

Discrete Time Cohort Markov Modelling in R



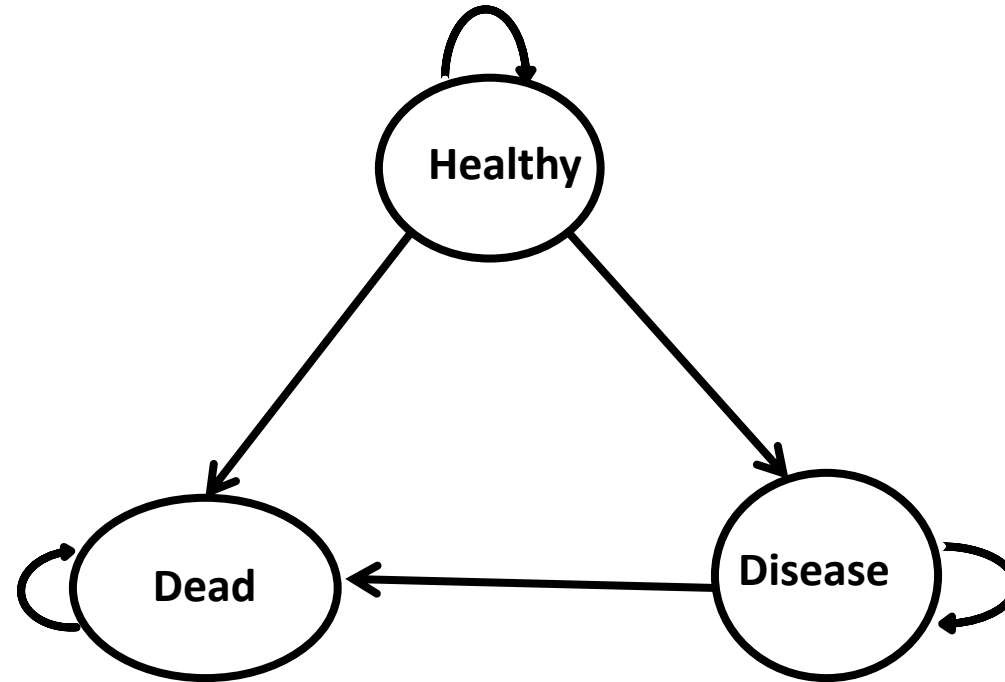
Multistate cohort Markov Models

- Are useful for repeated events over time
 - Modelling disease progression over time
- A special form of a state transition model
- Manageable number of health states
 - Mutually exclusive
 - Discrete
 - Absorbing state or else equilibrium

Cohort Markov Models

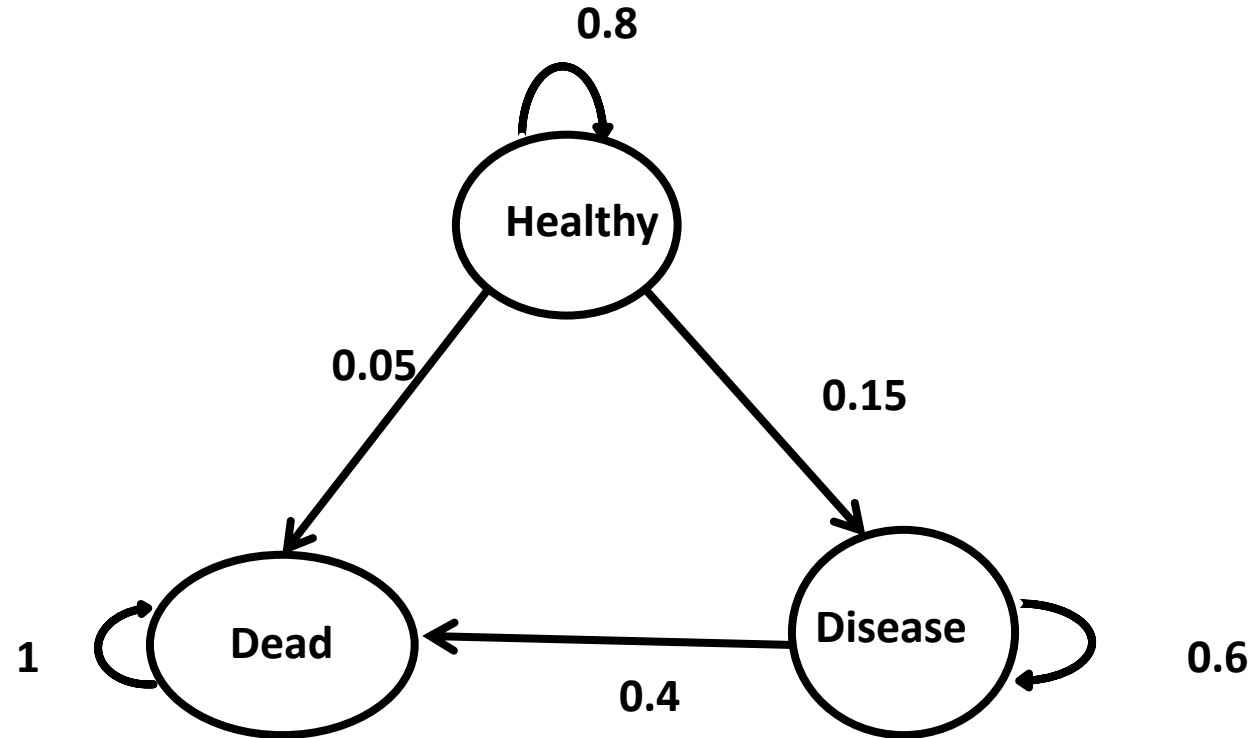
- Markov property – process depends only on present state and not on any previous state
 - Strictly a Markov model has this memory-less property but can get around assumptions!
- Movements between health states using transition probabilities
 - From a state must sum to 1
 - Movement in a time period is called a cycle (often 1 year)
- Evaluate the Markov trace
 - How individuals move over each cycle of the model

Health states



- Arrows show what movements are possible in each cycle
- Health state associated with parameters such as costs & QALYs

Health states



- Transition probabilities add up to one from each health state
- Transition probabilities may be in a matrix form
- Repeatedly apply these probabilities to a cohort of patients over time

Health states

Year		Healthy	Disease	Dead		Total
1		1000				
2		800	150	50		1000
3		640	210	150		1000
4		512	222	266		1000
5		410	210	380		1000
6		328	187	485		1000
7		262	162	576		1000
8		210	136	654		1000
9		168	113	719		1000
10		134	93	773		1000
11		107	76	817		1000
12		86	62	852		1000
13		69	50	881		1000
14		55	40	905		1000
15		44	32	924		1000

$$1000 * 0.8 = 800$$

$$1000 * 0.15 = 150$$

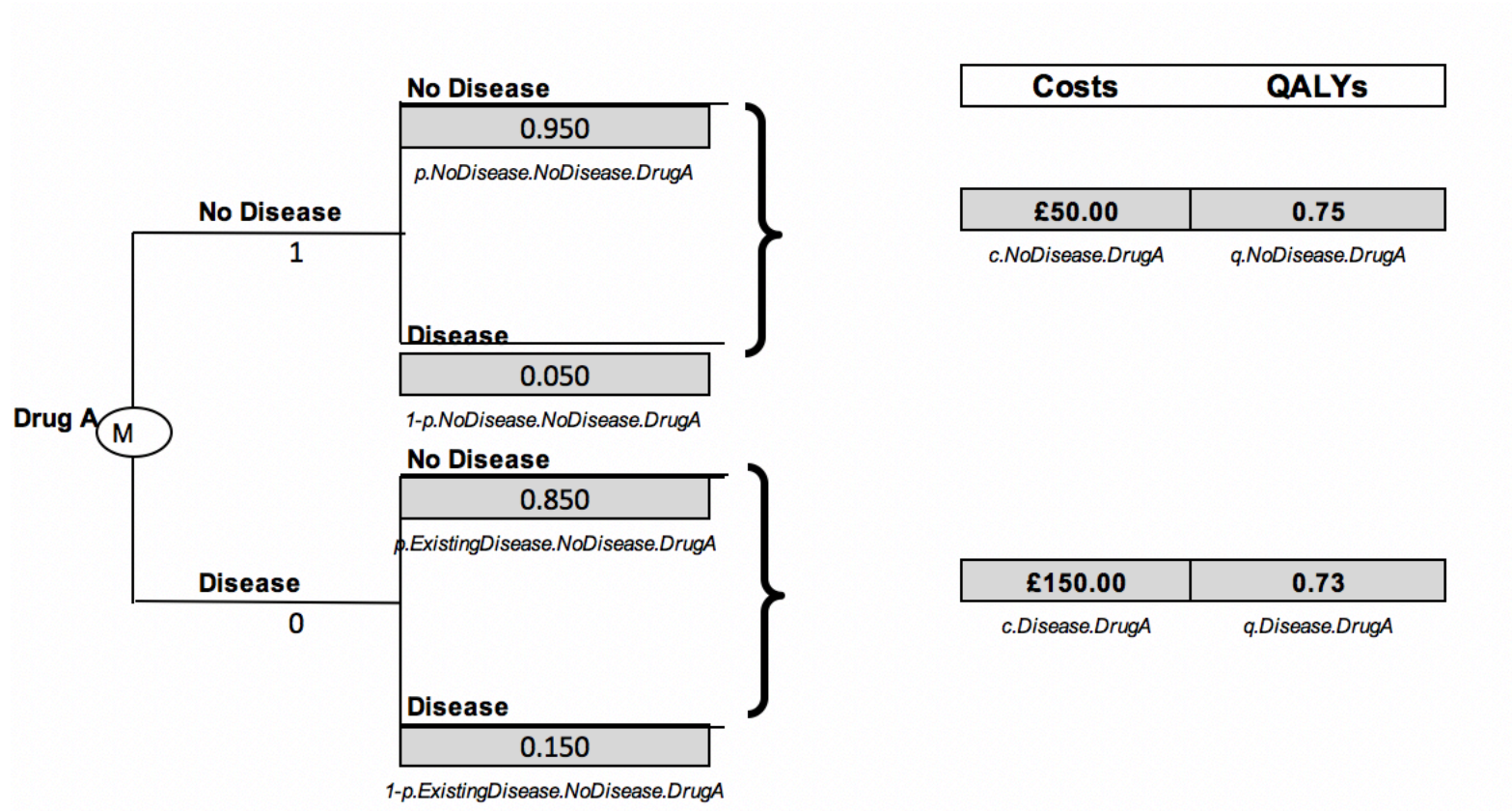
$$1000 * 0.05 = 50$$

$$800 * 0.8 = 640$$

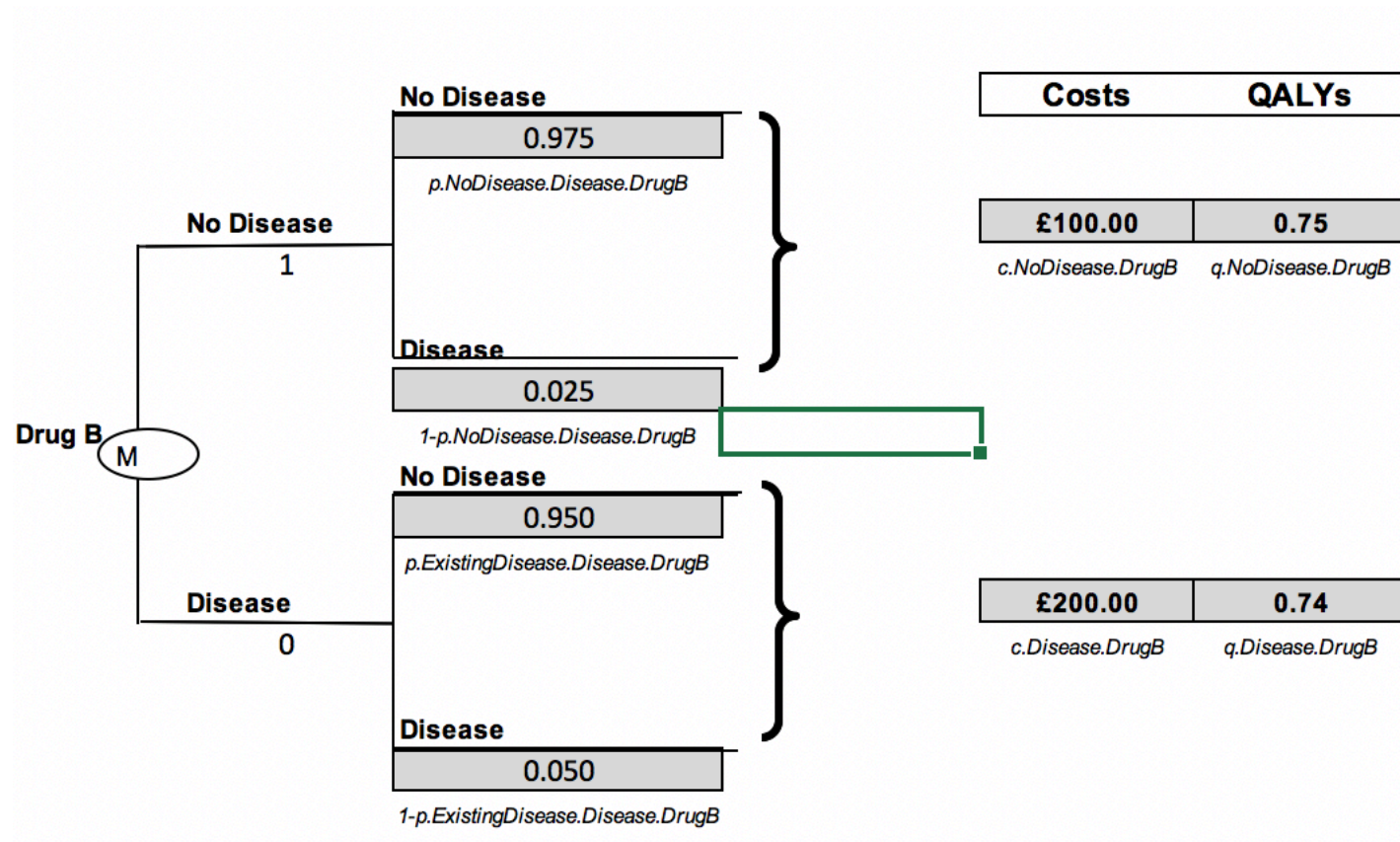
$$800 * 0.15 + 150 * 0.6 = 210$$

$$800 * 0.05 + 150 * 0.4 + 50 * 1 = 150$$

A Model in Excel – Drug A



Drug B



The Model in R

```
# Firstly, let's define the number of treatments we are looking at and their names  
n.treatments<-2  
t.names<-c("Drug A","Drug B")
```

```
# Let's also define the number of states we are looking at and their names  
n.states<-2  
s.names<-c("Disease","No Disease")
```

```
# In a Markov model we want to specify the number of cycles  
n.cycles<-12
```

Costs

```
# Now let's start with adding the cost information
```

```
c.matrix<-array(NA,dim=c(n.treatments, n.states),  
                dimnames=list(t.names,s.names))
```

```
c.matrix["Drug A","Disease"]<-150  
c.matrix["Drug A","No Disease"]<-50
```

```
c.matrix["Drug B","Disease"]<-200  
c.matrix["Drug B","No Disease"]<-100
```

QALYs

```
# Let's add in the QALY information

q.matrix<-array(NA,dim=c(n.treatments,n.states),
               dimnames=list(t.names,s.names))

q.matrix["Drug A","Disease"]<-0.73
q.matrix["Drug A","No Disease"]<-0.75

q.matrix["Drug B","Disease"]<-0.74
q.matrix["Drug B","No Disease"]<-0.75
```

Transition Probabilities

```
# Let's add in the probabilities
```

```
p.matrix<-array(NA,dim=c(n.treatments,n.states,n.states),  
               dimnames=list(t.names,s.names,s.names))
```

```
# The matrix containing transition probabilities for Drug A
```

```
p.matrix["Drug A","Disease","No Disease"]<-0.85  
p.matrix["Drug A","Disease","Disease"]<-1-p.matrix["Drug A","Disease","No Disease"]  
  
p.matrix["Drug A","No Disease","No Disease"]<-0.95  
p.matrix["Drug A","No Disease","Disease"]<-1-p.matrix["Drug A","No Disease","No Disease"]
```

```
# The matrix containing transition probabilities for Drug B
```

```
p.matrix["Drug B","Disease","No Disease"]<-0.95  
p.matrix["Drug B","Disease","Disease"]<-1-p.matrix["Drug B","Disease","No Disease"]  
  
p.matrix["Drug B","No Disease","No Disease"]<-0.975  
p.matrix["Drug B","No Disease","Disease"]<-1-p.matrix["Drug B","No Disease","No Disease"]
```

Cycles

```
# We want to store the information for each cycle

cycle.info<-array(NA,dim=c(n.treatments,n.cycles,n.states),
                 dimnames=list(t.names,NULL,s.names))

# Everyone starts without the disease
cycle.info[,1,"No Disease"]<-1
cycle.info[,1,"Disease"]<-0

# We add up the costs and QALYs for each cycle at a time for each drug

cycle.costs<-array(NA,dim=c(n.treatments,n.cycles),
                  dimnames=list(t.names,NULL))
cycle.QALYs<-array(NA,dim=c(n.treatments,n.cycles),
                  dimnames=list(t.names,NULL))

# The total costs and QALYs need to also be stored

total.costs<-array(NA,dim=c(n.treatments),
                  dimnames=list(t.names))
total.QALYs<-array(NA,dim=c(n.treatments),
                  dimnames=list(t.names))
```

Loop over cycles

```
for(i.treatment in 1:n.treatments)
{

  for(i.cycle in 2:n.cycles)
  {

    cycle.info[i.treatment,i.cycle,]<-
      cycle.info[i.treatment,i.cycle-1,]%*%
      p.matrix[i.treatment,,]
  }

  cycle.QALYs[i.treatment,]<-
    cycle.info[i.treatment,,]%*%q.matrix[i.treatment,]

  cycle.costs[i.treatment,]<-
    cycle.info[i.treatment,,]%*%c.matrix[i.treatment,]

  total.costs[i.treatment]<-sum(cycle.costs[i.treatment,])

  total.QALYs[i.treatment]<-sum(cycle.QALYs[i.treatment,])
}
```

Results

```
# Let's look at the results
```

```
# Incremental costs and QALYs of Drug B compared to Drug A
```

```
incremental.costs<-total.costs["Drug B"]-total.costs["Drug A"]
```

```
incremental.QALYs<-total.QALYs["Drug B"]-total.QALYs["Drug A"]
```

```
# Calculate the incremental cost effectiveness ratio
```

```
ICER<-incremental.costs/incremental.QALYs
```