Decision Analysis in R for Technologies in Health

Appendix C

Microsimulation modeling for health decision sciences using R: A tutorial.

March 2018

Fernando Alarid-Escudero, PhD[1]
Eva A. Enns, MS, PhD[1]
M.G. Myriam Hunink, MD, PhD[2,3]
Hawre J. Jalal, MD, PhD[4]
Eline M. Krijkamp, MSc[2]
Petros Pechlivanoglou, PhD[5]

Appendix C - Microsimulation modeling for health decision sciences using R: A tutorial.

Krijkamp EM, Alarid-Escudero F, Enns EA, Jalal HJ, Hunink MGM, Pechlivanoglou P.

2018

This code forms the basis for the microsimulation model of the article: 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.

Please cite the article when using this code.

See GitHub for more information or code updates https://github.com/DARTH-git/Microsimulation-tutorial

To program this tutorial we made use of R: 3.3.0 GUI 1.68 Mavericks build (7202) RStudio: Version 1.0.136 2009-2016 RStudio, Inc.

Code of Appendix C

```r
# rm(list = ls())  # remove any variables in R's memory
```

Model input

```r
n.t   <- 30                  # time horizon, 30 cycles
v.n   <- c("H","S1","S2","D") # the model states: Healthy (H), Sick (S1), Sicker (S2), Dead (D)
n.s   <- length(v.n)         # the number of health states
v.M_1 <- c(1, 0, 0, 0)       # everyone begins in the healthy state
d.c   <- d.e <- 0.03         # equal discounting of costs and QALYs by 3%
v.Trt <- c("No Treatment", "Treatment") # store the strategy names
```

Transition probabilities (per cycle)

```r
p.HD    <- 0.005             # probability to die when healthy
p.HS1   <- 0.15              # probability to become sick when healthy
p.S1H   <- 0.5               # probability to become healthy when sick
p.S1S2  <- 0.105             # probability to become sicker when sick
rr.S1   <- 3                 # rate ratio of death when sick vs healthy
rr.S2   <- 10                # rate ratio of death when sicker vs healthy
r.HD    <- -log(1 - p.HD)    # rate of death when healthy
r.S1D   <- rr.S1 * r.HD      # rate of death when sick
r.S2D   <- rr.S2 * r.HD      # rate of death when sicker
p.S1D   <- 1 - exp(-r.S1D)   # probability to die when sick
p.S2D   <- 1 - exp(-r.S2D)   # probability to die when sikcer
```

Cost and utility inputs

```
c.H     <- 2000                    # cost of remaining for one cycle healthy
c.S1    <- 4000                    # cost of remaining for one cycle sick
c.S2    <- 15000                   # cost of remaining for one cycle sicker
c.Trt   <- 12000                   # cost of treatment (per cycle)

u.H     <- 1                       # utility when healthy
u.S1    <- 0.75                    # utility when sick
u.S2    <- 0.5                     # utility when sicker
u.Trt   <- 0.95                    # utility when being treated
```

Markov Model

The Markov function for the 'Sick-Sicker' model keeps track of what happens to the cohort during each cycle.

```
Markov <- function(v.M_1, n.t, v.n, d.c, d.e, Trt = FALSE) {
# Arguments:
  # v.M_1:   initial allocation of cohort across states
  # n.t:     total number of cycles to run the model
  # v.n:     vector of health state names
  # d.c:     discount rate for costs
  # d.e:     discount rate for effectiveness (QALYs)
  # Trt:     is this the cohort receiving treatment? (default is FALSE)

  v.dwc <- 1 / (1 + d.c) ^ (0:n.t)   # calculate the cost discount weight bas
ed on the discount rate d.c
  v.dwe <- 1 / (1 + d.e) ^ (0:n.t)   # calculate the QALY discount weight bas
ed on the discount rate d.e

# create transition probability matrix
m.P <- matrix(c(1 - (p.HS1 + p.HD), p.HS1, 0,   p.HD,
                p.S1H, 1 - (p.S1H + p.S1S2 + p.S1D), p.S1S2,   p.S1D,
                0, 0, 1 - p.S2D,   p.S2D,
                0, 0, 0, 1),
                nrow = n.s, ncol = n.s, byrow = T,
                 dimnames = list (v.n, v.n))

# create the transition trace matrix (m.TR) capturing the proportion of the c
ohort in each state at each time point
m.TR <-  matrix(0, nrow = n.t + 1, ncol = n.s,
                dimnames = list( paste("cycle", 0:n.t, sep = ""), v.n))

m.TR[1, ]  <- v.M_1  # indicate the initial health state

# create vectors of utility and costs for each state
v.c <- c(c.H, c.S1 + c.Trt * Trt, c.S2 + c.Trt * Trt, 0)
v.u <- c(u.H, Trt * u.Trt + (1 - Trt) * u.S1, u.S2, 0)
```

```r
# run the simulation
for (i in 2:(n.t + 1)){
# calculate the proportion of the cohort in each state at time t
    m.TR[i, ] <- t(m.TR[i - 1, ]) %*% m.P
}  # close the loop for the individuals

tc <- m.TR %*% v.c  # calculate the costs
te <- m.TR %*% v.u  # calculate the QALYs

tc_hat <- t(tc) %*% v.dwc  # total (discounted) cost
te_hat <- t(te) %*% v.dwe  # total (discounted) QALY

results <- list(m.TR = m.TR, tc_hat = tc_hat, te_hat = te_hat) # store the re
sults from the simulation in a list
return(results)  # return the results
}
```

Run the simulation

```r
sim_markov_no_trt <- Markov(v.M_1, n.t, v.n, d.c, d.e, Trt = FALSE) # run for
no treatment
sim_markov_trt    <- Markov(v.M_1, n.t, v.n, d.c, d.e, Trt = TRUE)  # run for
treateement
```

Cost-effectiveness analysis

```r
# store the mean costs of each strategy in a new variable C (vector costs)
v.C <- c(sim_markov_no_trt$tc_hat, sim_markov_trt$tc_hat)
# store the mean QALYs of each strategy in a new variable E (vector effects)
v.E <- c(sim_markov_no_trt$te_hat, sim_markov_trt$te_hat)

delta.C <- v.C[2] - v.C[1]              # calculate incremental costs
delta.E <- v.E[2] - v.E[1]              # calculate incremental QALYs
ICER <- delta.C / delta.E              # calculate the ICER
results <- c(delta.C, delta.E, ICER)  # store the values in a new variable
```

Create full incremental cost-effectiveness analysis table

```r
table_markov <- data.frame(
  round(v.C, 0),              # costs per arm
  round(v.E, 3),              # health outcomes per arm
  c("", round(delta.C, 0)),   # incremental costs
  c("", round(delta.E, 3)),   # incremental QALYs
  c("", round(ICER, 0))       # ICER
)
rownames(table_markov) = v.Trt   # name the rows
```

```
colnames(table_markov) = c("Costs", "QALYs","Incremental Costs", "QALYs Gaine
d", "ICER") # name the columns

kable(table_markov)              # print the table
```

	Costs	QALYs	Incremental Costs	QALYs Gained	ICER
No Treatment	75976	15.839			
Treatment	141623	16.400	65647	0.562	116901