Markov models for cost-effectiveness analysis in R

*Fernando Alarid-Escudero, PhD**
Eric Jutkowitz, PhD†
*Karen M. Kuntz, ScD**

2018-01-28

Contents

This document shows an example of how to conduct decision-analytic models in R (Jalal et al. 2017) using the R package `dampack`.

The R package `dampack` has a couple functions that could be used to evaluate Markov models. Specifically, the function `CalculateMarkovTrace` takes a transition probability matrix (time-homogeneous) or array (time-dependent) and then simulates a cohort of hypothetical individuals, and returns a cohort trace and transition array. The function `ExpectedValue` takes a transition array and rewards matrix, and calculates expected outcomes. The package can be used to compare competing strategies and calculate their cost-effectiveness. Below, we provide an applied example of these functions.

1 Model Description

We will develop a 4-state Markov model to quantify the expected outcomes for a cohort of 55-year-old women who have undergone tumor excision for localized breast cancer (BCA). After surgery, all patients are initially tumor-free (i.e., will start in the "Local" Markov state). Each year, patients in the Local state face a 2% chance that they will experience a recurrence. Of these recurrences, 75% will present as metastatic disease (and patients will enter the "Mets"" Markov state), and the remainder present as local disease (and patients will enter the "Recur"" Markov state). Patients in the Recur state undergo repeat resection. Following this operation, they face a 6% chance of recurrence each year. (Assume that all local recurrences - 1st, 2nd, 3rd, etc. - have equivalent prognoses.) Of these recurrences, 90% will present as metastatic disease. Only patients

---

*Division of Health Policy and Management, University of Minnesota School of Public Health, Minneapolis, MN, USA
†Department of Health Services, Policy and Practice, Brown University School of Public Health, Providence, RI, USA

with metastatic disease can die of breast cancer. Finally, all patients in the local and recur state face a 7.6% annual probability of mortality and patients in the mets state face a 39% added risk of death.

The Markov model described above represents usual care for women with breast cancer. Once we construct that Markov model, we will then add a second strategy of a hypothetical new treatment. The effect of this new treatment is to reduce the initial rate of cancer recurrence by 50%; however, the treatment also lowers the quality of life of women (disutility of 0.05) taking the medication because of side effects. Treatment is only taken by women who have not experienced a recurrence (i.e., those who reside in the Local state). The Markov model for the treatment strategy will be structurally identical to that of the usual care strategy.

After comparing the two strategies based on quality-adjusted life expectancy (QALE) and costs, we will incorporate costs and calculate the incremental cost per quality-adjusted life year (QALY) gained of treatment compared with usual care.

2 Markov structure: Usual care

The cancer model consists of 4-states (Local, Mets, Recur, and Dead). Therefore, we will be creating a 4x4 transition matrix. We start by labeling the states in the transition matrix.

```
state.names <- c("Local", "Recur", "Mets", "Dead")
```

We then define the number of states in the model.

```
n.s <- length(state.names)
```

From the decision problem, we know that patients in the local state face a 2% chance of recurrence under usual care (defined as `p.BCA1`). Of these recurrences, 75% will present as metastatic disease (and patients will enter the "Mets" Markov state; defined as `p.Mets1`), and the remainder present as local disease (and patients will enter the "Recur" Markov state; i.e., `1-p.Mets1`). We will create variables to represent these probabilities of the transition matrix. Patients in the recur state face a 6% chance of recurrence each year (defined as `p.BCA2`). Of these recurrences, 90% will present as metastatic disease (defined as `p.Mets2`).

All patients in the local and recur state face a 0.076 annual rate of dying from non-BCA-related causes, μ_{Die} (defined as `mu.Die`). Patients in the Mets state face a 0.5 anual rate of of dying of metastatic cancer in addition to dying from other causes, $\mu_{DieMets}$ (defined as `mu.DieMets`).

To compute the annual probability of dying from non-BCA-related causes (defined as `p.Die`) we transform the annual rate assuming constant exponential rate

$$p_{Die} = 1 - exp(-\mu_{Die})$$

The overall annual mortality rate for women with metastatic disease is equal to the sum of the rates of dying from other causes and the excess disease-specific mortality rate (defined as muMets). Therefore, the annual probability of dying in the Mets state (defined as `p.DieMets`) is converted by summing these rates to an convert them to annual probabilities

$$p_{DieMets} = 1 - exp(-(\mu_{Die} + \mu_{DieMets}))$$

The annual costs of breast cancer in the Local, Recur and Mets states (`c.Local`, `c.Recur`, and `c.Mets`, respectively) will be the same for both strategies. Table 1 shows parameters of the model of usual care.

Table 1: Table of parameters.

Parameter	R name	Value
Time horizon (n_t)	n.t	51 years

Parameter	R name	Value
Annual discount rate (costs/QALYs)	d.c/d.e	3%
Annual probability of recurrence after first local tumor diagnosis under usual care	p.BCA1	0.02
Proportion of metastatic disease of recurrences from the Local state	p.Mets1	0.75
Annual probability of recurrence after > 1 local tumor diagnoses	p.BCA2	0.06
Proportion of metastatic disease of recurrences from the Recur state	p.Mets2	0.90
Annual mortality		
- All-cause mortality rate	mu.Die	0.08 or age-specific
- Cancer specific mortality rate	mu.DieMets	0.5
Annual costs		
- Cost of new treatment	c.Rx	1000
- Local state	c.Local	500
- Recur state	c.Recur	5000
- Mets state	c.Mets	20000
Utilities		
- Local state under usual care	u.Local	0.95
- Recur state	u.Recur	0.80
- Mets state	u.Mets	0.40
- Dead state	u.Dead	0.00

We input the parameters as follows:

```r
n.t        <- 51
d.c        <- 0.03
d.e        <- 0.03
p.BCA1     <- 0.02
p.Mets1    <- 0.75
p.BCA2     <- 0.06
p.Mets2    <- 0.90
mu.Die     <- 0.08 # Mean hazard
p.Die      <- 1 - exp(-mu.Die)
mu.DieMets <- 0.5
p.DieMets  <- 1 - exp(-(mu.Die + mu.DieMets))
u.Local    <- 0.95
u.Recur    <- 0.80
u.Mets     <- 0.40
u.Dead     <- 0.00
c.Local    <- 500
c.Recur    <- 5000
c.Mets     <- 20000
c.Rx       <- 1000
```

2.1 Transition matrix

To construct the transition matrix for the usual care (UC) strategy, we first need to define a matrix that we will name `m.P.UC`. We initialize the matrix with `NaN` so then it can be filled with numeric elements.

```r
# Initialize matrix
m.P.UC <- matrix(NaN,
                 nrow = n.s, ncol = n.s,
                 dimnames = list(state.names, state.names))
m.P.UC
```

```
##       Local Recur Mets Dead
## Local   NaN   NaN  NaN  NaN
## Recur   NaN   NaN  NaN  NaN
## Mets    NaN   NaN  NaN  NaN
## Dead    NaN   NaN  NaN  NaN
```

Each row in `m.P.UC` represent a health state and each of its elements store the transition probbailities from that particular state. We need to manually store each of these transition probabilities into the matrix.

```r
# Fill in matrix
# From Local
m.P.UC["Local", "Local"] <- (1 - p.Die) * (1 - p.BCA1)
m.P.UC["Local", "Recur"] <- (1 - p.Die) * p.BCA1 * (1 - p.Mets1)
m.P.UC["Local", "Mets"]  <- (1 - p.Die) * p.BCA1 * p.Mets1
m.P.UC["Local", "Dead"]  <- p.Die
# From Recur
m.P.UC["Recur", "Local"] <- 0
m.P.UC["Recur", "Recur"] <- (1 - p.Die) * ((1 - p.BCA2) + (p.BCA2 * (1 - p.Mets2)))
m.P.UC["Recur", "Mets"]  <- (1-p.Die)*p.BCA2*p.Mets2
m.P.UC["Recur", "Dead"]  <- p.Die
# From Mets
m.P.UC["Mets", "Local"]  <- 0
m.P.UC["Mets", "Recur"]  <- 0
m.P.UC["Mets", "Mets"]   <- 1 - p.DieMets
m.P.UC["Mets", "Dead"]   <- p.DieMets
# From Dead
m.P.UC["Dead", "Local"]  <- 0
m.P.UC["Dead", "Recur"]  <- 0
m.P.UC["Dead", "Mets"]   <- 0
m.P.UC["Dead", "Dead"]   <- 1

m.P.UC
```

```
##          Local       Recur       Mets       Dead
## Local 0.904654 0.004615582 0.01384675 0.07688365
## Recur 0.000000 0.873268064 0.04984828 0.07688365
## Mets  0.000000 0.000000000 0.55989837 0.44010163
## Dead  0.000000 0.000000000 0.00000000 1.00000000
```

We assume that all women start in the Local state, therefore the initial state vector defined as 's0'

```r
s0 <- c(1, 0, 0, 0) # Everyone starts in the local state
```

2.2 Run Markov model

All of the components necessary to run the Markov model are now in place. We use the function `CalculateMarkovTrace` from the `dampack` package to run the model. The function requires a transition probability matrix, `m.P.UC`, in this case, the initial state vector (`s0`), and the number of cycles we want the model to run, `n.t`. The function produces a list with the cohort trace and the transition array, which we will

call `m.Markov.UC` for the UC strategy.

```
l.Markov.UC <- CalculateMarkovTrace(M = m.P.UC, p0 = s0, n.cycles = n.t)
```

2.3 Rewards matrices

To calculate life years, quality adjusted life years, and cost, we will specify a reward matrix for each of these outcomes. The elements of each of the columns of these matrices must contain the state rewards of the health state corresponding to that column.

For the life years, for each year spent in any of the non-dead states, the cohort will get one year of life. That is, the elements of all the columns corresponding non-dead states equal to one.

```
m.LY.UC <- matrix(rep(c(1, 1, 1, 0), (n.s-1)),
                  byrow = TRUE,
                  nrow = n.s, ncol = n.s,
                  dimnames = list(state.names, state.names))
m.LY.UC
```

```
##       Local Recur Mets Dead
## Local     1     1    1    0
## Recur     1     1    1    0
## Mets      1     1    1    0
## Dead      1     1    1    0
```

For the QALY, we follow the similar procedure for the LY's reward matrix but now each column has the utilities that correspond with that particular state

```
m.QALY.UC <- matrix(rep(c(u.Local, u.Recur, u.Mets, u.Dead), n.s),
                    byrow = TRUE,
                    nrow = n.s, ncol = n.s,
                    dimnames = list(state.names, state.names))
m.QALY.UC
```

```
##       Local Recur Mets Dead
## Local  0.95   0.8  0.4    0
## Recur  0.95   0.8  0.4    0
## Mets   0.95   0.8  0.4    0
## Dead   0.95   0.8  0.4    0
```

```
m.C.UC <- matrix(rep(c(c.Local, c.Recur, c.Mets, 0), n.s),
                 byrow = TRUE,
                 nrow = n.s, ncol = n.s,
                 dimnames = list(state.names, state.names))
m.C.UC
```

```
##       Local Recur  Mets Dead
## Local   500  5000 20000    0
## Recur   500  5000 20000    0
## Mets    500  5000 20000    0
## Dead    500  5000 20000    0
```

2.4 Compute expected values

Using the `ExpectedValue` function from the `dampack` package, we calculate the expected life years of the Markov cohort under usual care. In the expected value function we input the transition array (generated

from the `CalculateMarkovTrace` function), the reward matrix for life years `m.LY.UC`, discount rate `d.e` and ask to conduct a half cycle correction.

```
ev.LY.UC <- ExpectedValue(trans = l.Markov.UC$trans,
                          rwd = m.LY.UC,
                          r = d.e, half = T)
ev.LY.UC
```

```
## [1] 8.213022
```

To compute the expected QALYs, we follow the same procedure but now using the reward matrix for QALYs `m.QALY.UC`

```
ev.QALY.UC <- ExpectedValue(trans = l.Markov.UC$trans,
                            rwd = m.QALY.UC,
                            r = d.e, half = T)
ev.QALY.UC
```

```
## [1] 7.61944
```

To compute the expected costs, we follow the same procedure but now using the reward matrix for costs `m.C.UC`

```
ev.C.UC <- ExpectedValue(trans = l.Markov.UC$trans,
                         rwd = m.C.UC,
                         r = d.c, half = T)
ev.C.UC
```

```
## [1] 10395.49
```

3 Markov structure: New treatment

We will now model the effect of a hypothetical new treatment. This new treatment competes with usual care. From the decision problem, we know that the treatment is only taken by women who have not experienced a recurrence (i.e., those who reside in the Local state).

The new treatment reduces the initial rate of cancer recurrence by 50% (i.e., a hazard ratio or recurrence under new treatment, defined as `hr.Recur`). To compute the probability of cancer recurrence associated with treatment (defined as `p.BCA1.Rx`), we first transform the probability of cancer recurrence without treatment to a rate. We then multiply the rate of cancer recurrence without treatment by the treatment effect (`hr.Recur = 0.50`). Finally, we convert the rate back to a probability to represent the probability of cancer recurrence for those who receive treatment.

```
hr.Recur   <- 0.5
p.BCA1.Rx  <- 1-exp(log(1-p.BCA1)*hr.Recur)
```

3.1 Transition matrix

We will incorporate the effects of treatment in a new transition matrix (defined as `m.P.Rx`). The treatment transition matrix will incorporate the treatment effects which impact transitions from the local disease state to the disease recurrence state and from the local disease state to the metastatic disease state. All other transitions in the model are identical to those in the usual care transition matrix (Table 1).

As before, each row in the transition matrix represents a health state and each element of the matrix store the transition probabilities. Ultimately, we will be able to compare the expected outcomes between the usual care and treatment models. We manually enter the transition probabilities into the matrix.

```r
# Initialize matrix
m.P.Rx <- matrix(NaN,
                 nrow = n.s, ncol = n.s,
                 dimnames = list(state.names, state.names))
m.P.Rx
```

```
##       Local Recur Mets Dead
## Local   NaN   NaN  NaN  NaN
## Recur   NaN   NaN  NaN  NaN
## Mets    NaN   NaN  NaN  NaN
## Dead    NaN   NaN  NaN  NaN
```

```r
# Fill in matrix
# From Local
m.P.Rx["Local", "Local"] <- (1 - p.Die) * (1 - p.BCA1.Rx)
m.P.Rx["Local", "Recur"] <- (1 - p.Die) * p.BCA1.Rx * (1 - p.Mets1)
m.P.Rx["Local", "Mets"]  <- (1 - p.Die) * p.BCA1.Rx * p.Mets1
m.P.Rx["Local", "Dead"]  <- p.Die
# From Recur
m.P.Rx["Recur", "Local"] <- 0
m.P.Rx["Recur", "Recur"] <- (1 - p.Die) * ((1 - p.BCA2) + (p.BCA2 * (1 - p.Mets2)))
m.P.Rx["Recur", "Mets"]  <- (1-p.Die)*p.BCA2*p.Mets2
m.P.Rx["Recur", "Dead"]  <- p.Die
# From Mets
m.P.Rx["Mets", "Local"]  <- 0
m.P.Rx["Mets", "Recur"]  <- 0
m.P.Rx["Mets", "Mets"]   <- 1 - p.DieMets
m.P.Rx["Mets", "Dead"]   <- p.DieMets
# From Dead
m.P.Rx["Dead", "Local"]  <- 0
m.P.Rx["Dead", "Recur"]  <- 0
m.P.Rx["Dead", "Mets"]   <- 0
m.P.Rx["Dead", "Dead"]   <- 1

m.P.Rx
```

```
##           Local       Recur       Mets       Dead
## Local 0.9138386 0.002319447 0.00695834 0.07688365
## Recur 0.0000000 0.873268064 0.04984828 0.07688365
## Mets  0.0000000 0.000000000 0.55989837 0.44010163
## Dead  0.0000000 0.000000000 0.00000000 1.00000000
```

3.2   Run Markov model

```r
l.Markov.Rx <- CalculateMarkovTrace(M = m.P.Rx, p0 = s0, n.cycles = n.t)
```

3.3   Reward matrices

The LY's reward matrix for Rx is the same than for UC.

```r
m.LY.Rx <- m.LY.UC
```

As noted in the decision problem, treatment lowers the quality of life (i.e., utility decrement) of women taking

the medication due to side effects (defined as `du.Local`). We incorporate the disutility (0.05) associated with treatment into the reward matrix for the Local state of the Rx model.

```
du.Local <- 0.05
```

The reward matrix for Rx changes by adding the disutility from the treatment to the Local state.

```
m.QALY.Rx <- matrix(rep(c(u.Local-du.Local, u.Recur, u.Mets, u.Dead), n.s),
                    byrow = TRUE,
                 nrow = n.s, ncol = n.s,
                 dimnames = list(state.names, state.names))
m.QALY.Rx
```

```
##       Local Recur Mets Dead
## Local   0.9   0.8  0.4    0
## Recur   0.9   0.8  0.4    0
## Mets    0.9   0.8  0.4    0
## Dead    0.9   0.8  0.4    0
```

```
m.C.Rx <- matrix(rep(c(c.Local + c.Rx, c.Recur, c.Mets, 0), n.s),
                 byrow = TRUE,
                 nrow = n.s, ncol = n.s,
                 dimnames = list(state.names, state.names))
m.C.Rx
```

```
##       Local Recur  Mets Dead
## Local  1500  5000 20000    0
## Recur  1500  5000 20000    0
## Mets   1500  5000 20000    0
## Dead   1500  5000 20000    0
```

3.4   Compute expected values

```
ev.LY.Rx <- ExpectedValue(trans = l.Markov.Rx$trans,
                          rwd = m.LY.Rx,
                          r = d.e, half = T)
ev.LY.Rx
```

```
## [1] 8.620571
```

To compute the expected QALYs, we follow the same procedure but now using the reward matrix for QALYs `m.QALY.Rx`

```
ev.QALY.Rx <- ExpectedValue(trans = l.Markov.Rx$trans,
                            rwd = m.QALY.Rx,
                            r = d.e, half = T)
ev.QALY.Rx
```

```
## [1] 7.673208
```

To compute the expected costs, we follow the same procedure but now using the reward matrix for costs `m.C.Rx`

```
ev.C.Rx <- ExpectedValue(trans = l.Markov.Rx$trans,
                         rwd = m.C.Rx,
                         r = d.c, half = T)
ev.C.Rx
```

```
## [1] 16061.16
```

4 Outputs

Table 2 shows the outputs of the Markov model for the two strategies.

Table 2: Outcomes of Markov model by strategy

Strategy	LYs	QALYs	Costs
Usual care	8.21	7.62	$10,395.49
New Treatment	8.62	7.67	$16,061.16

5 Cost-effectiveness analysis

Finally, in Table 3, we show the results for the CEA of the Markov model.

Table 3: CEA of new treatment for breast cancer patients

Strategy	QALYs	Incremental.QALYs	Costs	Incremental.Costs	ICER
Usual care	7.62	-	$10,395.49	-	-
New Treatment	7.67	0.05	$16,061.16	$5,665.66	$105,372 /QALY

References

Jalal, Hawre, Petros Pechlivanoglou, Eline Krijkamp, Fernando Alarid-Escudero, Eva A. Enns, and M. G. Myriam Hunink. 2017. "An Overview of R in Health Decision Sciences." *Medical Decision Making* 37 (7): 735–46. doi:10.1177/0272989X16686559.