3-state cSTM in R

With simulation-time dependency

The DARTH workgroup

Developed by the Decision Analysis in R for Technologies in Health (DARTH) workgroup: Fernando Alarid-Escudero, PhD  
Eva A. Enns, MS, PhD M.G. Myriam Hunink, MD, PhD Hawre J. Jalal, MD, PhD Eline Krijkamp, PhD Petros Pechlivanoglou, PhD Alan Yang, MSc

Please cite relevant publications when using this code, see suggestions below.

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

# use this package to conveniently install other packages  
if (!require('pacman')) install.packages('pacman'); library(pacman)   
  
# load (install if required) packages from CRAN  
p\_load("devtools","diagram","dampack")  
  
# load (install if required) packages from GitHub  
#install\_github("DARTH-git/darthtools", force = TRUE) # Uncomment if there is a newer version  
p\_load\_gh("DARTH-git/darthtools")

# 02 Load functions

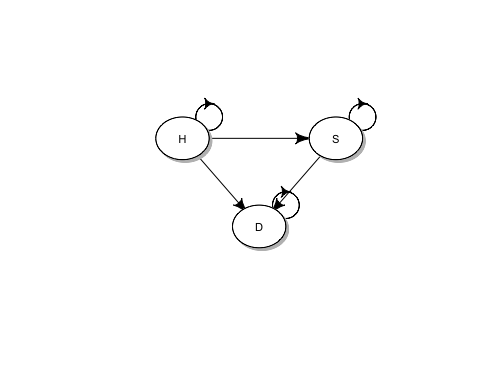
# all functions are in the darthtools package

# 03 Model input

## General setup  
cycle\_length <- 1 # cycle length equal to one year (use 1/12 for monthly)  
n\_cycles <- 60 # number of cycles  
v\_names\_cycles <- paste("cycle", 0:n\_cycles) # cycle names  
v\_names\_states <- c("H", "S", "D") # state names, healthy, sick and dead  
n\_states <- length(v\_names\_states) # number of health states   
  
### Discounting factors   
d\_c <- 0.03 # annual discount rate for costs   
d\_e <- 0.03 # annual discount rate for QALYs, for Dutch setting 1.5% is recommended  
  
### Strategies   
v\_names\_str <- c("Standard of Care", # store the strategy names  
 "Treatment A",   
 "Treatment B")   
n\_str <- length(v\_names\_str) # number of strategies  
  
## Within-cycle correction (WCC) using Simpson's 1/3 rule, alternative method = "half-cycle"  
v\_wcc <- gen\_wcc(n\_cycles = n\_cycles, method = "Simpson1/3")  
  
### Transition intensity rates  
r\_HS\_SoC <- 0.05 # rate of becoming sick when healthy, under standard of care  
r\_HS\_trtA <- 0.04 # rate of becoming sick when healthy, under treatment A  
r\_HS\_trtB <- 0.02 # rate of becoming sick when healthy, under treatment B  
r\_SD <- 0.1 # rate of dying when sick  
r\_base <- 0.003 # rate of dying when healthy at t = 0  
rr\_annual <- 1.1 # annual increase of mortality rate  
# rate of dying when healthy (time-dependent) - this is now a sequence of numbers  
v\_r\_HD <- r\_base \* rr\_annual^(cycle\_length \* (0:(n\_cycles - 1)))  
  
  
### Transition Probabilities  
  
### Converting rates to probabilities  
# function in darthtoolds, rate\_to\_prob based on p = 1 - exp( -r \* cycle\_length)  
p\_HS\_SoC <- rate\_to\_prob(r = r\_HS\_SoC, t = cycle\_length) # probability of becoming sick when healthy, under SoC  
p\_HS\_trtA <- rate\_to\_prob(r = r\_HS\_trtA, t = cycle\_length) # probability of becoming sick when healthy, under treatment A  
p\_HS\_trtB <- rate\_to\_prob(r = r\_HS\_trtB, t = cycle\_length) # probability of becoming sick when healthy, under treatment B  
p\_SD <- rate\_to\_prob(r = r\_SD, t = cycle\_length) # probability of dying when sick  
v\_p\_HD <- rate\_to\_prob(r = v\_r\_HD, t = cycle\_length) # probability of dying when healthy (vector)  
  
### State rewards  
#### Costs   
c\_H <- 400 # cost of one year in healthy state  
c\_S <- 1000 # cost of one year in sick state  
c\_D <- 0 # cost of one year in dead state  
c\_trtA <- 800 # cost of treatment A (per year) in healthy state  
c\_trtB <- 1500 # cost of treatment B (per year) in healthy state  
  
#### Utilities  
u\_H <- 1 # utility when healthy   
u\_S <- 0.5 # utility when sick  
u\_D <- 0 # utility when dead  
  
### Discount weight for costs and effects   
v\_dwc <- 1 / ((1 + (d\_c \* cycle\_length)) ^ (0:n\_cycles))  
v\_dwe <- 1 / ((1 + (d\_e \* cycle\_length)) ^ (0:n\_cycles))

# 04 Construct state-transition models

m\_P\_diag <- matrix(0, nrow = n\_states, ncol = n\_states, dimnames = list(v\_names\_states, v\_names\_states))  
m\_P\_diag["H", "H" ] = ""  
m\_P\_diag["H", "S" ] = ""   
m\_P\_diag["H", "D" ] = ""  
m\_P\_diag["S", "S" ] = ""  
m\_P\_diag["S", "D" ] = ""  
m\_P\_diag["D", "D" ] = ""  
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)



## 04.1 Initial state vector

# All starting healthy  
v\_m\_init <- c("H" = 1, "S" = 0, "D" = 0)   
v\_m\_init

## H S D   
## 1 0 0

## 04.2 Initialize cohort traces

### Initialize cohort trace for SoC. Set all values to zero  
# dimensions are number of cycles + 1 (for initial state) and number of states  
m\_M\_SoC <- matrix(0,   
 nrow = (n\_cycles + 1), ncol = n\_states,   
 dimnames = list(v\_names\_cycles, v\_names\_states))  
# Store the initial state vector in the first row of the cohort trace  
m\_M\_SoC[1, ] <- v\_m\_init  
  
## Initialize cohort traces for treatments A and B  
# Structure and initial states are the same as for SoC  
m\_M\_trtA <- m\_M\_trtB <- m\_M\_SoC

## 04.3 Create transition probability arrays

## Create transition probability arrays for strategy SoC   
### Initialize transition probability array for strategy SoC   
# All transitions to a non-death state are assumed to be conditional on survival  
a\_P\_SoC <- array(0, # Create 3-D array  
 dim = c(n\_states, n\_states, n\_cycles),  
 dimnames = list(v\_names\_states, v\_names\_states,   
 v\_names\_cycles[-length(v\_names\_cycles)])) # name the dimensions of the array   
  
### Fill in array  
## Standard of Care  
# from Healthy  
a\_P\_SoC["H", "H", ] <- (1 - v\_p\_HD) \* (1 - p\_HS\_SoC)  
a\_P\_SoC["H", "S", ] <- (1 - v\_p\_HD) \* p\_HS\_SoC  
a\_P\_SoC["H", "D", ] <- v\_p\_HD  
  
# from Sick  
a\_P\_SoC["S", "S", ] <- 1 - p\_SD  
a\_P\_SoC["S", "D", ] <- p\_SD  
  
# from Dead  
a\_P\_SoC["D", "D", ] <- 1  
  
## Treatment A  
a\_P\_trtA <- a\_P\_SoC # Store array, only replace values that are different in the treatment  
a\_P\_trtA["H", "H", ] <- (1 - v\_p\_HD) \* (1 - p\_HS\_trtA)  
a\_P\_trtA["H", "S", ] <- (1 - v\_p\_HD) \* p\_HS\_trtA  
  
## Treatment B  
a\_P\_trtB <- a\_P\_SoC  
a\_P\_trtB["H", "H", ] <- (1 - v\_p\_HD) \* (1 - p\_HS\_trtB)  
a\_P\_trtB["H", "S", ] <- (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)

## [1] "Valid transition probabilities"

check\_transition\_probability(a\_P\_trtA, verbose = TRUE)

## [1] "Valid transition probabilities"

check\_transition\_probability(a\_P\_trtB, verbose = TRUE)

## [1] "Valid transition probabilities"

# Check that all rows sum to 1  
check\_sum\_of\_transition\_array(a\_P\_SoC, n\_states = n\_states, n\_cycles = n\_cycles, verbose = TRUE)

## [1] "This is a valid transition array"

check\_sum\_of\_transition\_array(a\_P\_trtA, n\_states = n\_states, n\_cycles = n\_cycles, verbose = TRUE)

## [1] "This is a valid transition array"

check\_sum\_of\_transition\_array(a\_P\_trtB, n\_states = n\_states, n\_cycles = n\_cycles, verbose = TRUE)

## [1] "This is a valid transition array"

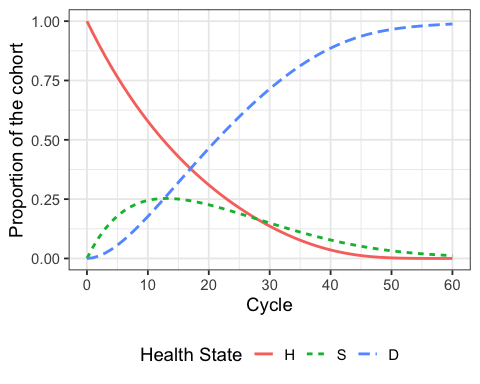
# 05 Run Markov model

# Iterative solution of age-dependent cSTM  
for(t in 1:n\_cycles){  
 ## Fill in cohort trace  
 # For SoC  
 m\_M\_SoC[t + 1, ] <- m\_M\_SoC[t, ] %\*% a\_P\_SoC[, , t]  
 # For strategy A  
 m\_M\_trtA[t + 1, ] <- m\_M\_trtA[t, ] %\*% a\_P\_trtA[, , t]  
 # For strategy B  
 m\_M\_trtB[t + 1, ] <- m\_M\_trtB[t, ] %\*% a\_P\_trtB[, , t]  
}  
  
## Store the cohort traces in a list   
l\_m\_M <- list(SoC = m\_M\_SoC,  
 A = m\_M\_trtA,  
 B = m\_M\_trtB)  
names(l\_m\_M) <- v\_names\_str

# 06 Plot Outputs

## 06.1 Plot the cohort trace for strategies SoC

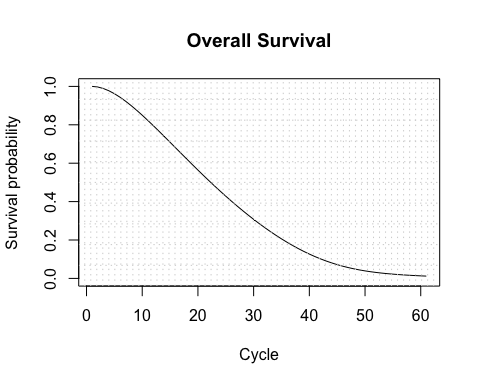
plot\_trace(m\_M\_SoC)



## 06.2 Overall Survival (OS)

Print the overall survival for the Standard of Care

v\_os\_SoC <- 1 - m\_M\_SoC[, "D"] # 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") # create a simple plot showing the OS  
  
# add grid   
grid(nx = n\_cycles, ny = 10, col = "lightgray", lty = "dotted", lwd = par("lwd"),   
 equilogs = TRUE)



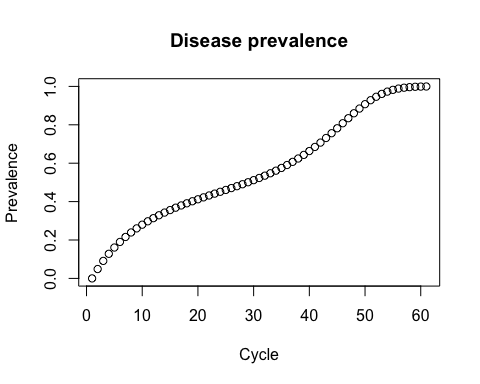
## 06.2.1 Life Expectancy (LE)

le\_SoC <- sum(v\_os\_SoC) # summing probability of OS over time (i.e. life expectancy)  
le\_SoC

## [1] 23.49363

## 06.2.2 Disease prevalence

v\_prev <- m\_M\_SoC[, "S"]/v\_os\_SoC  
plot(v\_prev,  
 ylim = c(0, 1),  
 ylab = "Prevalence",  
 xlab = "Cycle",  
 main = "Disease prevalence")



# 07 State Rewards

## Scale by the cycle length  
  
# Standard of Care  
# vector of state QALYs accrued each cycle  
v\_u\_SoC <- c(H = u\_H,   
 S = u\_S,  
 D = u\_D) \* cycle\_length  
# vector of state costs accrued each cycle  
v\_c\_SoC <- c(H = c\_H,   
 S = c\_S,  
 D = c\_D) \* cycle\_length  
  
# Treatment A  
# vector of state QALYs accrued each cycle  
v\_u\_trtA <- c(H = u\_H,   
 S = u\_S,   
 D = u\_D) \* cycle\_length  
# vector of state costs accrued each cycle  
v\_c\_trtA <- c(H = c\_H + c\_trtA,   
 S = c\_S,   
 D = c\_D) \* cycle\_length  
  
# Treatment B  
# vector of state QALYs accrued each cycle  
v\_u\_trtB <- c(H = u\_H,   
 S = u\_S,   
 D = u\_D) \* cycle\_length  
# vector of state costs accrued each cycle  
v\_c\_trtB <- c(H = c\_H + c\_trtB,   
 S = c\_S,   
 D = c\_D) \* cycle\_length  
  
## Store state rewards   
# Store the vectors of state utilities for each strategy in a list   
l\_u <- list(SoQ = v\_u\_SoC,  
 A = v\_u\_trtA,  
 B = v\_u\_trtB)  
# Store the vectors of state cost for each strategy in a list   
l\_c <- list(SoQ = v\_c\_SoC,  
 A = v\_c\_trtA,  
 B = v\_c\_trtB)  
  
# assign strategy names to matching items in the lists  
names(l\_u) <- names(l\_c) <- v\_names\_str

# 08 Compute expected outcomes

# Create empty vectors to store total utilities and costs   
v\_tot\_qaly <- v\_tot\_cost <- vector(mode = "numeric", length = n\_str)  
names(v\_tot\_qaly) <- names(v\_tot\_cost) <- v\_names\_str  
  
## Loop through each strategy and calculate total utilities and costs   
for (i in 1:n\_str) {  
 v\_u\_str <- l\_u[[i]] # select the vector of state utilities for the i-th strategy  
 v\_c\_str <- l\_c[[i]] # select the vector of state costs for the i-th strategy  
   
 ### Expected QALYs and costs per cycle   
 ## Vector of QALYs and Costs  
 # Apply state rewards   
 v\_qaly\_str <- l\_m\_M[[i]] %\*% v\_u\_str # sum the utilities of all states for each cycle  
 v\_cost\_str <- l\_m\_M[[i]] %\*% v\_c\_str # sum the costs of all states for each cycle  
   
 ### Discounted total expected QALYs and Costs per strategy and apply within-cycle correction if applicable  
 # QALYs  
 v\_tot\_qaly[i] <- t(v\_qaly\_str) %\*% (v\_dwe \* v\_wcc)  
 # Costs  
 v\_tot\_cost[i] <- t(v\_cost\_str) %\*% (v\_dwc \* v\_wcc)  
}

# 09 Cost-effectiveness analysis (CEA)

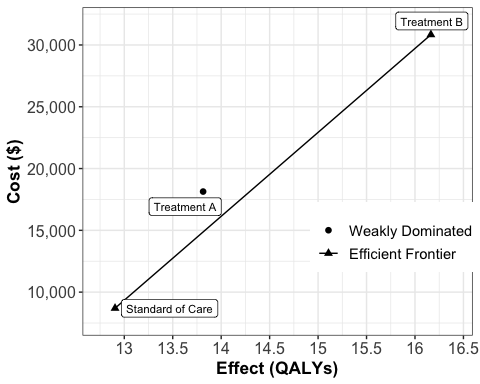
## Incremental cost-effectiveness ratios (ICERs)   
df\_cea <- calculate\_icers(cost = v\_tot\_cost,   
 effect = v\_tot\_qaly,  
 strategies = v\_names\_str)  
df\_cea

## Strategy Cost Effect Inc\_Cost Inc\_Effect  
## Standard of Care Standard of Care 8696.761 12.90463 NA NA  
## Treatment B Treatment B 30834.091 16.16428 22137.33 3.259648  
## Treatment A Treatment A 18138.224 13.81353 NA NA  
## ICER Status  
## Standard of Care NA ND  
## Treatment B 6791.325 ND  
## Treatment A NA ED

## CEA table in proper format   
table\_cea <- format\_table\_cea(df\_cea)   
table\_cea

## Strategy Costs ($) QALYs Incremental Costs ($)  
## Standard of Care Standard of Care 8,697 12.90 <NA>  
## Treatment B Treatment B 30,834 16.16 22,137  
## Treatment A Treatment A 18,138 13.81 <NA>  
## Incremental QALYs ICER ($/QALY) Status  
## Standard of Care NA <NA> ND  
## Treatment B 3.26 6,791 ND  
## Treatment A NA <NA> ED

## CEA frontier   
plot(df\_cea, label = "all", txtsize = 14) +  
 expand\_limits(x = max(table\_cea$QALYs) + 0.1) +  
 theme(legend.position = c(0.8, 0.3))



We kindly request you to add the following Acknowledgement paragraph to your further work where DARTH code formed the basis. We also like to remind you that you can add other sources of reference to this paragraph to acknowledge code you got from others.

# Acknowlegdement

For this work we made use of the template developed by the Decision Analysis in R for Technologies in Health (DARTH) workgroup: <http://darthworkgroup.com>.

The notation of our code is based on the following provided framework and coding convention: Alarid-Escudero, F., Krijkamp, E., Pechlivanoglou, P. et al. A Need for Change! A Coding Framework for Improving Transparency in Decision Modeling. PharmacoEconomics 37, 1329–1339 (2019). <https://doi.org/10.1007/s40273-019-00837-x>.

* Alarid-Escudero F, Krijkamp EM, Enns EA, Yang A, Hunink MGM Pechlivanoglou P, Jalal H. An Introductory Tutorial on Cohort State-Transition Models in R Using a Cost-Effectiveness Analysis Example. Medical Decision Making, 2023; 43(1). (Epub). <https://doi.org/10.1177/0272989X221103163>
* Alarid-Escudero F, Krijkamp EM, Enns EA, Yang A, Hunink MGM Pechlivanoglou P, Jalal H. A Tutorial on Time-Dependent Cohort State-Transition Models in R using a Cost-Effectiveness Analysis Example. Medical Decision Making, 2023; 43(1). <https://doi.org/10.1177/0272989X221121747>

Other work from DARTH can be found on the website: <http://darthworkgroup.com/publications/>

# Copyright for assignment work

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.