

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",
       "reshape2", "knitr")
# 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("functions.R")
```

03 Input model parameters

```
set.seed(1)          # set the seed
v_n  <- c("H", "S1", "S2", "D") # the model states names

# Model structure
n_t  <- 30            # time horizon, 30 cycles
c_l  <- 1
n_i  <- 100000        # number of simulated individuals
n_s  <- length(v_n)   # the number of health states
d_r  <- 0.03          # discount rate of 3% per cycle
v_dw <- 1 / ((1 + d_r) ^ (0:n_t)) # discount weight
v_names_str <- c("no treatment", "treatment") # strategy names

# Event probabilities (per cycle)
# Annual transition probabilities
p_HS1 <- 0.15         # probability of becoming sick when healthy

# Annual probabilities of death
# load age dependent probability
p_mort <- read.csv(here::here("data", "mortProb_age.csv"))
# load age distribution
dist_Age <- read.csv(here::here("data", "MyPopulation-AgeDistribution.csv"))

# 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_Trtr  <- 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

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

```

04 Sample individual level characteristics

04.1 Static characteristics

```

set.seed(2019) # set the seed for the simulation of individual characteristics
v_x           <- runif(n_i, min = 0.95, max = 1.05) # treatment effect modifier at baseline

# sample from age distribution an initial age for every individual
v_age0       <- sample(x = dist_Age$age, prob = dist_Age$prop, size = n_i, replace = TRUE)

# store the information at baseline into a data frame.
df_X         <- data.frame(ID = 1:n_i, x = v_x, Age = v_age0)

```

Survival analysis component

```

# load the Sicker data
data_long <- read.csv(here::here("data", "data_long_Sicker.csv"), row.names = 1)
head(data_long)

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

# Create subsets for each transition
data_S1H <- subset(data_long, trans == 1)
data_S1S2 <- subset(data_long, trans == 2)
data_S1D <- subset(data_long, trans == 3)
data_S2D <- subset(data_long, trans == 4)

# fit independent models for each transition and pick the one with the lowest AIC
fit_S1H <- fit.fun(time = "time", status = "status", data = data_S1H, times = times,
  extrapolate = F)
fit_S1S2 <- fit.fun(time = "time", status = "status", data = data_S1S2, times = times,
  extrapolate = F)
fit_S1D <- fit.fun(time = "time", status = "status", data = data_S1D, times = times,
  extrapolate = F)
fit_S2D <- fit.fun(time = "time", status = "status", data = data_S2D, times = times,
  extrapolate = F)

```

```
best.fit_S1H <- fit_S1H [[which.min(fit_S1H$AIC)]]
best.fit_S1S2 <- fit_S1S2[[which.min(fit_S1S2$AIC)]]
best.fit_S1D <- fit_S1D [[which.min(fit_S1D$AIC)]]
best.fit_S2D <- fit_S2D [["Weibull"]]

# Extract transition probabilities from the best fitting models
p_S1H <- flexsurvreg_prob(object = best.fit_S1H, t = times)
p_S1S2 <- flexsurvreg_prob(object = best.fit_S1S2, t = times)
p_S1D <- flexsurvreg_prob(object = best.fit_S1D, t = times)
p_S2D <- flexsurvreg_prob(object = best.fit_S2D, t = times)
```

04.2 Dynamic characteristics

```
# Specify the initial health state of the individuals
# everyone begins in the healthy state (in this example)
v_M_init <- rep("H", n_i) # a vector with the initial health state for all individuals
v_Ts1_init <- rep(0, n_i) # a vector with the time of being sick at the start of the model
v_Ts2_init <- rep(0, n_i) # a vector with the time of being sick at the start of the model
```

05 Define Simulation Functions

05.1 Probability function

The function that updates the transition probabilities of every cycle is shown below.

```
# your turn
```

05.2 Cost function

The Costs function estimates the costs at every cycle.

```
# your turn
```

05.3 Health outcome function

The Effs function to update the utilities at every cycle.

```
# your turn
```

06 Run Microsimulation

```
# your turn
```

07 Visualize results

your turn

08 Cost Effectiveness Analysis

your turn