Simple 3-state Markov model in R

with dependency for time-since model start AND with 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. 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. 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. Med Decis Making. 2020 Online first. https://doi.org/10.1177/0272989X19893973

```r
rm(list = ls())      # clear memory (removes all the variables from the workspace)
```

01 Load packages

```r
# no packages required
```

02 Load functions

```r
# no functions required
```

03 Input model parameters

```r
# Strategy names
v_names_str <- c("Base Case")

# Number of strategies
n_str <- length(v_names_str)

# Markov model parameters
v_n  <- c("Healthy", "Sick", "Dead")    # state names
n_states  <- length(v_n)                # number of states
n_t  <- 60                              # number of cycles

# Tunnels
n_tunnel_size <- n_t
# Sick state
v_Sick_tunnels <- paste("Sick_", seq(1, n_tunnel_size), "Yr", sep = "")
# Create variables for time-dependent model
v_n_tunnels  <- c("Healthy", v_Sick_tunnels, "Dead")  # state names
n_states_tunnels  <- length(v_n_tunnels)              # number of states

p_HD <- seq(0.003, 0.01, length.out = n_t)  # probability of dying when sick (age-dependent) - this is
p_HS <- 0.05                                # probability of becoming sick when healthy, conditioned on
p_SD <- 0.1                                 # probability of dying when sick

# Weibull parameters
l <- 0.08
g <- 1.1
p_SD <- l*g*(1:n_tunnel_size)^{g-1}         # probability of dying when sick (time-in-state dependent)

# 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
u_H  <- 0.8                                 # utility when healthy
u_S  <- 0.5                                 # utility when sick
u_D  <- 0                                   # utility when dead
```

```r
d_e <- d_c <- 0.03                          # equal discount of costs and QALYs by 3%

# calculate discount weights for costs for each cycle based on discount rate d_c
v_dwc <- 1 / (1 + d_e) ^ (0:n_t)
# calculate discount weights for effectiveness for each cycle based on discount rate d_e
v_dwe <- 1 / (1 + d_c) ^ (0:n_t)
```

04 Define and initialize matrices and vectors

04.1 Cohort trace

```r
m_M <- matrix(NA,
              nrow = n_t + 1,  # create Markov trace (n_t + 1 because R doesn't understand
                               # Cycle 0)
              ncol = n_states_tunnels,
              dimnames = list(0:n_t, v_n_tunnels))

# The cohort starts as healthy
# initialize first cycle of Markov trace accounting for the tunnels
m_M[1, ] <- c(1, rep(0, n_tunnel_size), 0)
```

04.2 Transition probability array

```r
# create the transition probability array
a_P <- array(0,                                          # Create 3-D array
             dim = c(n_states_tunnels, n_states_tunnels, n_t),
             dimnames = list(v_n_tunnels, v_n_tunnels, 0:(n_t-1)))
```

Fill in the transition probability array:

```r
# from Healthy
a_P["Healthy", "Healthy", ]  <- 1 - p_HD - p_HS
a_P["Healthy", "Sick_1Yr", ] <- p_HS
a_P["Healthy", "Dead", ]     <- p_HD

# from Sick
for(i in 1:(n_tunnel_size - 1)){
  a_P[v_Sick_tunnels[i], v_Sick_tunnels[i + 1], ] <- 1 - p_SD[i]
  a_P[v_Sick_tunnels[i], "Dead", ] <- p_SD[i]
}

a_P[v_Sick_tunnels[n_tunnel_size], v_Sick_tunnels[n_tunnel_size], ] <- 1 - p_SD[n_tunnel_size]
a_P[v_Sick_tunnels[n_tunnel_size], "Dead", ] <- p_SD[n_tunnel_size]

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

04.3 Check if transition array and probabilities are valid

```
# Check if transition matrix is valid (i.e., each row should add up to 1)
valid <- apply(a_P, 3, function(x) sum(rowSums(x))==n_states_tunnels)
if (!isTRUE(all.equal(as.numeric(sum(valid)), as.numeric(n_t)))) {
  stop("This is not a valid transition Matrix")
}
```

05 Run Markov model

```
for (t in 1:n_t) {                          # 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)
```

```
##     Healthy   Sick_1Yr   Sick_2Yr   Sick_3Yr   Sick_4Yr   Sick_5Yr
## 0 1.0000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 1 0.9470000 0.05000000 0.00000000 0.00000000 0.00000000 0.00000000
## 2 0.8966966 0.04735000 0.04560000 0.00000000 0.00000000 0.00000000
## 3 0.8489589 0.04483483 0.04318320 0.04129919 0.00000000 0.00000000
## 4 0.8036620 0.04244795 0.04088937 0.03911033 0.03724283 0.00000000
## 5 0.7606865 0.04018310 0.03871253 0.03703284 0.03526896 0.03347812
##   Sick_6Yr Sick_7Yr Sick_8Yr Sick_9Yr Sick_10Yr Sick_11Yr Sick_12Yr
## 0        0        0        0        0         0         0         0
## 1        0        0        0        0         0         0         0
## 2        0        0        0        0         0         0         0
## 3        0        0        0        0         0         0         0
## 4        0        0        0        0         0         0         0
## 5        0        0        0        0         0         0         0
##   Sick_13Yr Sick_14Yr Sick_15Yr Sick_16Yr Sick_17Yr Sick_18Yr Sick_19Yr
## 0         0         0         0         0         0         0         0
## 1         0         0         0         0         0         0         0
## 2         0         0         0         0         0         0         0
## 3         0         0         0         0         0         0         0
## 4         0         0         0         0         0         0         0
## 5         0         0         0         0         0         0         0
##   Sick_20Yr Sick_21Yr Sick_22Yr Sick_23Yr Sick_24Yr Sick_25Yr Sick_26Yr
## 0         0         0         0         0         0         0         0
## 1         0         0         0         0         0         0         0
## 2         0         0         0         0         0         0         0
## 3         0         0         0         0         0         0         0
## 4         0         0         0         0         0         0         0
## 5         0         0         0         0         0         0         0
##   Sick_27Yr Sick_28Yr Sick_29Yr Sick_30Yr Sick_31Yr Sick_32Yr Sick_33Yr
## 0         0         0         0         0         0         0         0
## 1         0         0         0         0         0         0         0
## 2         0         0         0         0         0         0         0
## 3         0         0         0         0         0         0         0
## 4         0         0         0         0         0         0         0
```

```
## 5         0         0         0         0         0         0         0
##   Sick_34Yr Sick_35Yr Sick_36Yr Sick_37Yr Sick_38Yr Sick_39Yr Sick_40Yr
## 0         0         0         0         0         0         0         0
## 1         0         0         0         0         0         0         0
## 2         0         0         0         0         0         0         0
## 3         0         0         0         0         0         0         0
## 4         0         0         0         0         0         0         0
## 5         0         0         0         0         0         0         0
##   Sick_41Yr Sick_42Yr Sick_43Yr Sick_44Yr Sick_45Yr Sick_46Yr Sick_47Yr
## 0         0         0         0         0         0         0         0
## 1         0         0         0         0         0         0         0
## 2         0         0         0         0         0         0         0
## 3         0         0         0         0         0         0         0
## 4         0         0         0         0         0         0         0
## 5         0         0         0         0         0         0         0
##   Sick_48Yr Sick_49Yr Sick_50Yr Sick_51Yr Sick_52Yr Sick_53Yr Sick_54Yr
## 0         0         0         0         0         0         0         0
## 1         0         0         0         0         0         0         0
## 2         0         0         0         0         0         0         0
## 3         0         0         0         0         0         0         0
## 4         0         0         0         0         0         0         0
## 5         0         0         0         0         0         0         0
##   Sick_55Yr Sick_56Yr Sick_57Yr Sick_58Yr Sick_59Yr Sick_60Yr       Dead
## 0         0         0         0         0         0         0 0.00000000
## 1         0         0         0         0         0         0 0.00300000
## 2         0         0         0         0         0         0 0.01035336
## 3         0         0         0         0         0         0 0.02172383
## 4         0         0         0         0         0         0 0.03664758
## 5         0         0         0         0         0         0 0.05463798
```

Create aggregated trace.

```
m_M_tunnels <- cbind(Healthy = m_M[, "Healthy"],
                Sick = rowSums(m_M[, 2:(n_tunnel_size + 1)]),
                Dead = m_M[, "Dead"])
head(m_M_tunnels) # show the first rows of the aggregated Markov trace
```

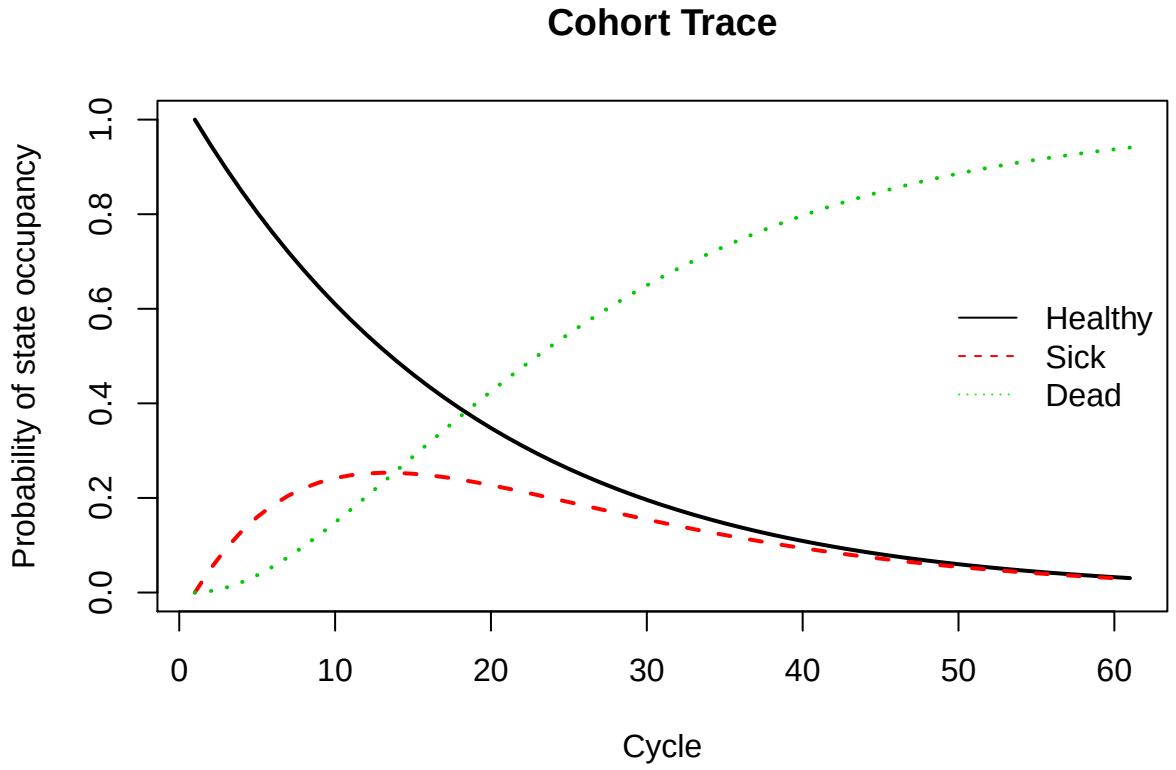
```
##     Healthy      Sick       Dead
## 0 1.0000000 0.0000000 0.00000000
## 1 0.9470000 0.0500000 0.00300000
## 2 0.8966966 0.0929500 0.01035336
## 3 0.8489589 0.1293172 0.02172383
## 4 0.8036620 0.1596905 0.03664758
## 5 0.7606865 0.1846755 0.05463798
```

06 Compute and Plot Epidemiological Outcomes

06.1 Cohort trace

```
# create a plot of the data
matplot(m_M_tunnels, type = 'l',
        ylab = "Probability of state occupancy",
        xlab = "Cycle",
```

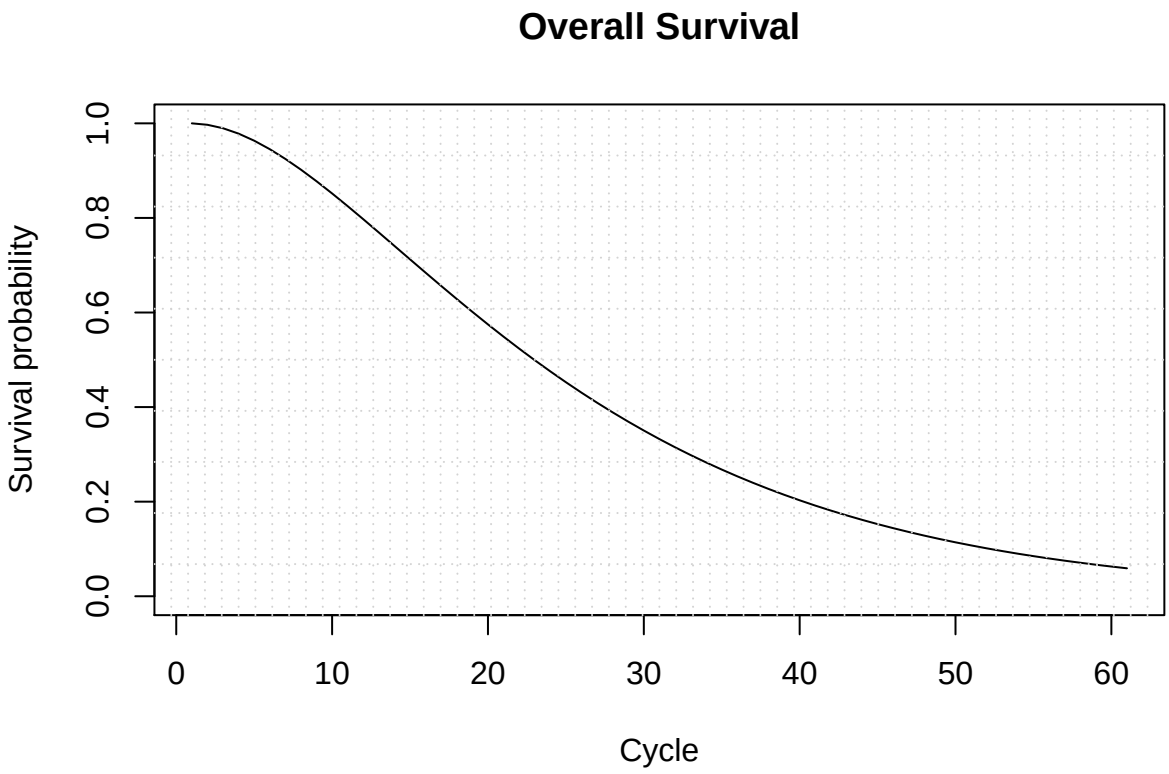
```
       main = "Cohort Trace", lwd = 2)
# add a legend to the graph
legend("right", v_n, col = c("black", "red", "green"), lty = 1:3, bty = "n")
```



06.2 Overall Survival (OS)

```
v_os <- 1 - m_M_tunnels[, "Dead"]    # calculate the overall survival (OS) probability
v_os <- rowSums(m_M_tunnels[, 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_t, ny = 10, col = "lightgray", lty = "dotted", lwd = par("lwd"), equilogs = TRUE)
```

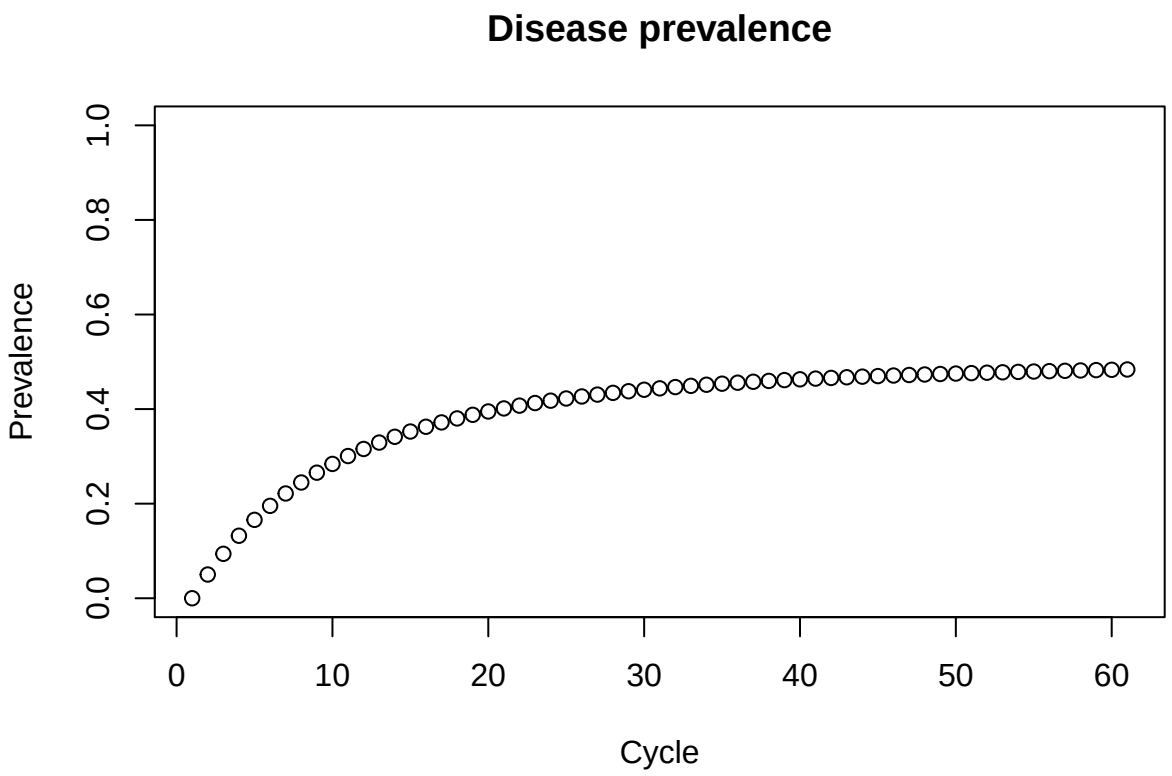


06.2.1 Life Expectancy (LE)

```
v_le <- sum(v_os)           # summing probablity of OS over time  (i.e. life expectancy)
```

06.3 Disease prevalence

```
v_prev <- m_M_tunnels[, "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 m_M with the cost vector for the different
# health states
v_tc <- m_M_tunnels %*% c(c_H, c_S, c_D)
# calculate expected QALYs by multiplying m_M with the utilities for the different
# health states
v_tu <- m_M_tunnels %*% 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)
v_tc_d <-  t(v_tc) %*% v_dwc
# Discount QALYS  by multiplying the QALYs vector with discount weights (v_dw)
v_te_d <-  t(v_tu) %*% v_dwe
```

07.3 Results

```
results <- data.frame( "Total Discounted Cost" = v_tc_d,
                       "Life Expectancy" = v_le,
                       "Total Discounted QALYs" = v_te_d,
```

```
                        check.names = F)
results
```

```
##   Total Discounted Cost Life Expectancy Total Discounted QALYs
## 1              9382.938        25.89709                11.98968
```