

Survival Analysis - Sick-Sicker model

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)

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. The Hospital for Sick Children, Toronto and 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. *BioRxiv* 670612 2019.<https://www.biorxiv.org/content/10.1101/670612v1>

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  
# load (install if required) packages from CRAN  
p_load("here", "dplyr", "devtools", "gems", "flexsurv", "survHE", "ggplot2", "msm", "igraph", "mstate",  
# load (install if required) packages from GitHub  
# install_github("DARTH-git/dampack", force = TRUE) Uncomment if there is a newer version  
p_load_gh("DARTH-git/dampack")
```

02 Load functions

```
source("survival_functions.R")
```

03 Input model parameters

```
set.seed(1)                # set the seed  
v_n      <- c("S1", "S2", "D") # the model states names  
  
# Model structure  
n_t      <- 30                # time horizon, 30 cycles  
c_l      <- 1                 # discount rate of 3% per cycle  
d_r      <- 0.03  
  
p_S2D    <- 0.2                # probability of dying in sicker state  
  
v_init   <- c("S1" = 1,  
              "S2" = 0,  
              "D"  = 0)        # initial cohort distribution (everyone allocated to the  
                                # "S1" state)  
  
# Cost inputs  
c_H      <- 2000               # cost of one cycle in the healthy state  
c_S1     <- 4000               # cost of one cycle in the sick state  
c_S2     <- 15000              # cost of one cycle in the sicker state  
c_D      <- 0                  # cost of one cycle in the dead state  
c_Trt    <- 12000              # cost of treatment (per cycle)  
  
# Utility inputs  
u_H      <- 1                  # utility when healthy  
u_S1     <- 0.75               # utility when sick  
u_S2     <- 0.5                # utility when sicker
```

```

u_D    <- 0                                # utility when dead
u_Trtr <- 0.95                             # utility when sick(er) and being treated

v_dw    <- 1 / ((1 + d_r) ^ (0:n_t))      # discount weight
n_states <- length(v_n)                    # the number of health states
v_names_str <- c("no treatment", "treatment") # strategy names

times    <- seq(0, n_t, c_l)               # the cycles in years

```

Survival analysis component

```

# Load the Sicker data
data_long <- read.csv("data_long_Sicker.csv")
head(data_long)

# Models can be fitted independently for each transition. This is more flexible!

# Create subsets for each transition
data_S1D <- data_long[data_long$from == "S1" & data_long$to == "D", ]
data_S1S2 <- data_long[data_long$from == "S1" & data_long$to == "S2", ]

# Fit independent models for each transition and pick one that fits best
fit_S1S2 <- fit.fun(time = "time", status = "status", data = data_S1S2, times = times)
fit_S1D <- fit.fun(time = "time", status = "status", data = data_S1D, times = times)

# Extract the transition probabilities from the fitted survival models
p_S1S2 <- trans_prob(fit_S1S2, choose_dist = "log-Logistic", times=times)$t.p
p_S1D <- trans_prob(fit_S1D, choose_dist = "Exponential", times=times)$t.p

```

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, ] <- v_init # 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 Sick
a_P["S1", "S1", ] <- (1 - p_S1D) * (1 - p_S1S2)
a_P["S1", "S2", ] <- (1 - p_S1D) * p_S1S2
a_P["S1", "D", ] <- p_S1D

# from Sicker
a_P["S2", "S2", ] <- 1 - p_S2D
a_P["S2", "D", ] <- p_S2D

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

04.3 Check if transition array and probabilities are valid

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

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
matplot(m_M, type = 'l')
```

07 Compute Cost-Effectiveness Outcomes

```
# Vectors with costs and utilities by treatment
v_u_notrt <- c(u_S1, u_S2, u_D)
v_u_trt <- c(u_Trt, u_S2, u_D)

v_c_notrt <- c(c_S1, c_S2, c_D)
v_c_trt <- c(c_S1 + c_Trt, c_S2 + c_Trt, c_D)
```

07.1 Mean Costs and QALYs for Treatment and NO Treatment

```

v_tu_notrt <- m_M %>% v_u_notrt
v_tu_trt   <- m_M %>% v_u_trt

v_tc_notrt <- m_M %>% v_c_notrt
v_tc_trt   <- m_M %>% v_c_trt

```

07.2 Discounted Mean Costs and QALYs

```

tu_d_notrt <- t(v_tu_notrt) %>% v_dw
tu_d_trt   <- t(v_tu_trt)   %>% v_dw

tc_d_notrt <- t(v_tc_notrt) %>% v_dw
tc_d_trt   <- t(v_tc_trt)   %>% v_dw

# store them into a vector
v_tc_d <- c(tc_d_notrt, tc_d_trt)
v_tu_d <- c(tu_d_notrt, 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")

```