3-state Markov model in R

with dependency for time-since model start & state-residency 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. 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 Mak. 2020;40(2):242-248. <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.

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 install other packages  
# load (install if required) packages from CRAN  
p\_load("diagram", "dampack")   
# install\_github("DARTH-git/darthtools", force = TRUE) Uncomment if there is a newer version  
p\_load\_gh("DARTH-git/darthtools")

# 02 Load functions

# No functions needed

# 03 Input model parameters

## General setup  
n\_cycles <- 60 # number of cycles  
v\_names\_cycles <- paste("cycle", 0:n\_cycles) # cycle names  
v\_names\_states <- c("Healthy", "Sick", "Dead") # state names  
n\_states <- length(v\_names\_states) # number of health states   
  
# Discounting factors  
d\_c <- 0.03 # discount rate for costs   
d\_e <- 0.03 # discount rate for QALYs  
  
# Strategy names  
v\_names\_str <- c("Standard of Care", # store the strategy names  
 "Treatment A",   
 "Treatment B")   
n\_str <- length(v\_names\_str) # number of strategies  
  
## Transition probabilities  
p\_HS\_SoC <- 0.05 # probability of becoming sick when healthy, conditional on surviving, under standard of care  
p\_HS\_trtA <- 0.04 # probability of becoming sick when healthy, conditional on surviving, under treatment A  
p\_HS\_trtB <- 0.02 # probability of becoming sick when healthy, conditional on surviving, under treatment B  
p\_SD <- 0.1 # probability of dying   
p\_HD\_min <- 0.003 # probability of dying when healthy at t = 0  
p\_HD\_max <- 0.01 # probability of dying when healthy at t = n\_cycles  
# probabilities of dying when healthy (age-dependent) - this is now a sequence of numbers  
v\_p\_HD <- seq(p\_HD\_min, p\_HD\_max, length.out = n\_cycles)   
  
## Tunnels  
n\_tunnel\_size <- n\_cycles  
# Sick state  
v\_Sick\_tunnels <- paste("Sick\_", seq(1, n\_tunnel\_size), "Yr", sep = "")  
# Create variables for time-dependent model  
v\_names\_states\_tunnels <- c("Healthy", v\_Sick\_tunnels, "Dead") # state names  
n\_states\_tunnels <- length(v\_names\_states\_tunnels) # number of states  
  
# Weibull parameters  
l <- 0.08  
g <- 1.1  
  
S\_SD <- pweibull(0:n\_tunnel\_size, l , g, lower.tail = F)  
p\_SD <- trans\_prob(S\_SD) # probability of dying when sick (time-in-state dependent)  
  
  
  
## State rewards  
# 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\_trtA <- 800 # cost of treatment A (per cycle) in healthy state  
c\_trtB <- 1500 # cost of treatment B (per cycle) in healthy state  
u\_H <- 1 # utility when healthy   
u\_S <- 0.5 # utility when sick  
u\_D <- 0 # utility when dead  
d\_e <- 0.03 # discount rate per cycle equal discount of costs and QALYs by 3%  
d\_c <- 0.03 # discount rate per cycle equal discount of costs and QAL   
  
# Discount weight (equal discounting is assumed for costs and effects)  
v\_dwc <- 1 / (1 + d\_c) ^ (0:n\_cycles)   
v\_dwe <- 1 / (1 + d\_e) ^ (0:n\_cycles)

# 04 Define and initialize matrices and vectors

## 04.1 Cohort trace

m\_M\_SoC <- matrix(NA,   
 nrow = (n\_cycles + 1), ncol = n\_states\_tunnels,   
 dimnames = list(v\_names\_cycles, v\_names\_states\_tunnels))   
  
# The cohort starts as healthy  
# initialize first cycle of Markov trace accounting for the tunnels  
m\_M\_SoC[1, ] <- c(1, rep(0, n\_tunnel\_size), 0)   
m\_M\_trtA <- m\_M\_trtB <- m\_M\_SoC # structure and initial states remain the same

## 04.2 Transition probability array

# create the transition probability array  
a\_P\_SoC <- array(0, # Create 3-D array  
 dim = c(n\_states\_tunnels, n\_states\_tunnels, n\_cycles),   
 dimnames = list(v\_names\_states\_tunnels, v\_names\_states\_tunnels,  
 v\_names\_cycles[-length(v\_names\_cycles)])) # name the dimensions of the array

Fill in the transition probability array:

## Standard of Care  
# from Healthy  
a\_P\_SoC["Healthy", "Healthy", ] <- (1 - v\_p\_HD) \* (1 - p\_HS\_SoC)  
a\_P\_SoC["Healthy", "Sick\_1Yr", ] <- (1 - v\_p\_HD) \* p\_HS\_SoC  
a\_P\_SoC["Healthy", "Dead", ] <- v\_p\_HD  
  
# from Sick  
for(i in 1:(n\_tunnel\_size - 1)){   
 a\_P\_SoC[v\_Sick\_tunnels[i], v\_Sick\_tunnels[i + 1], ] <- 1 - p\_SD[i]  
 a\_P\_SoC[v\_Sick\_tunnels[i], "Dead", ] <- p\_SD[i]  
}  
  
a\_P\_SoC[v\_Sick\_tunnels[n\_tunnel\_size], v\_Sick\_tunnels[n\_tunnel\_size], ] <- 1 - p\_SD[n\_tunnel\_size]  
a\_P\_SoC[v\_Sick\_tunnels[n\_tunnel\_size], "Dead", ] <- p\_SD[n\_tunnel\_size]  
  
# from Dead  
a\_P\_SoC["Dead", "Dead", ] <- 1  
  
## Treatment A  
a\_P\_trtA <- a\_P\_SoC  
a\_P\_trtA["Healthy", "Healthy", ] <- (1 - v\_p\_HD) \* (1 - p\_HS\_trtA)  
a\_P\_trtA["Healthy", "Sick\_1Yr", ] <- (1 - v\_p\_HD) \* p\_HS\_trtA  
  
## Treatment B  
a\_P\_trtB <- a\_P\_SoC  
a\_P\_trtB["Healthy", "Healthy", ] <- (1 - v\_p\_HD) \* (1 - p\_HS\_trtB)  
a\_P\_trtB["Healthy", "Sick\_1Yr", ] <- (1 - v\_p\_HD) \* p\_HS\_trtB

Check if transition array and probabilities are valid.

# Check that transition probabilities are in [0, 1]  
check\_transition\_probability(a\_P\_SoC, verbose = TRUE)  
check\_transition\_probability(a\_P\_trtA, verbose = TRUE)  
check\_transition\_probability(a\_P\_trtB, verbose = TRUE)  
# Check that all rows sum to 1  
check\_sum\_of\_transition\_array(a\_P\_SoC, n\_states = n\_states\_tunnels, n\_cycles = n\_cycles, verbose = TRUE)  
check\_sum\_of\_transition\_array(a\_P\_trtA, n\_states = n\_states\_tunnels, n\_cycles = n\_cycles, verbose = TRUE)  
check\_sum\_of\_transition\_array(a\_P\_trtB, n\_states = n\_states\_tunnels, n\_cycles = n\_cycles, verbose = TRUE)

# 05 Run Markov model

for (t in 1:n\_cycles){ # loop through the number of cycles  
 # estimate the cohort trace for cycle t + 1 using the t-th matrix from the probability array   
 m\_M\_SoC [t + 1, ] <- m\_M\_SoC [t, ] %\*% a\_P\_SoC [, , t]   
 m\_M\_trtA[t + 1, ] <- m\_M\_trtA[t, ] %\*% a\_P\_trtA[, , t]   
 m\_M\_trtB[t + 1, ] <- m\_M\_trtB[t, ] %\*% a\_P\_trtB[, , t]   
}

Create aggregated trace. Sum all Sick states together.

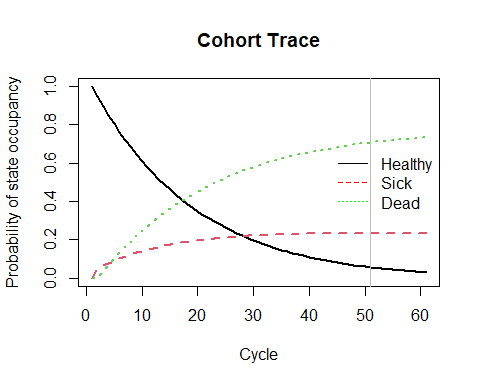
m\_M\_tunnels\_SoC <- cbind(Healthy = m\_M\_SoC[, "Healthy"],   
 Sick = rowSums(m\_M\_SoC[, 2:(n\_tunnel\_size + 1)]),   
 Dead = m\_M\_SoC[, "Dead"])  
m\_M\_tunnels\_trtA <- cbind(Healthy = m\_M\_trtA[, "Healthy"],   
 Sick = rowSums(m\_M\_trtA[, 2:(n\_tunnel\_size + 1)]),   
 Dead = m\_M\_trtA[, "Dead"])  
m\_M\_tunnels\_trtB <- cbind(Healthy = m\_M\_trtB[, "Healthy"],   
 Sick = rowSums(m\_M\_trtB[, 2:(n\_tunnel\_size + 1)]),   
 Dead = m\_M\_trtB[, "Dead"])  
head(m\_M\_tunnels\_SoC) # show the first rows of the aggregated Markov trace

## Healthy Sick Dead  
## cycle 0 1.0000000 0.00000000 0.00000000  
## cycle 1 0.9471500 0.04985000 0.00300000  
## cycle 2 0.8969864 0.06568843 0.03732521  
## cycle 3 0.8493784 0.07966611 0.07095546  
## cycle 4 0.8042016 0.09230318 0.10349524  
## cycle 5 0.7613370 0.10384287 0.13482017

# 06 Compute and Plot Epidemiological Outcomes

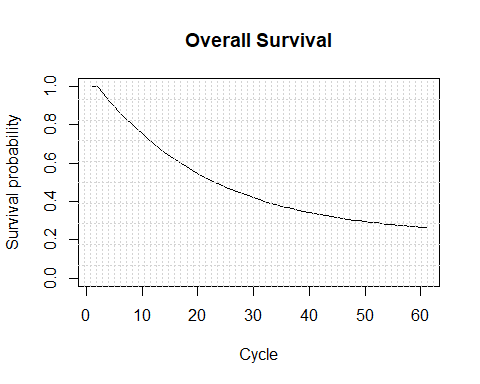
## 06.1 Cohort trace

# create a plot of the data  
matplot(m\_M\_tunnels\_SoC, type = 'l',   
 ylab = "Probability of state occupancy",  
 xlab = "Cycle",  
 main = "Cohort Trace", lwd = 2)   
# add a legend to the graph  
legend("right", v\_names\_states, col = c("black", "red", "green"), lty = 1:3, bty = "n")  
  
# plot a vertical line that helps identifying at which cycle the prevalence of sick is highest  
abline(v = which.max(m\_M\_tunnels\_SoC[, "Sick"]), col = "gray")



## 06.2 Overall Survival (OS)

v\_os <- 1 - m\_M\_tunnels\_SoC[, "Dead"] # calculate the overall survival (OS) probability  
v\_os <- rowSums(m\_M\_tunnels\_SoC[, 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\_cycles, ny = 10, col = "lightgray", lty = "dotted", lwd = par("lwd"), equilogs = TRUE)

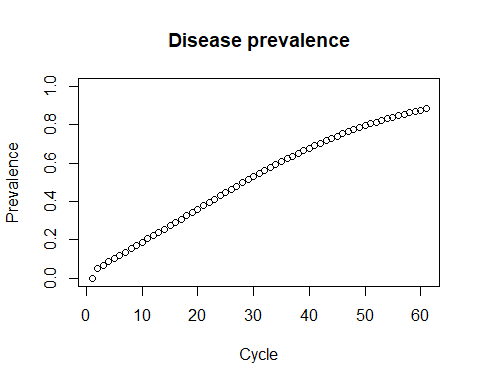


## 06.2.1 Life Expectancy (LE)

v\_le <- sum(v\_os) # summing probablity of OS over time (i.e. life expectancy)

## 06.3 Disease prevalence

v\_prev <- m\_M\_tunnels\_SoC[, "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 cohort trace with the cost vector for the different health states   
v\_tc\_SoC <- m\_M\_tunnels\_SoC %\*% c(c\_H, c\_S, c\_D)   
v\_tc\_trtA <- m\_M\_tunnels\_trtA %\*% c(c\_H + c\_trtA, c\_S, c\_D)   
v\_tc\_trtB <- m\_M\_tunnels\_trtB %\*% c(c\_H + c\_trtB, c\_S, c\_D)   
  
# calculate expected QALYs by multiplying cohort trace with the utilities for the different health states   
v\_tu\_SoC <- m\_M\_tunnels\_SoC %\*% c(u\_H, u\_S, u\_D)   
v\_tu\_trtA <- m\_M\_tunnels\_trtA %\*% c(u\_H, u\_S, u\_D)   
v\_tu\_trtB <- m\_M\_tunnels\_trtB %\*% 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)   
tc\_d\_SoC <- t(v\_tc\_SoC) %\*% v\_dwc  
tc\_d\_trtA <- t(v\_tc\_trtA) %\*% v\_dwc  
tc\_d\_trtB <- t(v\_tc\_trtB) %\*% v\_dwc  
  
# Discount QALYS by multiplying the QALYs vector with discount weights (v\_dw)  
tu\_d\_SoC <- t(v\_tu\_SoC) %\*% v\_dwe  
tu\_d\_trtA <- t(v\_tu\_trtA) %\*% v\_dwe  
tu\_d\_trtB <- t(v\_tu\_trtB) %\*% v\_dwe  
  
# Store them into a vector  
v\_tc\_d <- c(tc\_d\_SoC, tc\_d\_trtA, tc\_d\_trtB)  
v\_tu\_d <- c(tu\_d\_SoC, tu\_d\_trtA, tu\_d\_trtB)  
  
# Dataframe with discounted costs and effectiveness  
df\_ce <- data.frame(Strategy = v\_names\_str,  
 Cost = v\_tc\_d,   
 Effect = v\_tu\_d)  
df\_ce

## Strategy Cost Effect  
## 1 Standard of Care 9786.301 14.64746  
## 2 Treatment A 20857.149 15.92605  
## 3 Treatment B 37387.089 19.60426

## 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

## Strategy Cost Effect Inc\_Cost Inc\_Effect ICER Status  
## 1 Standard of Care 9786.301 14.64746 NA NA NA ND  
## 2 Treatment B 37387.089 19.60426 27600.79 4.956805 5568.262 ND  
## 3 Treatment A 20857.149 15.92605 NA NA NA ED

## 07.4 Plot frontier of the Markov model

plot(df\_cea, effect\_units = "QALYs")

