Conducting CEA in R

Ashutosh Kumar

2024-10-10

In this document, we are calculating incremental cost-effectiveness ratio (ICER) using simple markov model.

1. Load packages

```
if (!require('pacman')) install.packages('pacman'); library(pacman)
```

```
## Loading required package: pacman
```

```
# load (install if required) packages from CRAN
p_load("kableExtra", "magrittr", "ggplot2", "dplyr", "stargazer")
```

2. Define transition matrix with Standard of Care (SoC)

```
p_hd <- 0.002   # constant probability of dying when Healthy (all-cause mortality)
p_hs1 <- 0.15   # probability of becoming Sick when Healthy
p_s1h <- 0.5    # probability of becoming Healthy when Sick
p_s1s2 <- 0.105 # probability of becoming Sicker when Sick
p_s1d <- 0.006  # constant probability of dying when Sick
p_s2d <- 0.02   # constant probability of dying when Sicker
p_hh <- 1 - p_hs1 - p_hd
p_s1s1 <- 1 - p_s1h - p_s1s2 - p_s1d
p_s2s2 <- 1 - p_s2d

p_soc <- matrix(
c(p_hh, p_hs1, 0, p_hd,
p_s1h, p_s1s1, p_s1s2, p_s1d,
0, 0, p_s2s2, p_s2d,
0, 0, 0, 1),
byrow = TRUE,
nrow = 4, ncol = 4
)
state_names <- c("H", "S1", "S2", "D")
colnames(p_soc) <- rownames(p_soc) <- state_names
print(p_soc)
```

```
##        H    S1    S2     D
```

```
## H  0.848 0.150 0.000 0.002
## S1 0.500 0.389 0.105 0.006
## S2 0.000 0.000 0.980 0.020
## D  0.000 0.000 0.000 1.000
```

3. Relative risk and transition matrix with new treatment

```r
apply_rr <- function(p, rr = .8){
  p["H", "S1"] <- p["H", "S1"] * rr
  p["H", "S2"] <- p["H", "S2"] * rr
  p["H", "D"] <- p["H", "D"] * rr
  p["H", "H"] <- 1 - sum(p["H", -1])

  p["S1", "S2"] <- p["S1", "S2"] * rr
  p["S1", "D"] <- p["S1", "D"] * rr
  p["S1", "S1"] <- 1 - sum(p["S1", -2])

  p["S2", "D"] <- p["S2", "D"] * rr
  p["S2", "S2"] <- 1 - sum(p["S2", -3])

  return(p)
}
p_new <- apply_rr(p_soc, rr = .8)
print(p_new)
```

```
##         H     S1    S2      D
## H  0.8784 0.1200 0.000 0.0016
## S1 0.5000 0.4112 0.084 0.0048
## S2 0.0000 0.0000 0.984 0.0160
## D  0.0000 0.0000 0.000 1.0000
```

4. Utility and costs for SOC and new treatment

```r
utility <- c(1, .75, 0.5, 0)
costs_medical <- c(2000, 4000, 15000, 0)
costs_treat_soc <- c(rep(2000, 3), 0)
costs_treat_new <- c(rep(12000, 3), 0)
```

5. Simulation

```r
x_init <- c(1, 0, 0, 0)

# State vector at cycle 1
x_init %*% p_soc
```

```
##          H   S1 S2     D
## [1,] 0.848 0.15  0 0.002
```

```r
# State vector at cycle 2
x_init %*%
  p_soc %*%
  p_soc
```

```
##              H      S1      S2        D
## [1,] 0.794104 0.18555 0.01575 0.004596
```

```r
# Simulating state vectors for multiple cycles
simulate_sv <- function (x0, p, n_cycles = 85)
 {
     x <- matrix(NA, ncol = length(x0), nrow = n_cycles)
     x <- rbind(x0, x)
     colnames(x) <- colnames(p)
     rownames(x) <- 0:n_cycles
     for (t in 1:n_cycles) {
         x[t + 1, ] <- x[t, ] %*% p
     }
     return(x)
 }

x_soc <- simulate_sv(x_init,p = p_soc, n_cycles = 85)
x_new <- simulate_sv(x_init,p = p_new, n_cycles = 85)

head(x_soc)
```

```
##           H        S1          S2           D
## 0 1.0000000 0.0000000 0.00000000 0.000000000
## 1 0.8480000 0.1500000 0.00000000 0.002000000
## 2 0.7941040 0.1855500 0.01575000 0.004596000
## 3 0.7661752 0.1912946 0.03491775 0.007612508
## 4 0.7453638 0.1893399 0.05430532 0.010990981
## 5 0.7267385 0.1854578 0.07309990 0.014703854
```

```r
head(x_new)
```

```
##           H        S1          S2           D
## 0 1.0000000 0.0000000 0.00000000 0.000000000
## 1 0.8784000 0.1200000 0.00000000 0.001600000
## 2 0.8315866 0.1547520 0.01008000 0.003581440
## 3 0.8078416 0.1634244 0.02291789 0.005816068
## 4 0.7913203 0.1641411 0.03627885 0.008259738
## 5 0.7771663 0.1624533 0.04948624 0.010894190
```

6. Expected costs and quality-adjusted life-years (QALY)

```r
# Function to calculate present value
cal_pv <- function (z, dr, t)
 {
     z/(1 + dr)^t
```

```r
}

# QALYs
x_soc[2, ] # State occupancy probabilities after 1st cycle
```

```
##     H    S1    S2     D
## 0.848 0.150 0.000 0.002
```

```r
invisible(sum(x_soc[2, 1:3]))             # Expected life-years after 1st cycle for SOC
invisible(sum(x_soc[2, ] * utility))      # Expected utility after 1st cycle for SOC
sum(cal_pv(x_soc[2, ] * utility, .03, 1)) # Expected discounted utility after 1st cycle for SOC
```

```
## [1] 0.9325243
```

```r
# Function to compute expected (discounted) QALYs for each cycle
compute_qalys <- function(x, utility, dr = .03){
  n_cycles <- nrow(x) - 1
  cal_pv(x %*% utility, dr, 0:n_cycles)
}
# Non-discounted QALYS
qalys_soc <- x_soc %*% utility

# Discounted QALYS
dqalys_soc <- compute_qalys(x_soc, utility = utility)
dqalys_new <- compute_qalys(x_new, utility = utility)

head(qalys_soc)
```

```
##        [,1]
## 0 1.0000000
## 1 0.9605000
## 2 0.9411415
## 3 0.9271050
## 4 0.9145214
## 5 0.9023818
```

```r
head(dqalys_soc)
```

```
##        [,1]
## 0 1.0000000
## 1 0.9325243
## 2 0.8871161
## 3 0.8484324
## 4 0.8125404
## 5 0.7784024
```

```r
head(dqalys_new)
```

```
##        [,1]
## 0 1.0000000
```

```
## 1 0.9401942
## 2 0.8980022
## 3 0.8619435
## 4 0.8285724
## 5 0.7968343
```

7. Costs

```r
# Function to compute discounted costs taking inputs for medical and treatment costs
compute_costs <- function(x, costs_medical, costs_treat, dr = .03){
  n_cycles <- nrow(x) - 1
  costs <- cbind(
    cal_pv(x %*% costs_medical, dr, 0:n_cycles),
    cal_pv(x %*% costs_treat, dr, 0:n_cycles)
  )
  colnames(costs) <- c("medical", "treatment")
  return(costs)
}

# Calculate discounted costs
dcosts_soc <- compute_costs(x_soc, costs_medical, costs_treat_soc)
dcosts_new <- compute_costs(x_new, costs_medical, costs_treat_new)

head(dcosts_soc)
```

```
##    medical treatment
## 0 2000.000  2000.000
## 1 2229.126  1937.864
## 2 2419.321  1876.527
## 3 2581.884  1816.350
## 4 2721.140  1757.443
## 5 2839.541  1699.850
```

```r
head(dcosts_new)
```

```
##    medical treatment
## 0 2000.000  12000.00
## 1 2171.650  11631.84
## 2 2293.695  11270.64
## 3 2391.402  10917.83
## 4 2473.004  10573.78
## 5 2541.624  10238.54
```

8. Cost-effectiveness Analysis

```r
ICER <- (sum(dcosts_new[-1, ]) - sum(dcosts_soc[-1, ])) /
(sum(dqalys_new[-1, ]) - sum(dqalys_soc[-1, ]))

print(ICER)
```

```
## [1] 122946.8
```

9. Get a nice looking table

```r
format_costs <- function(x) formatC(x, format = "d", big.mark = ",")
format_qalys <- function(x) formatC(x, format = "f", digits = 2)
make_icer_tbl <- function(costs0, costs1, qalys0, qalys1){
  # Computations
  total_costs0 <- sum(costs0)
  total_costs1 <- sum(costs1)
  total_qalys0 <- sum(qalys0)
  total_qalys1 <- sum(qalys1)
  incr_total_costs <- total_costs1 - total_costs0
  inc_total_qalys <- total_qalys1 - total_qalys0
  icer <- incr_total_costs/inc_total_qalys

  # Make table
  tibble(
    `Strategy` = c("SOC", "New"),
    `Costs` = c(total_costs0, total_costs1) %>%
      format_costs(),
    `QALYs` = c(total_qalys0, total_qalys1) %>%
      format_qalys(),
    `Incremental costs` = c("--", incr_total_costs %>%
                            format_costs()),
    `Incremental QALYs` = c("--", inc_total_qalys %>%
                            format_qalys()),
    `ICER` = c("--", icer %>% format_costs())
  ) %>%
    kable() %>%
    kable_styling() %>%
    footnote(general = "Costs and QALYs are discounted at 3% per annum.",
             footnote_as_chunk = TRUE)
}
make_icer_tbl(costs0 = dcosts_soc[-1, ], costs1 = dcosts_new[-1, ],
              qalys0 = dqalys_soc[-1, ], qalys1 = dqalys_new[-1, ])
```

Strategy	Costs	QALYs	Incremental costs	Incremental QALYs	ICER
SOC	204,123	21.08	–	–	–
New	464,390	23.19	260,266	2.12	122,946

Note: Costs and QALYs are discounted at 3% per annum.