (per cycle)
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              # discount rate per cycle equal discount of costs and QALYs by 3%

n_str     <- length(v_names_str)      # Number of strategies
n_states  <- length(v_n)              # number of states

# Discount weights for costs and effects
v_dwc     <- 1 / (1 + d_c) ^ (0:n_t)
v_dwe     <- 1 / (1 + d_e) ^ (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_SoC <- m_M_trt <-  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_SoC[1, ] <- m_M_trt[1, ] <- v_init        # initialize first cycle of Markov trace
```

04.2 Transition probability matrix

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

# print the probability matrices
m_P_SoC  # for standard of care
m_P_trt  # treatment
```

Fill in the transition probability matrix:

```r
# For Standard of Care
# from Healthy
m_P_SoC["Healthy", "Healthy"] <- (1 - p_HD) * (1 - p_HS)
m_P_SoC["Healthy", "Sick"]    <- (1 - p_HD) * p_HS
m_P_SoC["Healthy", "Dead"]    <- p_HD

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

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

# Under treatment
m_P_trt <- m_P_SoC  # Assign the matrix for standard of care to the transition probability matrix for t
# replace values that are different under treatment
m_P_trt["Healthy", "Healthy"] <- (1 - p_HD) * (1 - p_HS_trt)
m_P_trt["Healthy", "Sick"]    <- (1 - p_HD) * p_HS_trt
```

04.3 Check if transition probability structure and probabilities are valid

```r
# Check that transition probabilities are in [0, 1]
check_transition_probability(m_P_SoC, verbose = TRUE)
check_transition_probability(m_P_trt, verbose = TRUE)
# Check that all rows sum to 1
check_sum_of_transition_array(m_P_SoC, n_states = n_states, n_cycles = n_t, verbose = TRUE)
check_sum_of_transition_array(m_P_trt, n_states = n_states, n_cycles = n_t, verbose = TRUE)
```

05 Run Markov model

```r
for (t in 1:n_t){  # loop through the number of cycles
  m_M_SoC[t + 1, ] <- m_M_SoC[t, ] %*% m_P_SoC  # estimate the state vector for the next cycle (t + 1)
  m_M_trt[t + 1, ] <- m_M_trt[t, ] %*% m_P_trt  # for treatment
}
```

06 Compute and Plot Epidemiological Outcomes

06.1 Cohort trace

Standard of Care:

```r
matplot(m_M_SoC, type = 'l',
        ylab = "Probability of state occupancy",
        xlab = "Cycle",
        main = "Cohort Trace - standard of care", lwd = 3)  # create a plot of the data
```

```r
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_SoC[, "Sick"]), col = "gray")     # plot a vertical line that helps identifyi
```

Treatment:

```r
matplot(m_M_trt, type = 'l',
        ylab = "Probability of state occupancy",
        xlab = "Cycle",
        main = "Cohort Trace - treatment", 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_trt[, "Sick"]), col = "gray")     # plot a vertical line that helps identifyi
```

06.2 Overall Survival (OS)

Standard of Care:

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

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

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

Treatment:

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

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

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

06.2.1 Life Expectancy (LE)

```r
v_le_SoC <- sum(v_os_SoC)  # summing probability of OS over time (i.e. life expectancy)
v_le_trt <- sum(v_os_trt)  # summing probability of OS over time (i.e. life expectancy), treatment
```

06.3 Disease prevalence

Standard of Care:

```r
v_prev_SoC <- m_M_SoC[, "Sick"]/v_os_SoC
plot(v_prev_SoC,
     ylim = c(0, 1),
     ylab = "Prevalence",
     xlab = "Cycle",
     main = "Disease prevalence - Standard of care")
```

Treatment:

```r
v_prev_trt <- m_M_trt[, "Sick"]/v_os_trt
plot(v_prev_trt,
     ylim = c(0, 1),
     ylab = "Prevalence",
     xlab = "Cycle",
     main = "Disease prevalence - Treatment")
```

07 Compute Cost-Effectiveness Outcomes

07.1 Mean Costs and QALYs

```r
# per cycle
# calculate expected costs by multiplying m_M with the cost vector for the different
# health states
v_tc_SoC <- m_M_SoC %*% c(c_H, c_S,          c_D)  # Standard of Care
v_tc_trt <- m_M_trt %*% c(c_H, c_S + c_trt, c_D)  # Treatment
# calculate expected QALYs by multiplying m_M with the utilities for the different
# health states
v_tu_SoC <- m_M_SoC %*% c(u_H, u_S, u_D)        # Standard of Care
v_tu_trt <- m_M_trt %*% c(u_H, u_S, u_D)        # Treatment
```

07.2 Discounted Mean Costs and QALYs

```r
# Discount costs by multiplying the cost vector with discount weights
tc_d_SoC  <-  t(v_tc_SoC) %*% v_dwc  # Standard of Care
tc_d_trt  <-  t(v_tc_trt) %*% v_dwc  # Treatment
# Discount QALYS by multiplying the QALYs vector with discount weights
tu_d_SoC  <-  t(v_tu_SoC) %*% v_dwe  # Standard of Care
```

```
tu_d_trt  <-  t(v_tu_trt) %*% v_dwe  # Treatment

# store them into a vector
v_tc_d    <- c(tc_d_SoC, tc_d_trt)
v_tu_d    <- c(tu_d_SoC, tu_d_trt)

# Dataframe with discounted costs and effectiveness
df_ce       <- data.frame(Strategy = v_names_str,
                          Cost     = v_tc_d,
                          Effect   = v_tu_d)
df_ce
```

07.3 Compute ICERs of the Markov model

```
df_cea <- calculate_icers(cost       = df_ce$Cost,
                          effect     = df_ce$Effect,
                          strategies = df_ce$Strategy)
df_cea
```

07.4 Plot frontier of the Markov model

```
plot(df_cea, effect_units = "QALYs", xlim = c(10, 12))

# note: you need to adjust the xlim values to values that are covering the range of effect values in you
```