Exercises chapter 6

Andreas Slåttelid

Exercise 6.1

In this exercise we are asked to calculate transition rates using an Euler scheme, we have four states, i.e $S = \{*, \diamond, \times, \dagger\}$, where $\times$ represents critically ill. From the text, we have the following transition rate matrix:

```r
#required libraries:
library(tidyverse)
library(scales)
```

```r
#0: alive, 1: disabeld, 2: critcally ill, 3: dead
lambda <- function(t){

  #constants given in exercise:
  a1 <- 4*10**(-4)
  b1 <- 3.4674*10**(-6)
  c1 <- 0.138155
  a2 <- 5*10**(-4)
  b2 <- 7.5858*10**(-5)
  c2 <- 0.08749

  #alive
  m01 <- a1 + b1*exp(c1*t)
  m02 <- 0.05*m01
  m03 <- a2 + b2*exp(c2*t)
  m00 <- -(m01 + m02 + m03)

  #disabeld
  m10 <- 0.1*m01
  m12 <- m02
  m13 <- m03
  m11 <- -(m10 + m12 + m13)

  #critically ill
  m23 <- 1.2*m13
  m21 <- 0
  m20 <- 0
  m22 <- -(m23 + m21 +m20)

  m30 <- m31 <- m32 <- m33 <- 0
  trans_rates <- c(m00, m01, m02, m03, m10, m11, m12, m13, m20, m21,m22,m23, m30, m31,m32,m33)

  L <- matrix(trans_rates, nrow=4, byrow=TRUE)

  return(L)
}
```

```r
field <- function(t,M){
  return(M%*%lambda(t))
}

#field(t0, P0)

Euler <- function(t0,P0,h,tn){
  if(t0==tn){ return(P0)}
  N <- (tn-t0)/h
  D <- dim(P0)[1] #gives dimension of mxn matrix dim = c(m,n)
  #Initial condition at s
  P <- array(diag(D*(N+1)), dim=c(D,D,N+1))
  #First iteration
  P[,,1] <- P0 + h*field(t0,P0)


  for(n in 1:N){
    P[,,n+1] <- P[,,n]+h*field(t0+n*h,P[,,n])
  }
  return(P) #returns array, array(data , dim= c(3,3,2)), this stores 2 3x3 matricies
}

t0 <- 30       #start age
P0 <- diag(4)  #initial start with P(s,s) = I
tn <- 110      #end age
h <- 1/12      #step size
N <- (tn-t0)/h #number of steps

sol <- Euler(t0,P0,h,tn) #contains ages transition probs from 30 to 110
#p_surv(30,75):
sol[1,1, ][12*35]
```

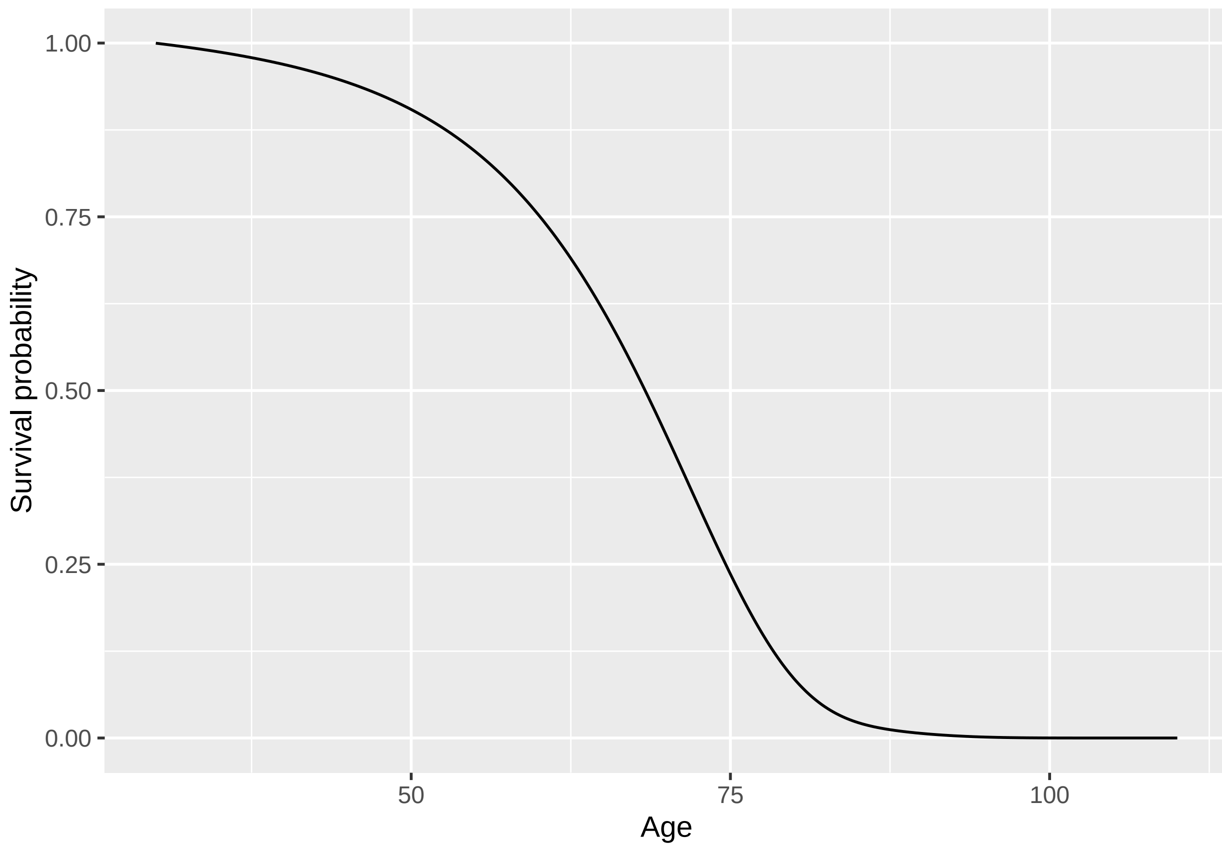
```
## [1] 0.6189569
```

```r
a <- seq(30, 110, length = length(sol[1,1, ]))
b <- sol[1,1, ]


df <- data.frame(a,b)
colnames(df) <- c("age", "survival_prob")

fig <- df %>%
  ggplot(aes(x = age, y = survival_prob)) +
  geom_line() +
  scale_y_continuous() +
  xlab("Age") +
  ylab("Survival probability")

fig
```



Exercise 6.2

We have the following contractual information:

```
T <- 10      #length of contract
x <- 60      #age of insured
D <- 20000   #disability pension while disabeld
B <- 50000   #death benefit
r <- 0.05    #intensity rate
```

Where our state space is $S = \{*, \diamond, \dagger\}$, we have the following policy-functions:

$$a_{\diamond}(t) = \begin{cases} 0, & t < 0 \\ Dt, & t \in [0, T) \\ DT, & t \geq T \end{cases} \quad a_{*\dagger}(t) = \begin{cases} B, & t \in [0, T) \\ 0, & else \end{cases}$$

Since we for now do not care about the premiums in the policy-functions, we get

$$\dot{a}_*(t) = 0$$

and

$$\dot{a}_{\diamond}(t) = D\mathbb{1}_{[0,T)}(t)$$

We can set up thieles differential equations:

$$\begin{aligned} \frac{d}{dt}V_*^+(t) &= r(t)V_*(t) - \mu_{*\diamond}^x(t)[V_{\diamond}^+(t) - V_*^+(t)] - \mu_{*\dagger}^x(t)[B\mathbb{1}_{[0,T)}(t) - V_*^+(t)] \\ \frac{d}{dt}V_{\diamond}^+(t) &= r(t)V_{\diamond}(t) - D\mathbb{1}_{[0,T)}(t) - \mu_{*\diamond}^x(t)[V_*^+(t) - V_{\diamond}^+(t)] + \mu_{\diamond\dagger}^x(t)V_{\diamond}^+(t) \end{aligned}$$

With finial conditions: $V_*^+(T) = V_{\diamond}^+(T) = 0$

In order to solve this numerically, we partition up $[0, T]$ in times $t_i = t_0 + ih$, with $0 = t_0 < \cdots < t_n = T$.

We also iterate backwards, giving us:

$$\frac{V_*^+(t_i) - V_*^+(t_{i-1})}{h} = rV_*(t_i) - \mu_{*\diamond}^x(t_i)[V_{\diamond}^+(t_i) - V_*^+(t_i)] - \mu_{*\dagger}^x(t_i)[B - V_*^+(t_i)]$$

We then solve for $V_*(t_{i-1})$, giving us:

$$V_*^+(t_{i-1}) = V_*^+(t_i) - h\left[rV_*^+(t_i) - \mu_{*\dagger}^x(t_i)[V_{\diamond}^+(t_i) - V_*^+(t_i)] - \mu_{*\dagger}^x(t)[B - V_*(t_i)]\right]$$

We do the same for $V_{\diamond}^+(t_{i-1})$:

$$V_{\diamond}^+(t_{i-1}) = V_{\diamond}^+(t_i) - h\left[rV_{\diamond}^+(t_i) - D\mathbb{1}_{[0,T)}(t) - \mu_{\diamond *}^x(t_i)[V_*^+(t_i) - V_{\diamond}^+(t_i)] + \mu_{\diamond\dagger}^x(t_i)V_{\diamond}^+(t_i)\right]$$

```
#states 0:alive, 1:disabled, 2:dead

mu01 <- function(u){
  a1 <- 4*10^{-4}
  b1 <- 3.4674*10^{-6}
  c1 <- 0.138755
  return(a1 + b1*exp(c1*u))
}

mu02 <- function(u){
```

```r
  a2 <- 5*10^{-4}
  b2 <- 7.5858*10^{-5}
  c2 <- 0.087498
  return(a2 + b2*exp(c2*u))
}

mu12 <- function(u){
  return(mu02(u))
}

mu10 <- function(u){
  return(0.1*mu01(u))
}
```

```r
#a.e. derivative of a_{\diamond}(t)
dis_dot <- function(t){
  if(t >= 0 && t<T){ return(D) }
  if(t >= T){ return(0) }
}

#PV of policy V0 and V1
h <- 1/12
N <- T/h

PV_act <- PV_dis <- rep(0,N+1)
PV_act[N+1] <- PV_dis[N+1] <- 0 #by construction of contract

for(i in N:1){
  PV_act[i] <- PV_act[i+1] - h*(r*PV_act[i+1]
                                - mu01(x+(i+1)*h)*(PV_dis[i+1]-PV_act[i+1])
                                - mu02(x+(i+1)*h)*(B-PV_act[i+1])
                                )

  PV_dis[i] <- PV_dis[i+1] - h*(r*PV_dis[i+1]- dis_dot((i+1)*h)
                                - mu10(x+(i+1)*h)*(PV_act[i+1]-PV_dis[i+1])
                                + mu12(x+(i+1)*h)*PV_dis[i+1]
                                )
}
```

And from this we can find the one-time premium $\pi_0 = V_*^+(0)$:

```
pi0 <- PV_act[1]
pi0
```

```
## [1] 20608.17
```

We were asked to find the yearly premium π, we then use the method, where we create an artificial policy, where one pays a premium of NOK 1 during $[0,T)$, i.e this is the only contribution of interest, thus:

$$
\begin{aligned}
&a_*^{Prem=1}(t) = \begin{cases} 0, & t < 0 \\ -t, & t \in [0,T) \\ -T, & t \geq T \end{cases} \implies \dot{a}_*^{Prem=1}(t) = -1\mathbb{1}_{[0,T)}(t) \\
&a_\diamond^{Prem=1}(t) = 0, \ \forall t \implies \dot{a}_\diamond^{Prem=1}0 \\
&a_{*\dagger}^{Prem=1}(t) = 0
\end{aligned}
$$

We can now set up the new thieles equations:

$$
\begin{aligned}
\frac{d}{dt}V_*^+(t, A^{Prem=1}) &= rV_*^+(t) - \dot{a}_*(t) - \mu_{*\diamond}^x(t)[V_\diamond^+(t) - V_*^+(t)] + \mu_{*\dagger}^x(t)V_*^+(t) \\
\frac{d}{dt}V_\diamond^+(t, A^{Prem=1}) &= rV_\diamond(t) - \mu_{*\diamond}^x(t)[V_*^+(t) - V_\diamond^+(t)] + \mu_{\diamond\dagger}^x(t)V_\diamond^+(t)
\end{aligned}
$$

And again solve for $V_*^+(t_{i-1})$ and $V_\diamond^+(t_{i-1})$:

$$
\begin{aligned}
V_*^+(t_{i-1}) &= V_*^+(t_i) - h\left[rV_*^+(t_i) - \dot{a}_*(t_i) - \mu_{*\dagger}^x(t_i)[V_\diamond^+(t_i) - V_*^+(t_i)] + \mu_{*\dagger}^x(t)V_*(t_i)\right] \\
V_\diamond^+(t_{i-1}) &= V_\diamond^+(t_i) - h\left[rV_\diamond^+(t_i) - \mu_{\diamond *}^x(t_i)[V_*^+(t_i) - V_\diamond^+(t_i)] + \mu_{\diamond\dagger}^x(t_i)V_\diamond^+(t_i)\right]
\end{aligned}
$$

```
#a.e derivative a_{*}:
a_dot_prem1 <- function(t){
  if(t >= 0 && t < T){ return(-1) }
  if(t >= T){ return(0) }
}

PV_prem1_act <- PV_prem1_dis <- rep(0,N+1)
PV_prem1_act[N+1] <- PV_prem1_dis[N+1] <- 0 #follows from contract

for(i in N:1){
  PV_prem1_act[i] <- PV_prem1_act[i+1] - h*(r*PV_prem1_act[i+1] - a_dot_prem1((i+1)*h)
                                 - mu01(x+(i+1)*h)*(PV_prem1_dis[i+1]-PV_prem1_act[i+1])
                                 + mu02(x+(i+1)*h)*(PV_prem1_act[i+1])
                                 )

  PV_prem1_dis[i] <- PV_prem1_dis[i+1] - h*(r*PV_prem1_dis[i+1]
                                 - mu10(x+(i+1)*h)*(PV_prem1_act[i+1]-PV_prem1_dis[i+1])
                                 + mu12(x+(i+1)*h)*PV_prem1_dis[i+1]
                                 )
}
PV_prem1_act[1]
```

```
## [1] -6.453883
```

We will now use the equvalence principle, which tells us:

$$\pi V_*^+(0, A^{Prem=1}) + V_*^+(0) = 0$$
$$\Downarrow$$
$$\pi = -\frac{V_*(0)}{V_*^+(0, A^{Prem=1})}$$

This gives us:

```
pi0
```

```
## [1] 20608.17
```

```
yearly_premium <- (-1)*pi0/PV_prem1_act[1]
yearly_premium
```

```
## [1] 3193.142
```

Exercise 6.4

We are dealing with an endownment insurance, meaning that $S = \{*, \dagger\}$, we are asked to calculate the second moment of the benefit payments V_t^+ for $t = 10, \ldots, 20$. The contractual information provided is:

```
x <- 40
T <- 20
E <- 115000
B <- 220000
r <- 0.03
```

We start by writing up the policy functions:

$$a_*^{Pre}(n) = \begin{cases} 0 & , n = 0, \ldots, T-1 \\ E & , else \end{cases} \quad a_{*\dagger}^{Post}(n) = \begin{cases} B & , n = 0, \ldots, T-1 \\ 0 & , else \end{cases}$$

From chapter.5 we are given the following formula:

$$\mathbb{E}[(V_t^+)^p | X_t = i] = (v_t)^p \sum_j p_{ij}^x(t, t+1) \sum_{k=0}^{p} \binom{p}{k} (a_{ij}^{Post}(t))^{p-k} \mathbb{E}[(V_{t+1}^+)^k | X_{t+1} = j]$$

Where v_t is the one-step discount factor in $[t, t+1]$ i.e:

$$v(t) = v_0 v_1 \ldots v_{t-1}$$

Let's denote $V_2(t) := \mathbb{E}[(V_t^+)^2 | X_t = *]$, and then translate the equation above into our situation:

$$\begin{aligned} V_2(t) &= (v_t)^2 \sum_j p_{*j}^x(t, t+1) \sum_{k=0}^{2} \binom{2}{k} (a_{*j}^{Post}(t))^{2-k} \mathbb{E}[(V_{t+1}^+)^k | X_{t+1} = j] \\ &= (v_t)^2 p_{**}^x(t, t+1) \mathbb{E}[(V_{t+1}^+)^2 | X_{t+1} = *] + (v_t)^2 p_{*\dagger}^x(t, t+1)(a_{*\dagger}^{Post}(t))^2 \\ &= (v_t)^2 p_{**}^x(t, t+1) V_2(t+1) + (v_t)^2 p_{*\dagger}^x(t, t+1)(a_{*\dagger}^{Post}(t))^2 \\ &= e^{-2r}[p_{**}^x(t, t+1) V_2(t+1) + p_{*\dagger}^x(t, t+1)(a_{*\dagger}^{Post}(t))^2] \end{aligned}$$

This has now become an difference equation, with final condition:

$$V_2(T) = (\Delta a_*^{Pre}(T))^2 = E^2$$

From the exercise it's also given that:

$$\mu_{*\dagger}^{x}(t) = 0.0018 + 0.0004t$$

```r
#0:alive, 1:dead

mu01_x <- function(t){
  return(0.0018 + 0.0004*t)
}

p_surv_x <- function(t,s){
  f <- mu01_x
  integral <- integrate(f, lower = t, upper = s)$value
  return(exp((-1)*integral))
}
```

```r
V2 <- rep(0,T+1)
V2[T+1] <- E^{2}

for(n in (T-1):0){
  V2[n+1] <- exp(-2*r)*(p_surv_x(n,n+1)*V2[n+2] + B^{2}*(1-p_surv_x(n,n+1)))
}
V2
```

```
##  [1]  6167825284  6465436986  6765434135  7068218098  7374219626  7683901471
##  [7]  7997761234  8316334482  8640198138  8969974190  9306333731  9650001385
## [13] 10001760131 10362456580 10733006746 11114402361 11507717773 11914117511
## [19] 12334864556 12771329405 13225000000
```