###############################################################################

### Cohort state-transition models in R: ##

### From conceptualization to implementation 2019 ##

###############################################################################

# This code forms the basis for the state-transition model of the article:

# 'Cohort state-transition models in R: From conceptualization to implementation'

# Authors:

# - Fernando Alarid-Escudero <fernando.alarid@cide.edu>

# - Eline Krijkamp

# - Eva A. Enns

# - Alan Yang

# - Myriam G.M. Hunink

# - Petros Pechlivanoglou

# - Hawre Jalal

# Please cite the article when using this code

#

# To program this tutorial we made use of

# R version 3.5.0 (2018-04-23)

# Platform: x86\_64-apple-darwin15.6.0 (64-bit)

# Running under: macOS 10.14.5

# RStudio: Version 1.1.453 2009-2018 RStudio, Inc

###############################################################################

################# Code of Appendix ###########################################

###############################################################################

# Implements an age-dependent Sick-Sicker cSTM model #

##################################### Initial setup ###########################

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

library(dplyr) # to manipulate data

library(reshape2) # to transform data

library(ggplot2) # for nice looking plots

library(scales) # for dollar signs and commas

# devtools::install\_github("DARTH-git/dampack") # to install dampack form GitHub

library(dampack) # for CEA and calculate ICERs

# Define required functions

calc\_logit <- function(p) {

logit <- log(p/(1-p))

return(logit)

}

calc\_invlogit <- function(x) {

invlogit <- exp(x)/(1+exp(x))

return(invlogit)

}

################################## DARTH colors ###############################

# code for the DARTH colors for the figures

DARTHgreen <- '#009999'

DARTHyellow <- '#FDAD1E'

DARTHblue <- '#006699'

DARTHlightgreen <- '#00adad'

DARTHgray <- '#666666'

##################################### Model input ##############################

## General setup

n\_age\_init <- 25 # age at baseline

n\_age\_max <- 110 # maximum age of follow up

n\_t <- n\_age\_max - n\_age\_init # time horizon, number of cycles

v\_n <- c("H", "S1", "S2", "D") # the 4 health states of the model:

# Healthy (H), Sick (S1), Sicker (S2), Dead (D)

v\_hcc <- rep(1, n\_t+1) # vector of half-cycle correction

v\_hcc[1] <- v\_hcc[n\_t+1] <- 0.5 # half-cycle correction weight

n\_states <- length(v\_n) # number of health states

d\_c <- 0.03 # discount rate for costs

d\_e <- 0.03 # discount rate for QALYs

v\_names\_str <- c("Usual care", "New treatment 1", "New treatment 2", "New treatments 1 & 2") # store the strategy names

## Transition probabilities (per cycle) and hazard ratios

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

hr\_S1 <- 3 # hazard ratio of death in Sick vs Healthy

hr\_S2 <- 10 # hazard ratio of death in Sicker vs Healthy

# For New treatment 2

or\_S1S2 <- 0.7 # odds ratio of becoming Sicker when Sick under New treatment 2

lor\_S1S2 <- log(or\_S1S2) # log-odd ratio of becoming Sicker when Sick

logitp\_S1S2 <- calc\_logit(p\_S1S2) # log-odds of becoming Sicker when Sick

p\_S1S2\_trt2 <- calc\_invlogit(logitp\_S1S2 + lor\_S1S2) # probability to become Sicker when Sick under New treatment 2

## Age-dependent mortality rates

lt\_usa\_2005 <- read.csv("data/LifeTable\_USA\_Mx\_2015.csv")

v\_r\_mort\_by\_age <- lt\_usa\_2005 %>%

# filter(Age >= age & Age <= n\_age\_max) %>%

select(Total) %>%

as.matrix()

## State rewards

# Costs

c\_H <- 2000 # cost of remaining one cycle Healthy

c\_S1 <- 4000 # cost of remaining one cycle Sick

c\_S2 <- 15000 # cost of remaining one cycle Sicker

c\_D <- 0 # cost of being dead (per cycle)

c\_trt1 <- 12000 # cost of New treatment 1 (per cycle)

c\_trt2 <- 13000 # cost of New treatment 2 (per cycle)

# Utilities

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 Healthy

u\_trt1 <- 0.95 # utility when being treated

## Transition rewards

du\_HS1 <- 0.01 # disutility when transitioning from Healthy to Sick

ic\_HS1 <- 1000 # increase in cost when transitioning from Healthy to Sick

ic\_D <- 2000 # increase in cost when dying

# Discount weight (equal discounting is assumed for costs and effects)

v\_dwc <- 1 / ((1 + d\_e) ^ (0:(n\_t)))

v\_dwe <- 1 / ((1 + d\_c) ^ (0:(n\_t)))

## Age-specific transition probabilities

# Age-specific probability of dying when Healthy (all-cause mortality)

#THIS COULD BE BROKEN IN TWO STEPS : FIRST SELECT APPROPRIATE YEARS AND THEN PROB TO RATE

p\_HDage <- 1 - exp(-v\_r\_mort\_by\_age[(n\_age\_init + 1) + 0:(n\_t - 1)])

# Age-specific mortality risk in the Sick state

p\_S1Dage <- 1 - exp(-v\_r\_mort\_by\_age[(n\_age\_init + 1) + 0:(n\_t - 1)] \* hr\_S1)

# Age-specific mortality risk in the Sicker state

p\_S2Dage <- 1 - exp(-v\_r\_mort\_by\_age[(n\_age\_init + 1) + 0:(n\_t - 1)] \* hr\_S2)

######################## Construct state-transition models ####################

#### Create transition arrays ####

# Initialize 3-D array

a\_P <- array(0, dim = c(n\_states, n\_states, n\_t),

dimnames = list(v\_n, v\_n, 0:(n\_t - 1)))

### Fill in array

## From H

a\_P["H", "H", ] <- 1 - (p\_HS1 + p\_HDage)

a\_P["H", "S1", ] <- p\_HS1

a\_P["H", "D", ] <- p\_HDage

## From S1

a\_P["S1", "H", ] <- p\_S1H

a\_P["S1", "S1", ] <- 1 - (p\_S1H + p\_S1S2 + p\_S1Dage)

a\_P["S1", "S2", ] <- p\_S1S2

a\_P["S1", "D", ] <- p\_S1Dage

## From S2

a\_P["S2", "S2", ] <- 1 - p\_S2Dage

a\_P["S2", "D", ] <- p\_S2Dage

## From D

a\_P["D", "D", ] <- 1

# For New treatment 2

# Only need to update the probabilities involving p\_S1S2

a\_P\_trt2 <- a\_P

a\_P\_trt2["S1", "S1", ] <- 1 - (p\_S1H + p\_S1S2\_trt2 + p\_S1Dage)

a\_P\_trt2["S1", "S2", ] <- p\_S1S2\_trt2

# ### 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)

if (!isTRUE(all.equal(as.numeric(sum(valid)), as.numeric(n\_t)))) {

stop("This is not a valid transition Matrix")

}

valid2 <- apply(a\_P\_trt2, 3, function(x) sum(rowSums(x))==n\_states)

if (!isTRUE(all.equal(as.numeric(sum(valid)), as.numeric(n\_t)))) {

stop("This is not a valid transition Matrix")

}

#### Run Markov model ####

## Initial state vector

# All starting healthy

v\_s\_init <- c(H = 1, S1 = 0, S2 = 0, D = 0) # initial state vector

v\_s\_init

## Initialize cohort trace for age-dependent cSTM

m\_M\_ad <- matrix(0,

nrow = (n\_t + 1), ncol = n\_states,

dimnames = list(0:n\_t, v\_n))

# Store the initial state vector in the first row of the cohort trace

m\_M\_ad[1, ] <- v\_s\_init

# For New treatment 2

m\_M\_ad\_trt2 <- m\_M\_ad

## Initialize transition array

a\_A <- array(0,

dim = c(n\_states, n\_states, n\_t + 1),

dimnames = list(v\_n, v\_n, 0:n\_t))

# Set first slice of A with the initial state vector in its diagonal

diag(a\_A[, , 1]) <- v\_s\_init

# For New treatment 2

a\_A\_trt2 <- a\_A

## Iterative solution of age-dependent cSTM

for(t in 1:n\_t){

# Fill in cohort trace

m\_M\_ad[t + 1, ] <- m\_M\_ad[t, ] %\*% a\_P[, , t]

m\_M\_ad\_trt2[t + 1, ] <- m\_M\_ad\_trt2[t, ] %\*% a\_P\_trt2[, , t]

# Fill in transition dynamics array

a\_A[, , t + 1] <- m\_M\_ad[t, ] \* a\_P[, , t]

a\_A\_trt2[, , t + 1] <- m\_M\_ad\_trt2[t, ] \* a\_P\_trt2[, , t]

}

#### Plot Outputs ####

### Cohort trace

## Define colors and line types

cols <- c("H" = DARTHgreen, "S1" = DARTHblue,

"S2" = DARTHyellow, "D" = DARTHgray)

lty <- c("H" = 1, "S1" = 2, "S2" = 4, "D" = 3)

## Plot the cohort trace

ggplot(melt(m\_M\_ad), aes(x = Var1, y = value,

color = Var2, linetype = Var2)) +

geom\_line(size = 1) +

scale\_colour\_manual(name = "Health state",

values = cols) +

scale\_linetype\_manual(name = "Health state",

values = lty) +

xlab("Cycle") +

ylab("Proportion of the cohort") +

theme\_bw(base\_size = 14) +

theme(legend.position = "bottom",

legend.background = element\_rect(fill = NA))

# For New Treatment 2

ggplot(melt(m\_M\_ad\_trt2), aes(x = Var1, y = value,

color = Var2, linetype = Var2)) +

geom\_line(size = 1) +

scale\_colour\_manual(name = "Health state",

values = cols) +

scale\_linetype\_manual(name = "Health state",

values = lty) +

xlab("Cycle") +

ylab("Proportion of the cohort") +

theme\_bw(base\_size = 14) +

theme(legend.position = "bottom",

legend.background = element\_rect(fill = NA))

### Survival

v\_S\_ad <- rowSums(m\_M\_ad[, -4]) # vector with survival curve

## ggplot

ggplot(data.frame(Cycle = 0:n\_t, Survival = v\_S\_ad),

aes(x = Cycle, y = Survival)) +

geom\_line(size = 1.3) +

xlab("Cycle") +

ylab("Proportion alive") +

theme\_bw(base\_size = 14) +

theme()

# For New treatment 2

v\_S\_ad\_trt2 <- rowSums(m\_M\_ad\_trt2[, -4]) # vector with survival curve

## ggplot

ggplot(data.frame(Cycle = 0:n\_t, Survival = v\_S\_ad\_trt2),

aes(x = Cycle, y = Survival)) +

geom\_line(size = 1.3) +

xlab("Cycle") +

ylab("Proportion alive") +

theme\_bw(base\_size = 14) +

theme()

### Prevalence

###is the number of people at risk those that are alive or those that are sicker?

v\_prev\_S1 <- m\_M\_ad[, "S1"] / v\_S\_ad # vector with prevalence of Sick

v\_prev\_S2 <- m\_M\_ad[, "S2"] / v\_S\_ad # vector with prevalence of Sicker

v\_prev\_S1S2 <- rowSums(m\_M\_ad[, c("S1", "S2")]) / v\_S\_ad # vector with prevalence of Sick and Sicker

## Data.frame with all prevalence

df\_prev\_states <- data.frame(Cycle = 0:n\_t,

States = ordered(rep(c("S1", "S2", "S1 and S2"),

each = (n\_t + 1)),

levels = c("S1", "S2", "S1 and S2")),

Prevalence = c(v\_prev\_S1,

v\_prev\_S2,

v\_prev\_S1S2))

# I THINK WE NEED BETTER TITLES OVER EACH PLOT.

## ggplot

ggplot(df\_prev\_states,

aes(x = Cycle, y = Prevalence,

color = States, linetype = States)) +

geom\_line(size = 1) +

scale\_y\_continuous(labels = scales::percent) +

scale\_color\_discrete(name = "Health State", l = 50) +

scale\_linetype(name = "Health State") +

xlab("Cycle") +

ylab("Prevalence (%)") +

theme\_bw(base\_size = 14) +

theme(legend.position = "bottom")

# For New treatment 2

v\_prev\_S1\_trt2 <- m\_M\_ad\_trt2[, "S1"] / v\_S\_ad\_trt2 # vector with prevalence of Sick

v\_prev\_S2\_trt2 <- m\_M\_ad\_trt2[, "S2"] / v\_S\_ad\_trt2 # vector with prevalence of Sicker

v\_prev\_S1S2\_trt2 <- rowSums(m\_M\_ad\_trt2[, c("S1", "S2")]) / v\_S\_ad\_trt2 # vector with prevalence of Sick and Sicker

## Data.frame with all prevalence

df\_prev\_states\_trt2 <- data.frame(Cycle = 0:n\_t,

States = ordered(rep(c("S1", "S2", "S1 and S2"),

each = (n\_t + 1)),

levels = c("S1", "S2", "S1 and S2")),

Prevalence = c(v\_prev\_S1\_trt2,

v\_prev\_S2\_trt2,

v\_prev\_S1S2\_trt2))

## ggplot

ggplot(df\_prev\_states\_trt2,

aes(x = Cycle, y = Prevalence,

color = States, linetype = States)) +

geom\_line(size = 1) +

scale\_y\_continuous(labels = scales::percent) +

scale\_color\_discrete(name = "Health State", l = 50) +

scale\_linetype(name = "Health State") +

xlab("Cycle") +

ylab("Prevalence (%)") +

theme\_bw(base\_size = 14) +

theme(legend.position = "bottom")

### Proportion of Sicker among sick individuals

v\_prop\_S2 <- m\_M\_ad[-1, "S2"] / v\_prev\_S1S2[-1]

## ggplot

ggplot(data.frame(Cycle = 1:n\_t,

Proportion = v\_prop\_S2),

aes(x = Cycle, y = Proportion)) +

geom\_line(size = 1) +

# scale\_y\_continuous(labels = scales::percent) +

xlab("Cycle") +

ylab("Proportion") +

theme\_bw(base\_size = 14) +

theme()

# For New treatment2

v\_prop\_S2\_trt2 <- m\_M\_ad\_trt2[-1, "S2"] / v\_prev\_S1S2\_trt2[-1]

## ggplot

ggplot(data.frame(Cycle = 1:n\_t,

Proportion = v\_prop\_S2\_trt2),

aes(x = Cycle, y = Proportion)) +

geom\_line(size = 1) +

# scale\_y\_continuous(labels = scales::percent) +

xlab("Cycle") +

ylab("Proportion") +

theme\_bw(base\_size = 14) +

theme()

### Life expectancy

le\_ad <- sum(v\_S\_ad)

le\_ad\_trt2 <- sum(v\_S\_ad\_trt2)

#### State and Transition Rewards ####

### State rewards

## Vector of state utilities under Usual care

# LONG VECTORS ARE PREFERED THAN WIDE VECTORS AS THEY ARE MORE LEGIBLE

v\_u\_UC <- c( H = u\_H,

S1 = u\_S1,

S2 = u\_S2,

D = u\_D)

## Vector of state costs per cycle under Usual care

v\_c\_UC <- c(H = c\_H, S1 = c\_S1, S2 = c\_S2, D = c\_D)

## Vector of state utilities under New treatment 1

v\_u\_trt1 <- c(H = u\_H, S1 = u\_trt1, S2 = u\_S2, D = u\_D)

## Vector of state costs per cycle under New Treatment

v\_c\_trt1 <- c(H = c\_H, S1 = c\_S1 + c\_trt1, S2 = c\_S2 + c\_trt1, D = c\_D)

## Vector of state utilities under New treatment 2

v\_u\_trt2 <- c(H = u\_H, S1 = u\_S1, S2 = u\_S2, D = u\_D)

## Vector of state costs per cycle under New treatment 2

v\_c\_trt2 <- c(H = c\_H, S1 = c\_S1 + c\_trt2, S2 = c\_S2 + c\_trt2, D = c\_D)

## Vector of state utilities under New treatment 1 & 2

v\_u\_trt1\_2 <- c(H = u\_H, S1 = u\_trt1, S2 = u\_S2, D = u\_D)

## Vector of state costs per cycle under New treatment 1 & 2

v\_c\_trt1\_2 <- c(H = c\_H, S1 = c\_S1 + (c\_trt1 + c\_trt2), S2 = c\_S2 + (c\_trt1 + c\_trt2), D = c\_D)

### Arrays of rewards

## Array of state and transition utilities under Usual Care

a\_R\_u\_UC <- aperm(array(v\_u\_UC,

dim = c(n\_states, n\_states, n\_t + 1),

dimnames = list(v\_n, v\_n, 0:n\_t)),

perm = c(2, 1, 3))

## Array of state and transition costs per cycle under Usual Care

a\_R\_c\_UC <- aperm(array(v\_c\_UC,

dim = c(n\_states, n\_states, n\_t + 1),

dimnames = list(v\_n, v\_n, 0:n\_t)),

perm = c(2, 1, 3))

## Array of utilities under New treatment 1

a\_R\_u\_trt1 <- aperm(array(v\_u\_trt1,

dim = c(n\_states, n\_states, n\_t + 1),

dimnames = list(v\_n, v\_n, 0:n\_t)),

perm = c(2, 1, 3))

## Array of costs per cycle under New treatment 1

a\_R\_c\_trt1 <- aperm(array(v\_c\_trt1,

dim = c(n\_states, n\_states, n\_t + 1),

dimnames = list(v\_n, v\_n, 0:n\_t)),

perm = c(2, 1, 3))

## Array of utilities under New treatment 2

a\_R\_u\_trt2 <- aperm(array(v\_u\_trt2,

dim = c(n\_states, n\_states, n\_t + 1),

dimnames = list(v\_n, v\_n, 0:n\_t)),

perm = c(2, 1, 3))

## Array of costs under New treatment 2

a\_R\_c\_trt2 <- aperm(array(v\_c\_trt2,

dim = c(n\_states, n\_states, n\_t + 1),

dimnames = list(v\_n, v\_n, 0:n\_t)),

perm = c(2, 1, 3))

## Array of utilities under New treatment 1 & 2

a\_R\_u\_trt1\_2 <- aperm(array(v\_u\_trt1\_2,

dim = c(n\_states, n\_states, n\_t + 1),

dimnames = list(v\_n, v\_n, 0:n\_t)),

perm = c(2, 1, 3))

## Array of costs under New treatment 1 & 2

a\_R\_c\_trt1\_2 <- aperm(array(v\_c\_trt1\_2,

dim = c(n\_states, n\_states, n\_t + 1),

dimnames = list(v\_n, v\_n, 0:n\_t)),

perm = c(2, 1, 3))

### Transition rewards

## For Usual Care

# Add disutility due to transition from H to S1

a\_R\_u\_UC["H", "S1", ] <- a\_R\_u\_UC["H", "S1", ] - du\_HS1

# Add transition cost per cycle due to transition from H to S1

a\_R\_c\_UC["H", "S1", ] <- a\_R\_c\_UC["H", "S1", ] + ic\_HS1

# Add transition cost per cycle of dying from all non-dead states

a\_R\_c\_UC[-n\_states, "D", ] <- a\_R\_c\_UC[-n\_states, "D", ] + ic\_D

## For New treatment 1

# Add disutility due to transition from Healthy to Sick

a\_R\_u\_trt1["H", "S1", ] <- a\_R\_u\_trt1["H", "S1", ] - du\_HS1

# Add transition cost per cycle due to transition from Healthy to Sick

a\_R\_c\_trt1["H", "S1", ] <- a\_R\_c\_trt1["H", "S1", ] + ic\_HS1

# Add transition cost per cycle of dying from all non-dead states

a\_R\_c\_trt1[-n\_states, "D", ] <- a\_R\_c\_trt1[-n\_states, "D", ] + ic\_D

# For New treatment 2

# Add disutility due to transition from Healthy to Sick

a\_R\_u\_trt2["H", "S1", ] <- a\_R\_u\_trt2["H", "S1", ] - du\_HS1

# Add transition cost per cycle due to transition from Healthy to Sick

a\_R\_c\_trt2["H", "S1", ] <- a\_R\_c\_trt2["H", "S1", ] + ic\_HS1

# Add transition cost per cycle of dying from all non-dead states

a\_R\_c\_trt2[-n\_states, "D", ] <- a\_R\_c\_trt2[-n\_states, "D", ] + ic\_D

# For New treatment 1 & 2

# Add disutility due to transition from Healthy to Sick

a\_R\_u\_trt1\_2["H", "S1", ] <- a\_R\_u\_trt1\_2["H", "S1", ] - du\_HS1

# Add transition cost per cycle due to transition from Healthy to Sick

a\_R\_c\_trt1\_2["H", "S1", ] <- a\_R\_c\_trt1\_2["H", "S1", ] + ic\_HS1

# Add transition cost per cycle of dying from all non-dead states

a\_R\_c\_trt1\_2[-n\_states, "D", ] <- a\_R\_c\_trt1\_2[-n\_states, "D", ] + ic\_D

#### Expected QALYs and Costs for all transitions per cycle ####

# QALYs = life years x QoL

# NOTE: all parameters are annual

### For Usual Care

a\_Y\_c\_UC <- a\_A \* a\_R\_c\_UC

a\_Y\_u\_UC <- a\_A \* a\_R\_u\_UC

### For New treatment 1

a\_Y\_c\_trt1 <- a\_A \* a\_R\_c\_trt1

a\_Y\_u\_trt1 <- a\_A \* a\_R\_u\_trt1

### For New treatment 2

a\_Y\_c\_trt2 <- a\_A\_trt2 \* a\_R\_c\_trt2

a\_Y\_u\_trt2 <- a\_A\_trt2 \* a\_R\_u\_trt2

### For New treatment 1 & 2

a\_Y\_c\_trt1\_2 <- a\_A\_trt2 \* a\_R\_c\_trt1\_2

a\_Y\_u\_trt1\_2 <- a\_A\_trt2 \* a\_R\_u\_trt1\_2

#### Expected QALYs and Costs per cycle ####

## Vector of qalys under Usual Care

v\_qaly\_UC <- rowSums(t(colSums(a\_Y\_u\_UC)))

## Vector of costs under Usual Care

v\_cost\_UC <- rowSums(t(colSums(a\_Y\_c\_UC)))

## Vector of qalys under New Treatment 1

v\_qaly\_trt1 <- rowSums(t(colSums(a\_Y\_u\_trt1)))

## Vector of costs under New Treatment 1

v\_cost\_trt1 <- rowSums(t(colSums(a\_Y\_c\_trt1)))

## Vector of qalys under New Treatment 2

v\_qaly\_trt2 <- rowSums(t(colSums(a\_Y\_u\_trt2)))

## Vector of costs under New Treatment 2

v\_cost\_trt2 <- rowSums(t(colSums(a\_Y\_c\_trt2)))

## Vector of qalys under New Treatment 1 & 2

v\_qaly\_trt1\_2 <- rowSums(t(colSums(a\_Y\_u\_trt1\_2)))

## Vector of costs under New Treatment 1 & 2

#THERE MUST BE A MORE ELEGANT WAY!

v\_cost\_trt1\_2 <- rowSums(t(colSums(a\_Y\_c\_trt1\_2)))

#### Discounted total expected QALYs and Costs per strategy ####

### For Usual Care

## QALYs

n\_totqaly\_UC <- t(v\_qaly\_UC) %\*% (v\_dwe \* v\_hcc)

## Costs

n\_totcost\_UC <- t(v\_cost\_UC) %\*% (v\_dwc \* v\_hcc)

### For New treatment 1

## QALYs

n\_totqaly\_trt1 <- t(v\_qaly\_trt1) %\*% (v\_dwe \* v\_hcc)

## Costs

n\_totcost\_trt1 <- t(v\_cost\_trt1) %\*% (v\_dwc \* v\_hcc)

### For New treatment 2

## QALYs

n\_totqaly\_trt2 <- t(v\_qaly\_trt2) %\*% (v\_dwe \* v\_hcc)

## Costs

n\_totcost\_trt2 <- t(v\_cost\_trt2) %\*% (v\_dwc \* v\_hcc)

### For New treatment 1 & 2

## QALYs

n\_totqaly\_trt1\_2 <- t(v\_qaly\_trt1\_2) %\*% (v\_dwe \* v\_hcc)

## Costs

n\_totcost\_trt1\_2 <- t(v\_cost\_trt1\_2) %\*% (v\_dwc \* v\_hcc)

########################### Cost-effectiveness analysis #######################

### Vector of total costs for all strategies

v\_ted\_cost <- c(n\_totcost\_UC, n\_totcost\_trt1, n\_totcost\_trt2, n\_totcost\_trt1\_2)

### Vector of effectiveness for all strategies

v\_ted\_qaly <- c(n\_totqaly\_UC, n\_totqaly\_trt1, n\_totqaly\_trt2, n\_totqaly\_trt1\_2)

### Calculate incremental cost-effectiveness ratios (ICERs)

df\_cea <- calculate\_icers(cost = v\_ted\_cost,

effect = v\_ted\_qaly,

strategies = v\_names\_str)

df\_cea

### Create CEA table

table\_cea <- df\_cea

## Format column names

colnames(table\_cea)[2:6] <- c("Costs ($)", "QALYs",

"Incremental Costs ($)", "Incremental QALYs",

"ICER ($/QALY)") # name the columns

## Format rows

table\_cea$`Costs ($)` <- comma(round(table\_cea$`Costs ($)`, 0))

table\_cea$`Incremental Costs ($)` <- comma(round(table\_cea$`Incremental Costs ($)`, 0))

table\_cea$QALYs <- round(table\_cea$QALYs, 2)

table\_cea$`Incremental QALYs` <- round(table\_cea$`Incremental QALYs`, 2)

table\_cea$`ICER ($/QALY)` <- comma(round(table\_cea$`ICER ($/QALY)`, 0))

table\_cea

### CEA frontier

plot(df\_cea, label="all") +

expand\_limits(x = max(table\_cea$QALYs + 0.5)) # change this