Markov_model_realworld

Kyu Lee

2023-04-12

In this document, I provided line-by-line explanation of the code, 'Markov_model_realworld.
by Briggs et al

Overall structure of the code < Base-case analysis> 1. Define variables and parameters for the model 2. Prepare cost, qaly, and transitional cost matrix 3. Prepare transition probability matrix (this will change in simulation given age-specific mortality) 4. Prepare population matrix that keeps track of health state transition by cycle 5. Prepare a matrix ('trans') to keep track of number of people transition from one state to another state (this will be used to calculate transitional cost) 6. Prepare empty matrices to record QALYs and Cost outcomes by cycle and by strategy 7. Define a function to calculate time-dependent transition probability based on age-specific mortality 8. Simulate a cohort for 'no drug' and 'drug' scenario 9. Calculate cost and qaly of each scenario 10. Calculate ICER 11. Plot

1. Make the base-case analysis code as a function so that we can repeat the simulation by the sampled parameter set 2. Random sampling of parameters 3. Define cost and qaly matrices to record outcomes by parameter sample 4. Run PSA 5. Plot

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

1. Define variables and parameters t_names : strategy/scenario label n_treatment: number of strategies

s_names: vector of health states n_states: number of health states

n_cohort: cohort size cycle: cycle length n_cylces: total number of cycles Initial_age: age at the beginning of simulation effect: drug efficacy in decreasing the risk of progressing from asymptomatic to progressive disease

cAsymp: cost of having asymptomatic disease (per-cycle cost) cDeath: cost of death (per-cycle cost) cDrug: cost of drug (one-time cost) cProg: cost of having progressive disease (per-cycle cost) uAsymp: utility of having asymptomatic disease uProg: utility of having progressive disease oDr: discount rate for qaly cDr: discount rate for cost tpDcm: excess mortality with progressive disease tpProg: transition probability from asymptomatic to progressive disease tpDn: baseline mortality <- mortality 0.0379 # over 65 year old

```
t_names <- c("without_drug", "with_drug")
n_treatments <- length(t_names)
```

```r
s_names  <- c("Asymptomatic_disease", "Progressive_disease", "Dead")
n_states <- length(s_names)

n_cohort <- 1000
cycle <- 1

n_cycles <- 46
Initial_age <- 55
effect <- 0.5

cAsymp <- 500
cDeath <- 1000
cDrug <- 1000
cProg <- 3000
uAsymp <- 0.95
uProg <- 0.75
oDr <- 0.06
cDr <- 0.06
tpDcm <- 0.15
tpProg <- 0.01
tpDn <- 0.0379  # over 65 year old
effect <- 0.5
```

2. Prepare cost, qaly, and transitional cost matrix cost and qaly matrix has the following structure row: strategies column: health states value: cost, qaly, or transitional cost Transitional cost matrix has the following structure row: health states (departing state) column: health states(arriving state) value: transitional cost(toll)

```r
# cost of staying in state
state_c_matrix <-
  matrix(c(cAsymp, cProg, 0,
           cAsymp + cDrug, cProg, 0),
         byrow = TRUE,
         nrow = n_treatments,
         dimnames = list(t_names,
                         s_names))
state_c_matrix
```

```
##              Asymptomatic_disease Progressive_disease Dead
## without_drug                  500                3000    0
## with_drug                    1500                3000    0
```

```r
# qaly when staying in state
state_q_matrix <-
  matrix(c(uAsymp, uProg, 0,
           uAsymp, uProg, 0),
         byrow = TRUE,
         nrow = n_treatments,
         dimnames = list(t_names,
                         s_names))

# cost of moving to a state
```

```r
# same for both treatments
trans_c_matrix <-
  matrix(c(0, 0, 0,
           0, 0, cDeath,
           0, 0, 0),
         byrow = TRUE,
         nrow = n_states,
         dimnames = list(from = s_names,
                         to = s_names))
```

3. Prepare transition probability matrix p_matrix is time-homogeneous transition probability, meaning the elements of matrix would not change depending on cycle # Later in the code, this will be replaced by a time-dependent transition probability matrix when incorporating age-specific mortality

Note: array is useful to create a multidimensional matrix. Here p_matrix has 3-dimensional matrix with (health states x health states) x strategies

```r
# Transition probabilities ----

# time-homogeneous
p_matrix <- array(data = c(1 - tpProg - tpDn, 0, 0,
                           tpProg, 1 - tpDcm - tpDn, 0,
                           tpDn, tpDcm + tpDn, 1,
                           1 - tpProg*(1-effect) - tpDn, 0, 0,
                           tpProg*(1-effect), 1 - tpDcm - tpDn, 0,
                           tpDn, tpDcm + tpDn, 1),
                  dim = c(n_states, n_states, n_treatments),
                  dimnames = list(from = s_names,
                                  to = s_names,
                                  t_names))
p_matrix
```

```
## , ,  = without_drug
##
##                       to
## from                   Asymptomatic_disease Progressive_disease   Dead
##   Asymptomatic_disease               0.9521              0.0100 0.0379
##   Progressive_disease                0.0000              0.8121 0.1879
##   Dead                               0.0000              0.0000 1.0000
##
## , ,  = with_drug
##
##                       to
## from                   Asymptomatic_disease Progressive_disease   Dead
##   Asymptomatic_disease               0.9571              0.0050 0.0379
##   Progressive_disease                0.0000              0.8121 0.1879
##   Dead                               0.0000              0.0000 1.0000
```

4. Prepare population matrix that keeps track of health state transition by cycle pop matrix will record health state distribution in the population for each cycle by srategy (multidimensional matrix: n_states x n_cycles x n_treatments)

In cycle = 1, everyone is in the asymptomatic disease state.

```
# Store population output for each cycle

# state populations
pop <- array(data = NA,
             dim = c(n_states, n_cycles, n_treatments),
             dimnames = list(state = s_names,
                             cycle = NULL,
                             treatment = t_names))

pop["Asymptomatic_disease", cycle = 1, ] <- n_cohort
pop["Progressive_disease", cycle = 1, ] <- 0
pop["Dead", cycle = 1, ] <- 0

head(pop)
```

```
## , , treatment = without_drug
##
##                         cycle
## state                    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
##   Asymptomatic_disease 1000   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA
##   Progressive_disease     0   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA
##   Dead                    0   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA
##                         cycle
## state                    [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]
##   Asymptomatic_disease    NA    NA    NA    NA    NA    NA    NA    NA    NA
##   Progressive_disease     NA    NA    NA    NA    NA    NA    NA    NA    NA
##   Dead                    NA    NA    NA    NA    NA    NA    NA    NA    NA
##                         cycle
## state                    [,21] [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29]
##   Asymptomatic_disease    NA    NA    NA    NA    NA    NA    NA    NA    NA
##   Progressive_disease     NA    NA    NA    NA    NA    NA    NA    NA    NA
##   Dead                    NA    NA    NA    NA    NA    NA    NA    NA    NA
##                         cycle
## state                    [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38]
##   Asymptomatic_disease    NA    NA    NA    NA    NA    NA    NA    NA    NA
##   Progressive_disease     NA    NA    NA    NA    NA    NA    NA    NA    NA
##   Dead                    NA    NA    NA    NA    NA    NA    NA    NA    NA
##                         cycle
## state                    [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46]
##   Asymptomatic_disease    NA    NA    NA    NA    NA    NA    NA    NA
##   Progressive_disease     NA    NA    NA    NA    NA    NA    NA    NA
##   Dead                    NA    NA    NA    NA    NA    NA    NA    NA
##
## , , treatment = with_drug
##
##                         cycle
## state                    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
##   Asymptomatic_disease 1000   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA
##   Progressive_disease     0   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA
##   Dead                    0   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA
##                         cycle
## state                    [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]
```

```
##   Asymptomatic_disease    NA    NA    NA    NA    NA    NA    NA    NA    NA
##   Progressive_disease     NA    NA    NA    NA    NA    NA    NA    NA    NA
##   Dead                    NA    NA    NA    NA    NA    NA    NA    NA    NA
##                        cycle
## state                   [,21] [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29]
##   Asymptomatic_disease    NA    NA    NA    NA    NA    NA    NA    NA    NA
##   Progressive_disease     NA    NA    NA    NA    NA    NA    NA    NA    NA
##   Dead                    NA    NA    NA    NA    NA    NA    NA    NA    NA
##                        cycle
## state                   [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38]
##   Asymptomatic_disease    NA    NA    NA    NA    NA    NA    NA    NA    NA
##   Progressive_disease     NA    NA    NA    NA    NA    NA    NA    NA    NA
##   Dead                    NA    NA    NA    NA    NA    NA    NA    NA    NA
##                        cycle
## state                   [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46]
##   Asymptomatic_disease    NA    NA    NA    NA    NA    NA    NA    NA
##   Progressive_disease     NA    NA    NA    NA    NA    NA    NA    NA
##   Dead                    NA    NA    NA    NA    NA    NA    NA    NA
```

5. Prepare a matrix ('trans') to record total transitional cost per cycle by state 'pop' matrix records total number of people in each health state, whereas trans records the number of people who 'enter' the state

dimension: n_states x n_cycles x n_treatments (here n_states indicate 'arrived state') So given treatment strategy and in a specific cycle, trans["asympomatic"] indicates the total "toll" or "transitional costs" to arrive the asymptomatic state by the end of the cycle

```r
# _arrived_ state populations
trans <- array(data = NA,
               dim = c(n_states, n_cycles, n_treatments),
               dimnames = list(state = s_names,
                               cycle = NULL,
                               treatment = t_names))

trans[, cycle = 1, ] <- 0
```

6. Prepare empty matrices to record QALYs and Cost outcomes by cycle and by strategy by cycle: cycle_costs, cycle_QALYs, cycle_QALE, LE, LYs (dimension: n_treatments, cycles) total: total_costs, total_QALYs (1xn_treatments)

```r
# Sum costs and QALYs for each cycle at a time for each drug

cycle_empty_array <-
  array(NA,
        dim = c(n_treatments, n_cycles),
        dimnames = list(treatment = t_names,
                        cycle = NULL))

cycle_state_costs <- cycle_trans_costs <- cycle_empty_array
cycle_costs <- cycle_QALYs <- cycle_empty_array
LE <- LYs <- cycle_empty_array    # life expectancy; life-years
cycle_QALE <- cycle_empty_array   # quality-adjusted life expectancy
```

```
total_costs <- setNames(c(NA, NA), t_names)
total_QALYs <- setNames(c(NA, NA), t_names)
```

7. Define a function to calculate time-dependent transition probability based on age-specific mortality Because non-mortality transition probabilities are dependent on mortality, if mortality is age-dependent (or time-dependent), then other probabilities will be time-dependent as well

```
# Time-dependent probability matrix ----

p_matrix_cycle <- function(p_matrix, age, cycle,
                           tpProg = 0.01,
                           tpDcm = 0.15,
                           effect = 0.5) {
  tpDn_lookup <-
    c("(34,44]" = 0.0017,
      "(44,54]" = 0.0044,
      "(54,64]" = 0.0138,
      "(64,74]" = 0.0379,
      "(74,84]" = 0.0912,
      "(84,100]" = 0.1958)

  age_grp <- cut(age, breaks = c(34,44,54,64,74,84,100)) # find the age group that this age falls into
  tpDn <- tpDn_lookup[age_grp]

  # Matrix containing transition probabilities for without_drug
  p_matrix["Asymptomatic_disease", "Progressive_disease", "without_drug"] <- tpProg*cycle
  p_matrix["Asymptomatic_disease", "Dead", "without_drug"] <- tpDn
  p_matrix["Asymptomatic_disease", "Asymptomatic_disease", "without_drug"] <- 1 - tpProg*cycle - tpDn
  p_matrix["Progressive_disease", "Dead", "without_drug"] <- tpDcm + tpDn
  p_matrix["Progressive_disease", "Progressive_disease", "without_drug"] <- 1 - tpDcm - tpDn
  p_matrix["Dead", "Dead", "without_drug"] <- 1

  # Matrix containing transition probabilities for with_drug
  p_matrix["Asymptomatic_disease", "Progressive_disease", "with_drug"] <- tpProg*(1 - effect)*cycle
  p_matrix["Asymptomatic_disease", "Dead", "with_drug"] <- tpDn
  p_matrix["Asymptomatic_disease", "Asymptomatic_disease", "with_drug"] <-
    1 - tpProg*(1 - effect)*cycle - tpDn
  p_matrix["Progressive_disease", "Dead", "with_drug"] <- tpDcm + tpDn
  p_matrix["Progressive_disease", "Progressive_disease", "with_drug"] <- 1 - tpDcm - tpDn
  p_matrix["Dead", "Dead", "with_drug"] <- 1

  return(p_matrix)
}
```

8. Simulate a cohort for 'no drug' and 'drug' scenario two for loops are implemented in this block

(1) given strategy
(2) given cycle #

- calculate age-specific transition probability matrix
- update pop matrix in the current cycle given pop matrix in last cycle and transition probability matrix
- calculate trans matrix to count the number of people move from one to another state within a cycle

A * B: element-wise multiplication A %*% B: matrix multiplication (take the row of A, column of B and sum the product)

To calculate the transitional cost per cycle to arrive in each state, For example, the transitional cost of arriving in "progressive disease" is N_1C_12P_12 + N_2C_22P_22+N+3C_32P_32 = c(N_1, N_2, N_3) %% $t(c(C_12$P_12, C_22P_22, C_32P_32)) = c(N_1, N_2, N_3) %% $t(c(C\_12,C\_22,C\_32)$ c(P_12,P_22,P_32))

trans_c_matrix * p_matrix -> transition probability-weighted transitional cost c(1,1,1) %*% column vector() -> sum of elements in the column vector

```
## Run model ----

for (i in 1:n_treatments) {

  age <- Initial_age

  for (j in 2:n_cycles) {

    p_matrix <- p_matrix_cycle(p_matrix, age, j - 1)

    pop[, cycle = j, treatment = i] <-
      pop[, cycle = j - 1, treatment = i] %*% p_matrix[, , treatment = i]

    trans[, cycle = j, treatment = i] <-
      pop[, cycle = j - 1, treatment = i] %*% (trans_c_matrix * p_matrix[, , treatment = i])

    age <- age + 1
  }

  cycle_state_costs[i, ] <-
    (state_c_matrix[treatment = i, ] %*% pop[, , treatment = i]) * 1/(1 + cDr)^(1:n_cycles - 1)

  # discounting at _previous_ cycle
  cycle_trans_costs[i, ] <-
    (c(1,1,1) %*% trans[, , treatment = i]) * 1/(1 + cDr)^(1:n_cycles - 2)

  cycle_costs[i, ] <- cycle_state_costs[i, ] + cycle_trans_costs[i, ]

  # life expectancy
  LE[i, ] <- c(1,1,0) %*% pop[, , treatment = i]

  # life-years
  LYs[i, ] <- LE[i, ] * 1/(1 + oDr)^(1:n_cycles - 1)

  # quality-adjusted life expectancy
  cycle_QALE[i, ] <-
    state_q_matrix[treatment = i, ] %*% pop[, , treatment = i]

  # quality-adjusted life-years
  cycle_QALYs[i, ] <- cycle_QALE[i, ] * 1/(1 + oDr)^(1:n_cycles - 1)

  total_costs[i] <- sum(cycle_costs[treatment = i, -1])
  total_QALYs[i] <- sum(cycle_QALYs[treatment = i, -1])
}
```

9. Calculate cost and qaly of each scenario

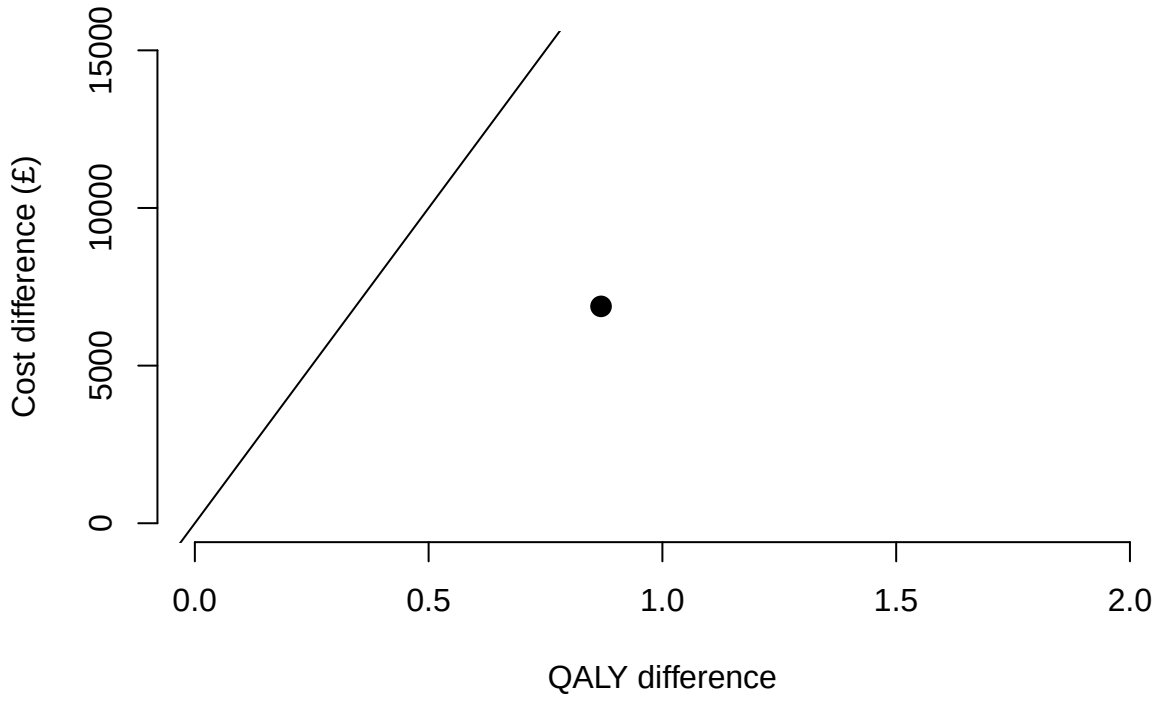
```
## Plot results ----

# Incremental costs and QALYs of with_drug vs to without_drug
c_incr <- total_costs["with_drug"] - total_costs["without_drug"]
q_incr <- total_QALYs["with_drug"] - total_QALYs["without_drug"]
```

10. Calculate ICER

```
# Incremental cost-effectiveness ratio
ICER <- c_incr/q_incr
```

11. Plot

```
wtp <- 20000
plot(x = q_incr/n_cohort, y = c_incr/n_cohort,
     xlim = c(0, 2),
     ylim = c(0, 15e3),
     pch = 16, cex = 1.5,
     xlab = "QALY difference",
     ylab = paste0("Cost difference (", enc2utf8("\u00A3"), ")"),
     frame.plot = FALSE)
abline(a = 0, b = wtp) # willingness-to-pay threshold
```



```r
png("figures/ceplane_point.png", width = 4, height = 4, units = "in", res = 640)
plot(x = q_incr/n_cohort, y = c_incr/n_cohort,
     xlim = c(0, 2),
     ylim = c(0, 15e3),
     pch = 16, cex = 1.5,
     xlab = "QALY difference",
     ylab = paste0("Cost difference (", enc2utf8("\u00A3"), ")"),
     frame.plot = FALSE)
abline(a = 0, b = wtp) # willingness-to-pay threshold
dev.off()
```

```
## pdf
##   2
```

PSA code 1. Turn the base-case analysis code into a function so that we can repeat the simulation by the sampled parameter set

```r
##########################################
# replace with sim_pop()

# simulate state populations
sim_pop <- function(n_cycles, age,
                    trans_c_matrix,
                    p_matrix, pop, trans, i) {

  for (j in 2:n_cycles) {
    p_matrix <- p_matrix_cycle(p_matrix, age, j - 1)
    pop[, cycle = j, i] <-
      pop[, cycle = j - 1, i] %*% p_matrix[, , i]
    trans[, cycle = j, i] <-
      pop[, cycle = j - 1, i] %*% (trans_c_matrix * p_matrix[, , i])
    age <- age + 1
  }

  list(pop = pop[, , i],
       trans = trans[, , i])
}
```

```r
for (i in 1:n_treatments) {

  # simulate state populations
  sim_res <-
    sim_pop(n_cycles, Initial_age,
            trans_c_matrix,
            p_matrix, pop, trans, i)

  trans[, , i] <- sim_res$trans
  pop[, , i] <- sim_res$pop

  cycle_state_costs[i, ] <-
    (state_c_matrix[treatment = i, ] %*% pop[, , treatment = i]) * 1/(1 + cDr)^(1:n_cycles - 1)

  # discounting at _previous_ cycle
```

```
  cycle_trans_costs[i, ] <-
    (c(1,1,1) %*% trans[, , treatment = i]) * 1/(1 + cDr)^(1:n_cycles - 2)

  cycle_costs[i, ] <- cycle_state_costs[i, ] + cycle_trans_costs[i, ]

  # life expectancy
  LE[i, ] <- c(1,1,0) %*% pop[, , treatment = i]

  # life-years
  LYs[i, ] <- LE[i, ] * 1/(1 + oDr)^(1:n_cycles - 1)

  # quality-adjusted life expectancy
  cycle_QALE[i, ] <-
    state_q_matrix[treatment = i, ] %*% pop[, , treatment = i]

  # quality-adjusted life-years
  cycle_QALYs[i, ] <- cycle_QALE[i, ] * 1/(1 + oDr)^(1:n_cycles - 1)

  total_costs[i] <- sum(cycle_costs[treatment = i, -1])
  total_QALYs[i] <- sum(cycle_QALYs[treatment = i, -1])


}
```

```
##############################################
# Probability Sensitivity Analysis (PSA)

ce_markov <- function(start_pop,
                      p_matrix,
                      state_c_matrix,
                      trans_c_matrix,
                      state_q_matrix,
                      n_cycles = 46,
                      init_age = 55,
                      s_names = NULL,
                      t_names = NULL) {

  n_states <- length(start_pop)
  n_treat <- dim(p_matrix)[3]

  pop <- array(data = NA,
               dim = c(n_states, n_cycles, n_treat),
               dimnames = list(state = s_names,
                               cycle = NULL,
                               treatment = t_names))
  trans <- array(data = NA,
                 dim = c(n_states, n_cycles, n_treat),
                 dimnames = list(state = s_names,
                                 cycle = NULL,
                                 treatment = t_names))

  for (i in 1:n_states) {
    pop[i, cycle = 1, ] <- start_pop[i]
```

```
}

cycle_empty_array <-
  array(NA,
        dim = c(n_treat, n_cycles),
        dimnames = list(treatment = t_names,
                        cycle = NULL))

cycle_state_costs <- cycle_trans_costs <- cycle_empty_array
cycle_costs <- cycle_QALYs <- cycle_empty_array
LE <- LYs <- cycle_empty_array    # life expectancy; life-years
cycle_QALE <- cycle_empty_array   # quality-adjusted life expectancy

total_costs <- setNames(rep(NA, n_treat), t_names)
total_QALYs <- setNames(rep(NA, n_treat), t_names)

for (i in 1:n_treat) {

  age <- init_age

  for (j in 2:n_cycles) {

    # difference from point estimate case
    # pass in functions for random sample
    # rather than fixed values
    p_matrix <- p_matrix_cycle(p_matrix, age, j - 1,
                               tpProg = tpProg(),
                               tpDcm = tpDcm(),
                               effect = effect())

    # Matrix multiplication
    pop[, cycle = j, treatment = i] <-
      pop[, cycle = j - 1, treatment = i] %*% p_matrix[, , treatment = i]

    trans[, cycle = j, treatment = i] <-
      pop[, cycle = j - 1, treatment = i] %*% (trans_c_matrix * p_matrix[, , treatment = i])

    age <- age + 1
  }

  cycle_state_costs[i, ] <-
    (state_c_matrix[treatment = i, ] %*% pop[, , treatment = i]) * 1/(1 + cDr)^(1:n_cycles - 1)

  cycle_trans_costs[i, ] <-
    (c(1,1,1) %*% trans[, , treatment = i]) * 1/(1 + cDr)^(1:n_cycles - 2)

  cycle_costs[i, ] <- cycle_state_costs[i, ] + cycle_trans_costs[i, ]

  LE[i, ] <- c(1,1,0) %*% pop[, , treatment = i]

  LYs[i, ] <- LE[i, ] * 1/(1 + oDr)^(1:n_cycles - 1)

  cycle_QALE[i, ] <-
```

```
     state_q_matrix[treatment = i, ] %*%  pop[, , treatment = i]

    cycle_QALYs[i, ] <- cycle_QALE[i, ] * 1/(1 + oDr)^(1:n_cycles - 1)

    total_costs[i] <- sum(cycle_costs[treatment = i, -1])
    total_QALYs[i] <- sum(cycle_QALYs[treatment = i, -1])
  }

  list(pop = pop,
       cycle_costs = cycle_costs,
       cycle_QALYs = cycle_QALYs,
       total_costs = total_costs,
       total_QALYs = total_QALYs)
}
```

2. Random sampling of parameters

```
# replace point values with functions to random sample

cAsymp <- function() rnorm(1, 500, 127.55)
cDeath <- function() rnorm(1, 1000, 255.11)
cDrug  <- function() rnorm(1, 1000, 102.04)
cProg  <- function() rnorm(1, 3000, 510.21)
effect <- function() rnorm(1, 0.5, 0.051)
tpDcm  <- function() rbeta(1, 29, 167)
tpProg <- function() rbeta(1, 15, 1506)
uAsymp <- function() rbeta(1, 69, 4)
uProg  <- function() rbeta(1, 24, 8)
```

3. Define cost and qaly matrices to record outcomes by parameter sample

```
# Define cost and QALYs as functions

state_c_matrix <- function() {
  matrix(c(cAsymp(), cProg(), 0,             # without drug
           cAsymp() + cDrug(), cProg(), 0), # with drug
           byrow = TRUE,
           nrow = n_treatments,
           dimnames = list(t_names,
                           s_names))
}

state_q_matrix <- function() {
  matrix(c(uAsymp(), uProg(), 0,  # without drug
           uAsymp(), uProg(), 0), # with drug
         byrow = TRUE,
         nrow = n_treatments,
         dimnames = list(t_names,
                         s_names))
}

trans_c_matrix <- function() {
```

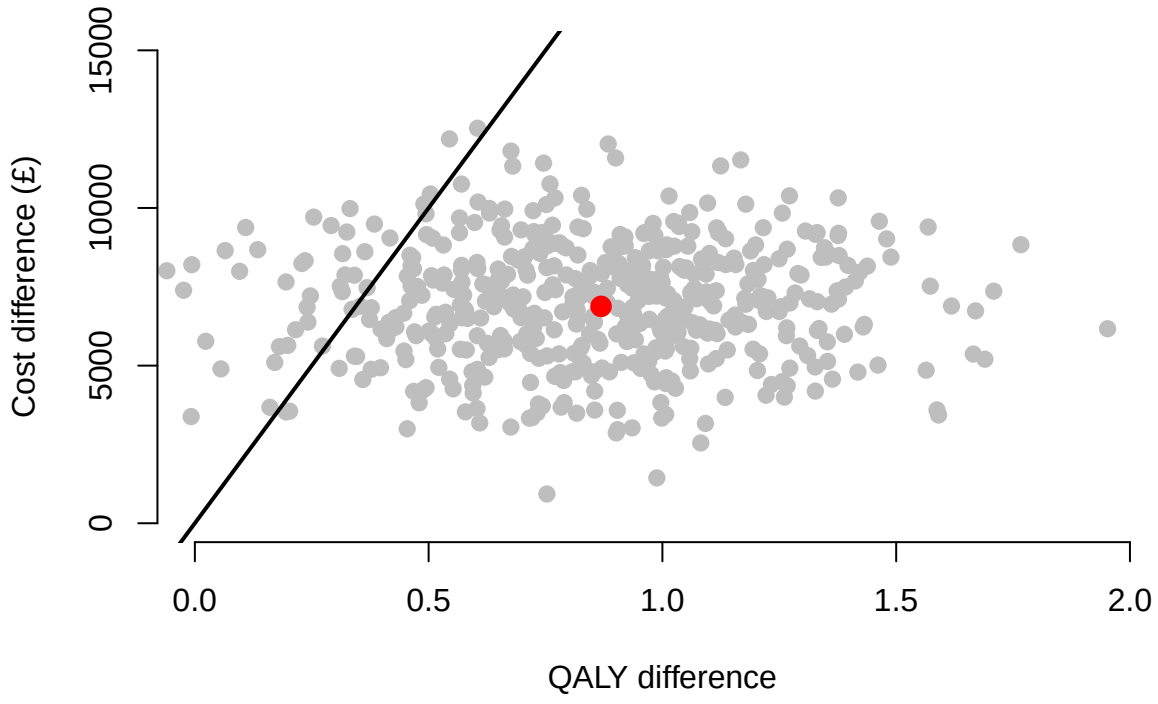
```
  matrix(c(0, 0, 0,          # Asymptomatic_disease
           0, 0, cDeath(),   # Progressive_disease
           0, 0, 0),         # Dead
         byrow = TRUE,
         nrow = n_states,
         dimnames = list(from = s_names,
                         to = s_names))
}
```

4. Run PSA

```
## Run PSA analysis ----

n_trials <- 500

costs <- matrix(NA, nrow = n_trials, ncol = n_treatments,
                dimnames = list(NULL, t_names))
qalys <- matrix(NA, nrow = n_trials, ncol = n_treatments,
                dimnames = list(NULL, t_names))

for (i in 1:n_trials) {
  ce_res <- ce_markov(start_pop = c(n_cohort, 0, 0),
                      p_matrix,
                      state_c_matrix(),
                      trans_c_matrix(),
                      state_q_matrix())

  costs[i, ] <- ce_res$total_costs
  qalys[i, ] <- ce_res$total_QALYs
}
```

5. Plot

```
## Plot results ----

# incremental costs and QALYs of with_drug vs to without_drug
c_incr_psa <- costs[, "with_drug"] - costs[, "without_drug"]
q_incr_psa <- qalys[, "with_drug"] - qalys[, "without_drug"]

plot(x = q_incr_psa/n_cohort, y = c_incr_psa/n_cohort,
     xlim = c(0, 2),
     ylim = c(0, 15e3),
     pch = 16, cex = 1.2,
     col = "grey",
     xlab = "QALY difference",
     ylab = paste0("Cost difference (", enc2utf8("\u00A3"), ")"),
     frame.plot = FALSE)
abline(a = 0, b = wtp, lwd = 2) # Willingness-to-pay threshold
points(x = q_incr/n_cohort, y = c_incr/n_cohort,
       col = "red", pch = 16, cex = 1.5)
```




```
png("figures/ceplane_psa.png", width = 4, height = 4, units = "in", res = 640)
plot(x = q_incr_psa/n_cohort, y = c_incr_psa/n_cohort,
     xlim = c(0, 2),
     ylim = c(0, 15e3),
     pch = 16, cex = 1.2,
     col = "grey",
     xlab = "QALY difference",
     ylab = paste0("Cost difference (", enc2utf8("\u00A3"), ")"),
     frame.plot = FALSE)
abline(a = 0, b = wtp, lwd = 2) # Willingness-to-pay threshold
points(x = q_incr/n_cohort, y = c_incr/n_cohort,
       col = "red", pch = 16, cex = 1.5)
dev.off()
```

```
## pdf
##   2
```