

Simple 3-state Partitioned Survival 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)

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", "survminer", "survHE", "ggplot2", "msm", "igraph")
```

02 Load functions

```
source("survival_functions_ZIN.R")
```

03 Input model parameters

```
v_n      <- c("healthy", "sick", "dead")  # state names
n_i      <- 50000                          # number of simulations
c_l      <- 1 / 12                         # cycle length (a month)
n_t      <- 30                            # number of years (20 years)
set.seed(2020)                            # set the seed
n_sim    <- 100                           # number of simulations

n_s      <- length(v_n)                   # No of states
times    <- seq(0, n_t, c_l)              # the cycles in years
```

Create a transition probability matrix with all transitions indicated and numbered.

```
tmat <- matrix(NA, n_s, n_s, dimnames = list(v_n, v_n))
tmat["healthy", "sick"] <- 1
tmat["healthy", "dead"] <- 2
tmat["sick", "dead"] <- 3

layout.fig <- c(2,1)
plotmat(t(tmat), t(layout.fig), self.cex = 0.5, curve = 0, arr.pos = 0.76,
        latex = T, arr.type = "curved", relsize = 0.85, box.prop=0.8,
        cex = 0.1, box.cex = 0.7, lwd = 1)
```

Generate data.

```
source("data.R")
head(true_data)
head(sim_data)
head(status)
head(OS_PFS_data)
```

04 Analysis

Showcasing the use of packages `survival`, `flexsurv`.

```
fit_KM <- survfit(Surv(time = OS_time, event = OS_status) ~ 1, data = OS_PFS_data)
plot(fit_KM, mark.time = T)

# a prettier way of plotting!!

ggsurvplot(
  fit_KM,
  data = OS_PFS_data,
  size = 1,                      # change line size
  palette = c("orange2"),        # custom color palettes
  conf.int = TRUE,               # Add confidence interval
  pval = TRUE,                   # Add p-value
  risk.table = TRUE,             # Add risk table
  risk.table.height = 0.25,      # Useful to change when you have multiple groups
  ggtheme = theme_bw(),          # Change ggplot2 theme
  xlab = 'Time in days',         # Change X-axis label
  title = "Survival curve for Progression-Free Survival (PFS)",
  subtitle = "Based on Kaplan-Meier estimates"
)
```

04.1 Partitioned Survival model

```
# R package flexsurv allows parametric fitting of curves
fit_weib <- flexsurvreg(Surv(time = OS_time, event = OS_status) ~ 1, data = OS_PFS_data, dist = "weibull")
plot(fit_weib)

# fit all parametric models to the data and extract the AIC/BIC.
# Select the one with the most appropriate fit
# Repeat for PFS and OS
fit_PFS <- fit.fun(time = "PFS_time", status = "PFS_status", data = OS_PFS_data,
  extrapolate = TRUE, times = times)
fit_OS <- fit.fun(time = "OS_time", status = "OS_status", data = OS_PFS_data,
  extrapolate = TRUE, times = times)

# Check AIC of each model to assess goodness-of-fit
GoF_OS <- data.frame(AIC = fit_OS$AIC, BIC = fit_OS$BIC)
GoF_PFS <- data.frame(AIC = fit_PFS$AIC, BIC = fit_PFS$BIC)

# "Exponential", "Weibull (AFT)", "Gamma", "log-Normal", "log-Logistic", "Gen. Gamma"
choose_PFS <- "Weibull (AFT)"
choose_OS <- "Weibull (AFT)"

# construct a partitioned survival model out of the fitted models
m_M_PSM <- partsurv(fit_PFS, fit_OS, time = times, choose_PFS = choose_PFS,
  choose_OS = choose_OS, n_sim = 100)

# plot the results of PSM and the trace
m_M_data <- transitionProbabilities(generate$cohort, times = times)$probabilities
```

```

matplot(times, m_M_data, type='l', lty = 1, col = 1,
        ylab = "Proportion of cohort", xlab = "Time",
        main = "Trace comparisons", xlim=c(0,25))
# matlines(times, m_M_PSM$trace, col = 4, lty = 1) # Uncomment if want to use deterministic
matlines(times, m_M_PSM$Mean, col = 4, lty = 1)
matlines(times, m_M_PSM$CI[,,"low"], col = 4, lty = 2)
matlines(times, m_M_PSM$CI[,,"high"], col = 4, lty = 2)
legend("right", c("True Data", "PSM", "Low CI", "High CI"),
      col = c(1,4,4,4), lty = c(1,1,2,2), bty= "n")

```