3-state Markov model in R

with age 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. Division of Public Administration, Center for Research and Teaching in Economics (CIDE), 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
- Alarid-Escudero F, Krijkamp EM, Enns EA, Yang A, Hunink MGM Pechlivanoglou P, Jalal H. Cohort State-Transition Models in R: A Tutorial. arXiv:200107824v2. 2020:1-48. http://arxiv.org/abs/2001.07824
- 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 Mak. 2020;40(2):242-248. https://doi.org/10.1177/0272989X19893973

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("diagram", "dampack")
# install_github("DARTH-git/darthtools", force = TRUE) Uncomment if there is a newer version
p_load_gh("DARTH-git/darthtools")
```

02 Load functions

```
# No functions needed
```

03 Input model parameters

```
## General setup
n_cycles        <- 60                             # number of cycles
v_names_cycles  <- paste("cycle", 0:n_cycles)     # cycle names
v_names_states  <- c("Healthy", "Sick", "Dead")   # state names
n_states        <- length(v_names_states)         # number of health states

# Discounting factors
d_c             <- 0.03                           # discount rate for costs
d_e             <- 0.03                           # discount rate for QALYs

# Strategy names
v_names_str     <- c("Standard of Care",          # store the strategy names
                     "Treatment A",
                     "Treatment B")
n_str           <- length(v_names_str)            # number of strategies

## Transition probabilities
p_HS_SoC  <- 0.05  # probability of becoming sick when healthy, conditional on surviving, under standar
p_HS_trtA <- 0.04  # probability of becoming sick when healthy, conditional on surviving, under treatme
p_HS_trtB <- 0.02  # probability of becoming sick when healthy, conditional on surviving, under treatme
p_SD      <- 0.1   # probability of dying
p_HD_min  <- 0.003 # probability of dying when healthy at t = 0
p_HD_max  <- 0.01  # probability of dying when healthy at t = n_cycles
# probabilities of dying when healthy (age-dependent) - this is now a sequence of numbers
v_p_HD    <- seq(p_HD_min, p_HD_max, length.out = n_cycles)

## State rewards
```

```r
# 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
c_trtA    <- 800   # cost of treatment A (per cycle) in healthy state
c_trtB    <- 1500  # cost of treatment B (per cycle) in healthy state
u_H       <- 1     # utility when healthy
u_S       <- 0.5   # utility when sick
u_D       <- 0     # utility when dead
d_e       <- 0.03  # discount rate per cycle equal discount of costs and QALYs by 3%
d_c       <- 0.03  # discount rate per cycle equal discount of costs and QAL

# Discount weight (equal discounting is assumed for costs and effects)
v_dwc <- 1 / (1 + d_c) ^ (0:n_cycles)
v_dwe <- 1 / (1 + d_e) ^ (0:n_cycles)
```

04 Define and initialize matrices and vectors

04.1 Cohort trace

```r
## Initial state vector
# All starting healthy
v_s_init <- c("Healthy" = 1, "Sick" = 0, "Dead" = 0)
v_s_init
```

```
## Healthy    Sick    Dead
##       1       0       0
```

```r
## Initialize cohort trace for cSTM for all strategies
m_M_SoC <- matrix(0,
                  nrow = (n_cycles + 1), ncol = n_states,
                  dimnames = list(v_names_cycles, v_names_states))
# Store the initial state vector in the first row of the cohort trace
m_M_SoC[1, ] <- v_s_init
## Initialize cohort traces
m_M_trtA <- m_M_trtB <- m_M_SoC # structure and initial states remain the same
```

04.2 Transition probability array

```r
## Initialize transition probability array
# all transitions to a non-death state are assumed to be conditional on survival
a_P_SoC <- array(0,  # Create 3-D array
                 dim = c(n_states, n_states, n_cycles),
                 dimnames = list(v_names_states, v_names_states,
                                 v_names_cycles[-length(v_names_cycles)])) # name the dimensions of the
```

Fill in the transition probability array:

```
## Standard of Care
# from Healthy
a_P_SoC["Healthy", "Healthy", ] <- (1 - v_p_HD) * (1 - p_HS_SoC)
a_P_SoC["Healthy", "Sick",    ] <- (1 - v_p_HD) * p_HS_SoC
a_P_SoC["Healthy", "Dead",    ] <-  v_p_HD

# from Sick
a_P_SoC["Sick", "Sick", ] <- 1 - p_SD
a_P_SoC["Sick", "Dead", ] <- p_SD

# from Dead
a_P_SoC["Dead", "Dead", ] <- 1

## Treatment A
a_P_trtA <- a_P_SoC
a_P_trtA["Healthy", "Healthy", ] <- (1 - v_p_HD) * (1 - p_HS_trtA)
a_P_trtA["Healthy", "Sick",    ] <- (1 - v_p_HD) * p_HS_trtA

## Treatment B
a_P_trtB <- a_P_SoC
a_P_trtB["Healthy", "Healthy", ] <- (1 - v_p_HD) * (1 - p_HS_trtB)
a_P_trtB["Healthy", "Sick",    ] <- (1 - v_p_HD) * p_HS_trtB
```

Check if transition array and probabilities are valid.

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

05 Run Markov model

```
for (t in 1:n_cycles){  # loop through the number of cycles
  # estimate the cohort trace for cycle t + 1 using the t-th matrix from the probability array
  m_M_SoC [t + 1, ] <- m_M_SoC [t, ] %*% a_P_SoC [, , t]
  m_M_trtA[t + 1, ] <- m_M_trtA[t, ] %*% a_P_trtA[, , t]
  m_M_trtB[t + 1, ] <- m_M_trtB[t, ] %*% a_P_trtB[, , t]
}
head(m_M_SoC)  # print the first few lines of the matrix
```

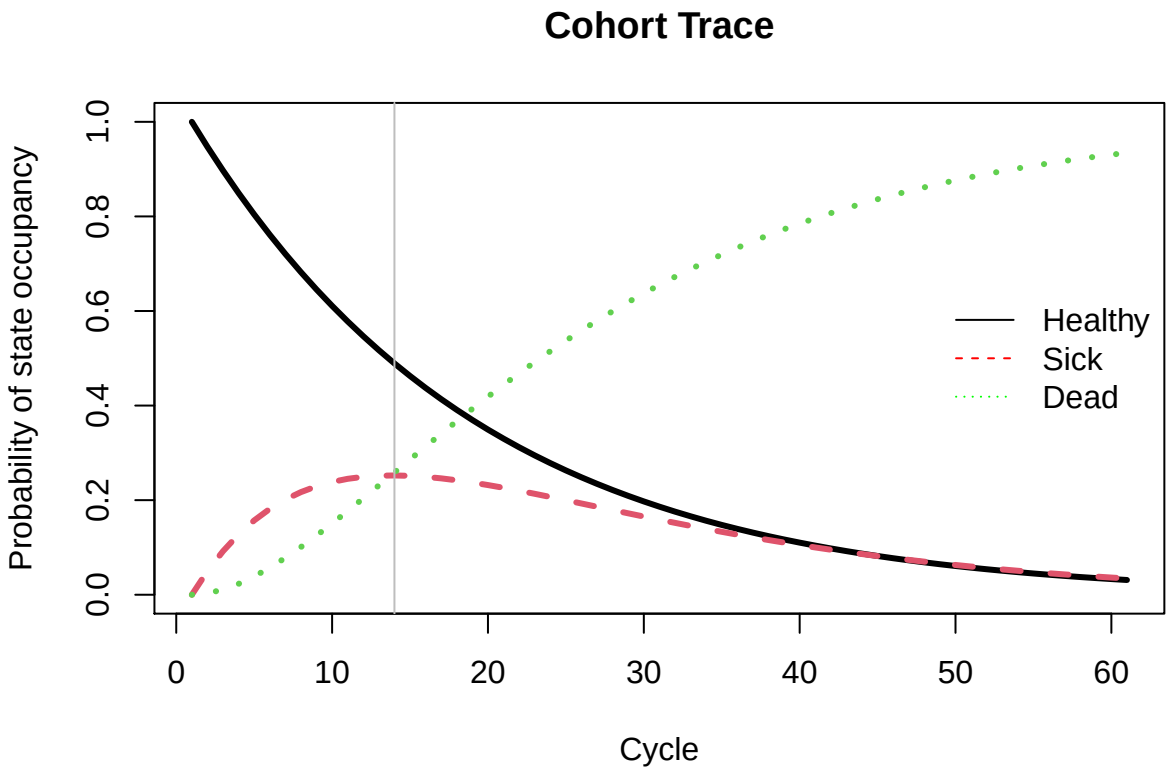
```
##           Healthy       Sick       Dead
## cycle 0 1.0000000 0.00000000 0.00000000
## cycle 1 0.9471500 0.04985000 0.00300000
## cycle 2 0.8969864 0.09207481 0.01093882
## cycle 3 0.8493784 0.12757146 0.02305011
## cycle 4 0.8042016 0.15714071 0.03865771
## cycle 5 0.7613370 0.18149700 0.05716604
```

06 Compute and Plot Epidemiological Outcomes

06.1 Cohort trace

```
matplot(m_M_SoC, type = 'l',
        ylab = "Probability of state occupancy",
        xlab = "Cycle",
        main = "Cohort Trace", lwd = 3)  # create a plot of the data
legend("right", v_names_states, col = c("black", "red", "green"),
       lty = 1:3, bty = "n")  # add a legend to the graph

# plot a vertical line that helps identifying at which cycle the prevalence of sick is highest
abline(v = which.max(m_M_SoC[, "Sick"]), col = "gray")
```

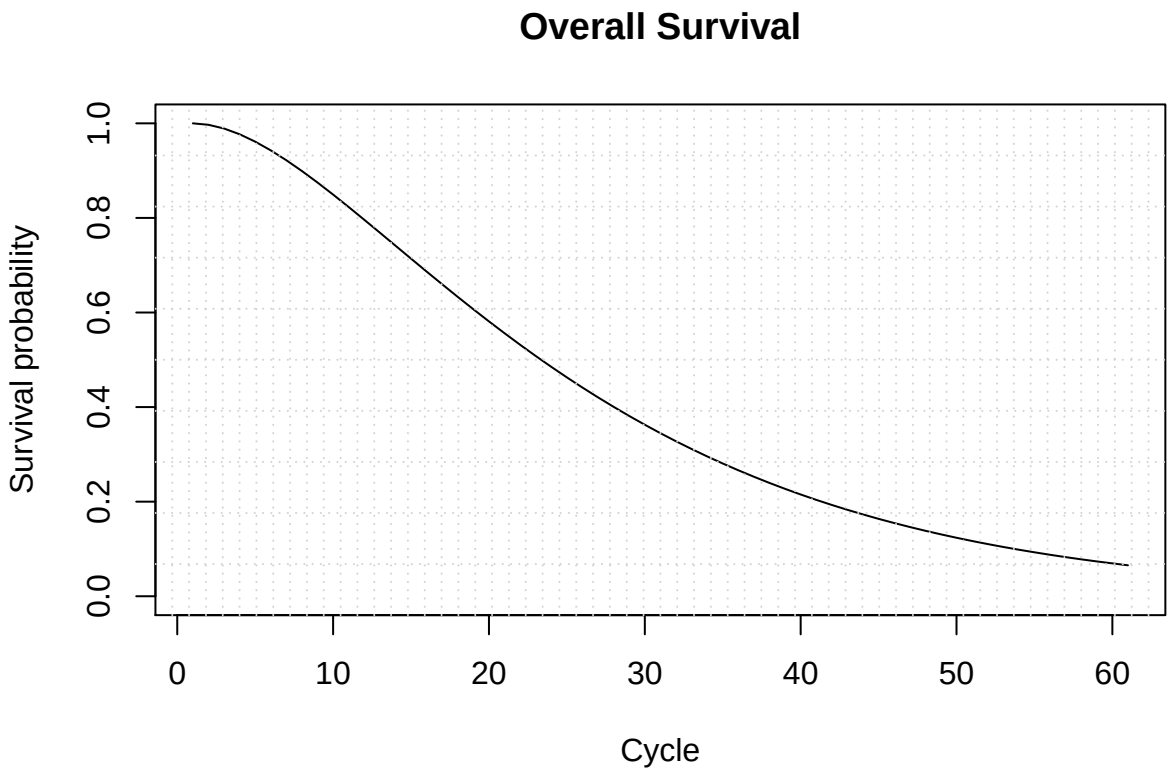


06.2 Overall Survival (OS)

```
v_os <- 1 - m_M_SoC[, "Dead"]     # calculate the overall survival (OS) probability
v_os <- rowSums(m_M_SoC[, 1:2])  # alternative way of calculating the OS probability

plot(v_os, type = 'l',
     ylim = c(0, 1),
     ylab = "Survival probability",
     xlab = "Cycle",
     main = "Overall Survival")  # create a simple plot showing the OS
```

```r
# add grid
grid(nx = n_cycles, ny = 10, col = "lightgray", lty = "dotted", lwd = par("lwd"),
     equilogs = TRUE)
```

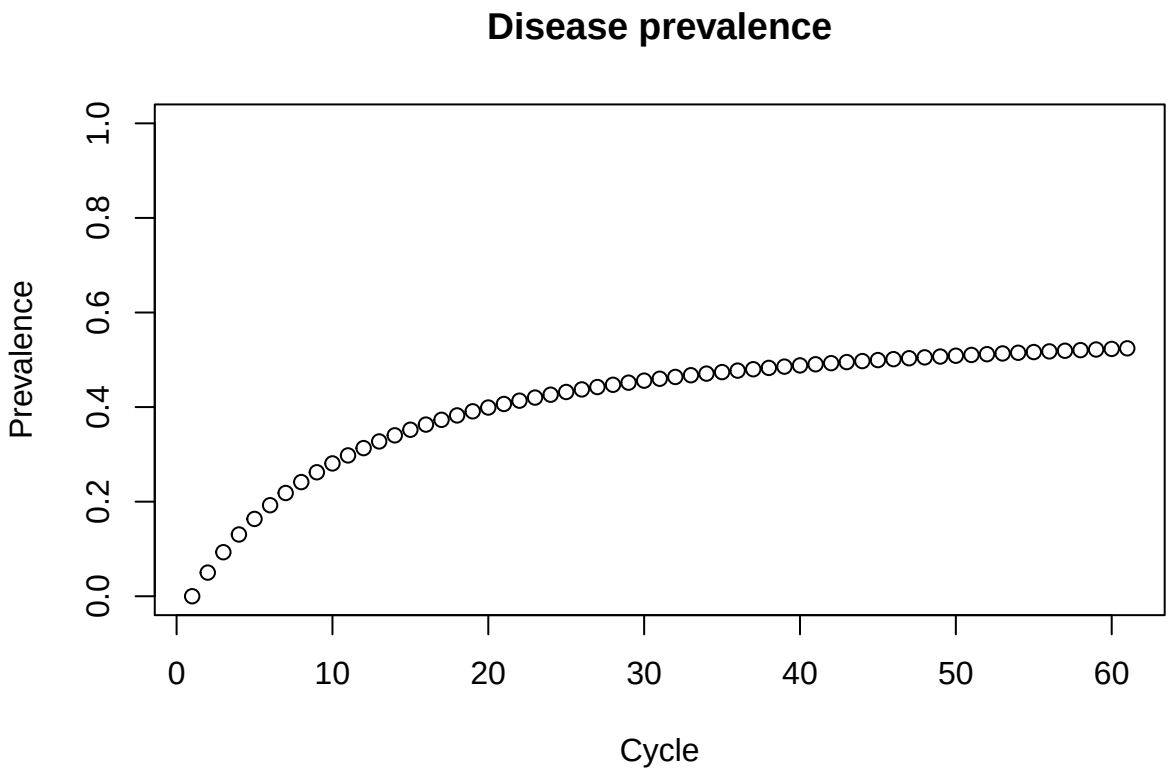


06.2.1 Life Expectancy (LE)

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

06.3 Disease prevalence

```r
v_prev <- m_M_SoC[, "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 cohort trace with the cost vector for the different health st
v_tc_SoC  <- m_M_SoC  %*% c(c_H, c_S, c_D)
v_tc_trtA <- m_M_trtA %*% c(c_H + c_trtA, c_S, c_D)
v_tc_trtB <- m_M_trtB %*% c(c_H + c_trtB, c_S, c_D)

# calculate expected QALYs by multiplying cohort trace with the utilities for the different health stat
v_tu_SoC  <- m_M_SoC  %*% c(u_H, u_S, u_D)
v_tu_trtA <- m_M_trtA %*% c(u_H, u_S, u_D)
v_tu_trtB <- m_M_trtB %*% 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)
tc_d_SoC  <-  t(v_tc_SoC)  %*% v_dwc
tc_d_trtA <-  t(v_tc_trtA) %*% v_dwc
tc_d_trtB <-  t(v_tc_trtB) %*% v_dwc

# Discount QALYS by multiplying the QALYs vector with discount weights (v_dw)
tu_d_SoC  <-  t(v_tu_SoC)  %*% v_dwe
```

```r
tu_d_trtA <-  t(v_tu_trtA) %*% v_dwe
tu_d_trtB <-  t(v_tu_trtB) %*% v_dwe

# Store them into a vector
v_tc_d <- c(tc_d_SoC, tc_d_trtA, tc_d_trtB)
v_tu_d <- c(tu_d_SoC, tu_d_trtA, tu_d_trtB)

# Dataframe with discounted costs and effectiveness
df_ce <- data.frame(Strategy = v_names_str,
                    Cost     = v_tc_d,
                    Effect   = v_tu_d)
df_ce
```

```
##           Strategy      Cost   Effect
## 1 Standard of Care  9504.301 14.50646
## 2      Treatment A 20648.861 15.82191
## 3      Treatment B 37325.926 19.57368
```

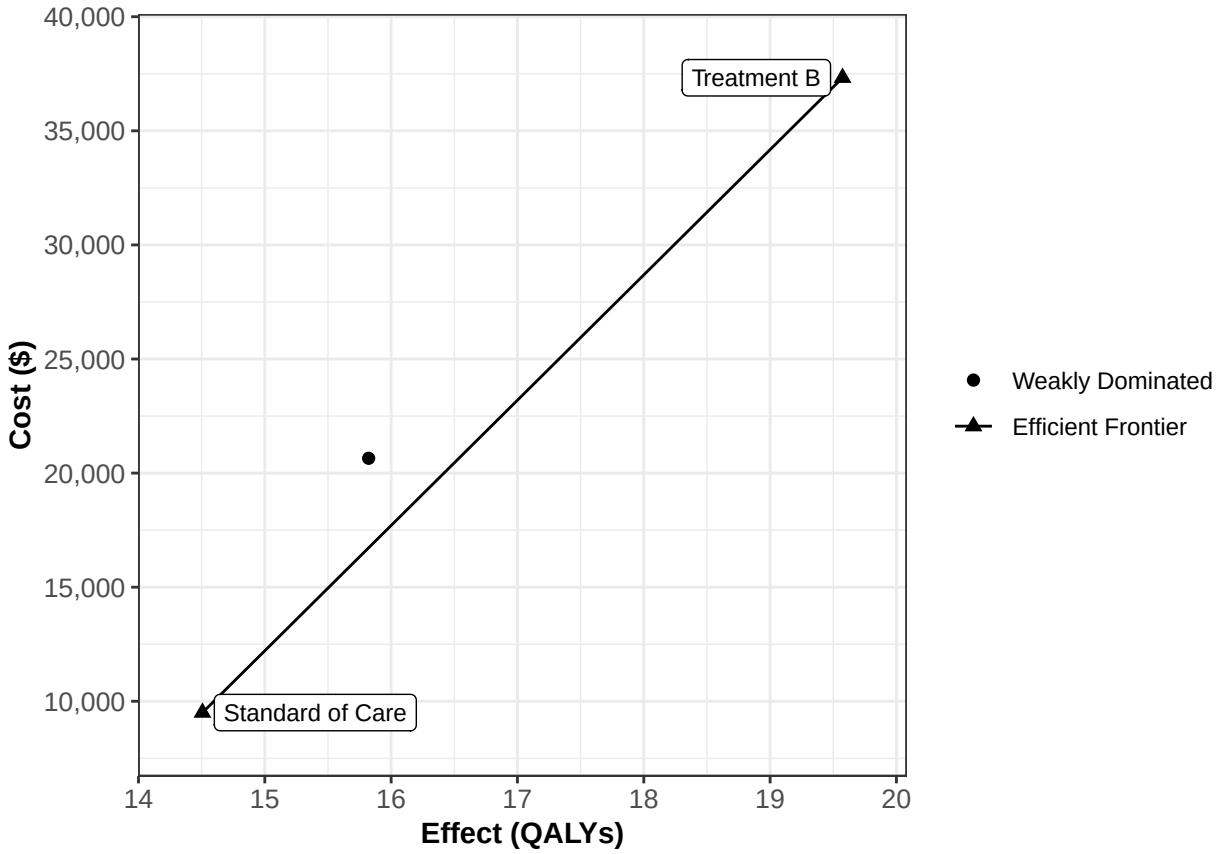
07.3 Compute ICERs of the Markov model

```r
df_cea <- calculate_icers(cost       = df_ce$Cost,
                          effect     = df_ce$Effect,
                          strategies = df_ce$Strategy
                          )
df_cea
```

```
##           Strategy      Cost   Effect Inc_Cost Inc_Effect     ICER Status
## 1 Standard of Care  9504.301 14.50646       NA         NA       NA     ND
## 2      Treatment B 37325.926 19.57368 27821.63   5.067224 5490.507     ND
## 3      Treatment A 20648.861 15.82191       NA         NA       NA     ED
```

07.4 Plot frontier of the Markov model

```r
plot(df_cea, effect_units = "QALYs")
```


40,000
35,000
30,000
25,000
20,000
15,000
10,000
Cost ($)
14
15
16
17
18
19
20
Effect (QALYs)
Treatment B
Standard of Care
Weakly Dominated
Efficient Frontier