Exerciseses chapter 4

Andreas Slåttelid

23/2/2022

Contents

Exercise 4.2 + 4.3

In this exercise we are considering a disability insurance, hence $S = \{*, \diamond, \dagger\}$. The contractual information given are: as follows:

```
x <- 30      #age
T <- 35      #length of contract
D <- 20000   #disability pension
r <- 0.03    #intereset rate
premium_yearly <- 2500
```

Furthermore the transition rates are given by:

```
lambda <- function(t){
  #we do not have any t-dependency
  m12 <- 0.0279
  m13 <- 0.0229
  m11 <- -(m12+m13)
  m21 <- 0
  m23 <- 0.0229
  m22 <- -(m21+m23)
  m31 <- m32 <- m33 <- 0
  L <- matrix(c(m11,m12,m13,m21,m22,m23,m31,m32,m33), nrow=3, byrow=TRUE)

  return(L)
}
```

We also have the following policy-functions:

$$a_*(t) = \begin{cases} 0, & t < 0 \\ -\pi t, & t \in [0, T) \\ -\pi T, & t \geq T \end{cases}$$

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

We were asked to calculate $V_j^+(t, A)$ for $j = *, \diamond$ and for $t = 30$. We then get:

$$\begin{aligned} V_*^+(t, A) &= \frac{1}{v(t)} \left[\int_t^T v(s) p_{**}(x+t, x+s) da_*(s) + \int_t^T v(s) p_{*\diamond}(x+t, x+s) da_\diamond(s) \right] \\ &= \frac{1}{v(t)} \left[-\pi \int_t^T v(s) p_{**}(x+t, x+s) ds + D \int_t^T v(s) p_{*\diamond}(x+t, x+s) ds \right] \end{aligned}$$

$$
\begin{aligned}
V_\diamond^+(t,A) &= \frac{1}{v(t)}\left[\int_t^T v(s)p_{\diamond\diamond}(x+t,x+s)da_\diamond(s)\right]\\
&= \frac{1}{v(t)}\left[D\int_t^T v(s)p_{\diamond\diamond}(x+t,x+s)ds\right]
\end{aligned}
$$

We will now use an Euler-scheme to simulate what's going on, and then use a Riemann-sum to compute the integrals:

$$
\int_a^b f(x)dx \approx \sum_{i=0}^{n-1} f(x_i^*)\Delta x
$$

Where $\Delta x = \frac{b-a}{n}$, which in our case corresponds to: h and $x_i^* \in [x_i, x_{i+1}]$

```
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


  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 <- 60       #start age
P0 <- diag(3)  #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 60 to 65.

v <- function(t){
  return(exp(-r*t))
}


#v(s), but s in [30,35]
v_ages <- sapply(seq(30, 35, by = h), v) #we apply the function v to each element
```

```r
#sum1 where we do not care about the premium until afterwards, we use Riemann-sum.
s1 <- 0
for (i in 1:(length(v_ages)-1)){
  s1 <- s1 + v_ages[i]*sol[1,1, ][i]*h #survival i.e * -> *
}

#sum2: do not care about D=20.000 until afterwards
s2 <- 0
for (i in 1:(length(v_ages)-1)){
  s2 <- s2 + v_ages[i]*sol[1,2,][i]*h #disability i.e * -> dis
}

#V*(30, A):
V_30_A <- (1/v(30))*((-1)*premium_yearly*s1 + D*s2)
V_30_A
```

```
## [1] -4781.495
```

```r
#V_[disabeld](30,A):

s3 <- 0
for (i in 1:(length(v_ages)-1)){
  s3 <- s3 + v_ages[i]*sol[2,2,][i]*h #disability i.e dis -> dis
}


#V_[disabeld](30,A):
V_30_dis <- (1/v(30))*D*s3
V_30_dis
```

```
## [1] 88057.1
```

Exercise 4.4 Term insurance in discrete time

In this exercise we are dealing with a term insurance in discrete time, hence: $S = \{*, \dagger\}$, the contractual information provided:

```r
x <- 50
T <- 10
B <- 200000
r <- 0.025
```

We are asked to use the equivalence principle to determine the fair premium π, we start of by describing the policy functions:

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

Furthermore the formula for dircrete reserves are given by:

$$V_i^+(t, A) = \sum_j \sum_{n \geq t} \frac{v(n)}{v(t)} p_{ij}^x(t, n) a_j^{Pre}(n) + \sum_{k \neq j} \sum_{n \geq t} \frac{v(n+1)}{v(t)} p_{ij}^x(t, n) p_{jk}^x(n, n+1) a_{jk}^{Post}(n)$$

In our case, this translates to:

$$V_*^+(t, A) = -\pi \sum_{n=t}^{T-1} \frac{v(n)}{v(t)} p_{**}^x(t, n) + B \sum_{n=t}^{T-1} \frac{v(n+1)}{v(t)} p_{**}^x(t, n) p_{*\dagger}^x(n, n+1)$$

The equivalence principle states that we should choose π so that $V_*^+(0, A) = 0$, giving us:

$$0 = -\pi \sum_{n=0}^{T-1} v(n) p_{**}^x(0, n) + B \sum_{n=0}^{T-1} v(n+1) p_{**}^x(0, n) p_{*\dagger}^x(n, n+1)$$
$$\Downarrow$$
$$\pi = \frac{B \sum_{n=0}^{T-1} v(n+1) p_{**}^x(0, n) p_{*\dagger}^x(n, n+1)}{\sum_{n=0}^{T-1} v(n) p_{**}^x(0, n)}$$

```r
#0: alive, 1:dead
mu01 <- function(u){
  a <- 0.002
  b <- 0.0005
  return(a + b*(u-50))
}

p_surv <- function(t,s){
  f <- mu01
  integral <- integrate(f, lower = t, upper = s)$value
  return(exp((-1)*integral))
}
```

```r
#discount-factor
v <- function(t){
  return(exp(-r*t))
}

s1_above <- 0
for (n in 0:(T-1)){
  s1_above <- s1_above + v(n+1)*p_surv(x, x+n)*(1-p_surv(x+n,x+n+1))
}

s2_below <- 0
for (n in 0:(T-1)){
  s2_below <- s2_below + v(n)*p_surv(x,x+n)
}

above1 <- B*s1_above
below2 <- s2_below

prem <- above1/below2
prem
```

```
## [1] 852.2476
```

Exercise 4.5 Permanent disability insurance discrete time

We have the following state space $S = \{*, \diamond, \dagger\}$. Furthermore the contractual information provided is:

```
x <- 45
T <- 20
D <- 12000
r <- 0.03

#discount factor
v <- function(t){
  return(exp(-r*t))
}
```

Exercise 4.5 i)

In this case we have the following transition rates:

$$\mu_{*\diamond}(t) = 0.0279 \;\; \mu_{*\dagger}(t) = 0.0229 \;\; \mu_{\diamond\dagger}(t) = \mu_{*\dagger}(t)$$

```
#1: alive
#2: disabled
#3: dead
m12 <- 0.0279
m13 <- 0.0229
m11 <- -(m12+m13)
m21 <- 0
m23 <- 0.0229
m22 <- -(m21+m23)
m31 <- m32 <- m33 <- 0
```

Since reactivation is not allowed, we can actually solve the transition-probabilites directly. We first start with the survival probability:

$$p_{11}(t,s) = \exp((-1) * 0.0508(s-t))$$

```
p_11 <- function(t,s){
  return(exp(-1*0.0508*(s-t)))
}
```

We describe the policy functions:

$$a_*^{Pre}(n) = \begin{cases} -\pi & , n = 0, \ldots, T-1 \\ 0 & , else \end{cases} \quad a_\diamond^{Pre}(n) = \begin{cases} D & , n = 0, \ldots, T-1 \\ 0 & , else \end{cases}$$

And we get the following formula for the reserve:

$$V_*^+(t, A) = \frac{1}{v(t)} \left[-\pi \sum_{n=t}^{T-1} v(n) p_{**}^x(t,n) + D \sum_{n=t}^{T-1} v(n) p_{*\diamond}^x(t,n) \right]$$

Now from the equivalence principle we should have that the premium π should be decided so that $V_*^+(0, A) = 0$, this leaves us with:

$$\pi = \frac{D \sum_{n=0}^{T-1} v(n) p_{*\diamond}^x(0, n)}{\sum_{n=0}^{T-1} v(n) p_{**}^x(0, n)}$$

We can use Kolmogorov's equations to find $p_{*\dagger}$ and $p_{*\diamond}$, as reactivation is not allowed, and $\mu_{\diamond\dagger}(t) = \mu_{*\dagger}(t)$

$$\partial_s p_{13}(t, s) = p_{11}(t, s)\mu_{13}(s) + p_{12}(t, s)\mu_{23}(s)$$

Now, lets introduce the notation: $\hat{\mu}(u) = \mu_{23}(u) = \mu_{13}(u)$, we then have:

$$\begin{aligned}\partial_s p_{13}(t, s) &= [p_{11}(t, s) + p_{12}(t, s)]\, \hat{\mu}(s), \quad (p_{11} + p_{12} + p_{13} = 1) \\ &= [1 - p_{13}(t, s)]\, \hat{\mu}(s)\end{aligned}$$

We can now multiply by (-1), and take the derivative of 1 (which is zero), this gives us:

$$\begin{gathered}\partial_s [1 - p_{13}(t, s)] = (-1) * [1 - p_{13}(t, s)]\, \hat{\mu}(s) \\ \Downarrow \\ p_{13}(t, s) = 1 - e^{-\int_t^s \hat{\mu}(u) du} \\ \Downarrow \\ p_{13}(t, s) = 1 - e^{-\int_t^s \hat{\mu}(u) du}\end{gathered}$$

```
#alive to dead
p_13 <- function(t,s){
  return(1-exp((-1)*m13*(s-t)))
}
```

Exercise 4.6 (Endowment policy in discrete time)

Required-libraries

```
list.of.packages <- c("tidyverse", "scales")
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)

library(tidyverse)
library(scales)   #nice formatting of plots
```

We are dealing with an endownment policy, in discrete time, this means that $S = \{*, \dagger\}$, the contractual information provided is:

```
x <- 35     #age
T <- 25     #length of contract
r <- 0.035  #interest rate
E <- 125000 #endownment
B <- 250000 #death-benfit
```

futhermore the force of mortality is given by:

```
mu <- function(t){
  return(0.0015 + 0.0004*(t-35))
}
```

discount-factor:

```
v <- function(t){
  return(exp(-r*t))
}
```

We can also calulate the survival probability:

$$p_{**}(s,t) = \exp\left(-\int_s^t \mu_{*\dagger}(u)du\right)$$

```
p_surv <- function(s,t){
  integral <- 0.0015*(t-s) + 0.0004*((1/2)*(t^2 -s^2) + 35*(s-t))
  return(exp(-integral))
}
p_surv(35,36)
```

```
## [1] 0.9983014
```

We are asked to calculate the yearly premium π, but first we start by defining the policy functions specified in the contract.

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

We can now start to calculate the reserves, but we must be aware of what happens at $n = T$:

$$\begin{aligned} V_*^+(t, A) &= \frac{1}{v(t)} \left[\sum_{n=t}^{T} v(n) p_{**}(x+t, x+n) a_*^{Pre}(n) + \sum_{n=t}^{T-1} v(n+1) p_{**}(x+t, x+n) p_{*\dagger}(x+n, x+n+1) a_{*\dagger}^{Post}(n) \right] \\ &= \frac{1}{v(t)} [\alpha + \beta] \end{aligned}$$

Here α corresponds to the first sum, and β corresponds to the second sum:

$$\alpha = \sum_{n=t}^{T-1} v(n)p_{**}(x+t, x+n)(-\pi) + v(T)p_{**}(x+t, x+T)E$$
$$\beta = \sum_{n=t}^{T-1} v(n+1)p_{**}(x+t, x+n)p_{*\dagger}(x+n, x+n+1)B$$

We will now use the equivalence principle, which states that the fair premium π should be such that $V_*(0, A) = 0$, and using this method we end up with:

$$\pi = \frac{v(T)p_{**}(x, x+T)E + B\sum_{n=0}^{T-1} v(n+1)p_{**}(x, x+n)p_{*\dagger}(x+n, x+n+1)}{\sum_{n=0}^{T-1} v(n)p_{**}(x, x+n)}$$

```
res1 <- 0
for (n in 0:(T-1)){
  res1 <- res1 + v(n+1)*p_surv(x, x+n)*(1-p_surv(x+n, x+n+1))
}

res2 <- 0
for (n in 0:(T-1)){
  res2 <- res2 + v(n)*p_surv(x, x+n)
}

above <- B*res1 + v(T)*p_surv(x,x+T)*E #whats above in the premium expression
below <- res2            #whats below in the premium expression

premium_yearly <- above/below
premium_yearly
```

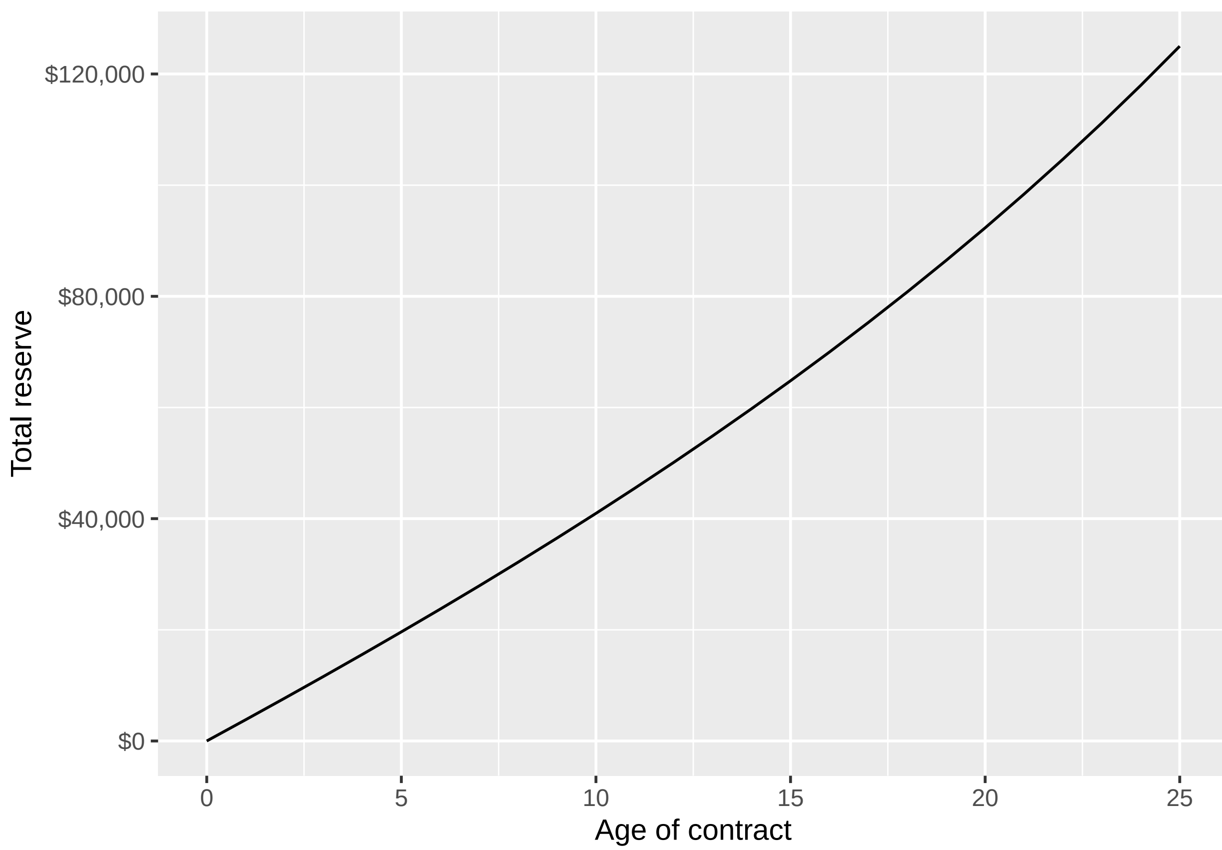
```
## [1] 4095.413
```

We now also want the prospective reserves: i.e $V_*^+(t, A)$ for $t = 0, \ldots, T$:

```
#Present value of reserve when including premiums.
V_reserve_1 <- function(x=35, E = 125000 , T=25){
  #x: age of insured
  #B: benefit specified in contract
  #T: length of contract

  V_star_pre <- NULL
  for (t in 0:(T-1)){
    s <- 0
    for (n in t:(T-1)){
      s <- s + (1/v(t))*(-1)*premium_yearly*v(n)*p_surv(t+x, n+x)
    }
    V_star_pre[t+1] <- s  + (1/v(t))*v(T)*p_surv(x+t, x+T)*E
  }
  V_star_pre[T+1] <- E #specified by contract
  return(V_star_pre)
}
```

```r
V_reserve_2 <- function(x=35, B=250000, T=25){
  #x: age of insured
  #B: benefit specified in contract
  #T: length of contract

  V_star_post <- NULL
  for (t in 0:(T-1)){
    s <- 0
    for (n in t:(T-1)){
      s <- s + (1/v(t))*B*v(n+1)*p_surv(x+t, x+n)*(1-p_surv(x+n, x+n+1))
    }
    V_star_post[t+1] <- s
  }
  V_star_post[T+1] <- 0 #specified by contract
  return(V_star_post)
}

reserve_total <- V_reserve_1() + V_reserve_2()
reserve_total
```

```
##  [1] 1.091394e-11 3.823145e+03 7.692302e+03 1.161236e+04 1.558847e+04
##  [6] 1.962604e+04 2.373075e+04 2.790862e+04 3.216595e+04 3.650944e+04
## [11] 4.094613e+04 4.548348e+04 5.012938e+04 5.489218e+04 5.978071e+04
## [16] 6.480437e+04 6.997309e+04 7.529742e+04 8.078858e+04 8.645848e+04
## [21] 9.231977e+04 9.838591e+04 1.046712e+05 1.111910e+05 1.179615e+05
## [26] 1.250000e+05
```



Exercise 4.7 Discrete vs continous time endowment insurance

In this exercise we are considering an endowment insurance, hence $S = \{*, \dagger\}$, the contractual information provided is:

```
x <- 30
T <- 35
E <- 100000
B <- 200000
r <- 0.035
```

We start with discrete time, the policy functions are as follows:

$$a_*^{Pre}(n) = \begin{cases} -\pi & , n = 0, \dots, T-1 \\ E & , n = T \end{cases}$$

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

Again:

$$V_i^+(t, A) = \sum_j \sum_{n \geq t} \frac{v(n)}{v(t)} p_{ij}^x(t, n) a_j^{Pre}(n) + \sum_{k \neq j} \sum_{n \geq t} \frac{v(n+1)}{v(t)} p_{ij}^x(t, n) p_{jk}^x(n, n+1) a_{jk}^{Post}(n)$$

Which translates to:

$$V_*^+(t, A) = -\pi \sum_{n=t}^{T-1} \frac{v(n)}{v(t)} p_{**}^x(t, n) + \frac{v(T)}{v(t)} p_{**}^x(t, T) E + B \sum_{n=t}^{T-1} \frac{v(n+1)}{v(t)} p_{**}^x(t, n) p_{*\dagger}^x(n, n+1)$$

We then use the equivalence principle, telling us that the fair premium π should be chosen so that $V_*^+(0, A) = 0$:

$$\pi = \frac{v(T) p_{**}^x(0, T) E + B \sum_{n=0}^{T-1} v(n+1) p_{**}^x(0, n) p_{*\dagger}^x(n, n+1)}{\sum_{n=0}^{T-1} v(n) p_{**}^x(0, n)}$$

We now discuss the situation in continous time, we then get the following policy functions:

$$a_*(t) = \begin{cases} 0, & t < 0 \\ -\pi t, & t \in [0, T) \\ -\pi T + E, & t \geq T \end{cases} \quad a_{*\dagger}(t) = \begin{cases} B, & t \in [0, T) \\ 0, & else \end{cases}$$

We start off by stating the general formula for reserves in continous-time:

$$V_i^+(t, A) = \sum_j \frac{v(T)}{v(t)} p_{ij}^x(t, T) \Delta a_j(T) + \sum_j \int_t^T \frac{v(s)}{v(t)} p_{ij}^x(t, s) da_j(s) + \sum_{k \neq j} \int_t^T \frac{v(s)}{v(t)} p_{ij}^x(t, s) \mu_{jk}^x(s) a_{jk}(s) ds$$

Translating this into our situation:

$$V_*^+(t, A) = \frac{v(T)}{v(t)} p_{**}^x(t, T) E - \pi \int_t^T \frac{v(s)}{v(t)} p_{**}^x(t, s) ds + B \int_t^T \frac{v(s)}{v(t)} p_{**}^x(t, s) \mu_{*\dagger}^x(s) ds$$

Now as always, the equivalence principle tells us to chose π such that $V_*^+(0, A) = 0$:

$$\pi = \frac{v(T) p_{**}^x(0, T) E + B \int_0^T v(s) p_{**}^x(0, s) \mu_{*\dagger}^x(s) ds}{\int_0^T v(s) p_{**}^x(0, s) ds}$$

```r
#want the map function from purrr:
library(tidyverse)
packageVersion("tidyverse")
```

```
## [1] '1.3.1'
```

```r
#0: alive, 1:dead
mu01 <- function(t){
  a2 <- 5*10^{-4}
  b2 <- 7.5858*10^{-5}
  c2 <- 0.087498

  ans <- a2 + b2*exp(c2*t)
  return(ans)
}
```

```r
p_surv <- function(t, s){
  f <- Vectorize(mu01)
  integral <- integrate(f, lower = t, upper = s)$value

  ans <- exp(-integral)
  return(ans)
}
```

```r
#discount factor
v <- function(t){
  return(exp(-r*t))
}
```

```r
#premium discrete time:
summand_upper <- function(n){
  v(n+1)*p_surv(x, x + n)*(1- p_surv(x + n, x + n +1))
}

summand_lower <- function(n){
  v(n)*p_surv(x, x +n)
}

#map_dbl: maps the function summand_upper to the elements 0:(T-1)
sum_upper <- sum(map_dbl(0:(T-1), summand_upper))
sum_lower <- sum(map_dbl(0:(T-1), summand_lower))

above_discrete <- v(T)*p_surv(x, x + T)*E + B*sum_upper
below_discrete <- sum_lower

premium_discrete <- above_discrete/below_discrete
premium_discrete
```

```
## [1] 2204.58
```

```r
#premium continous time
integrand_upper <- function(s){
  v(s)*p_surv(x, x + s)*mu01(x + s)
}

integrand_lower <- function(s){
  v(s)*p_surv(x, x + s)
}


f1 <- Vectorize(integrand_upper)
f2 <- Vectorize(integrand_lower)

integral_upper <- integrate(f1, lower = 0, upper = T)$value
integral_lower <- integrate(f2, lower = 0, upper = T)$value

above_cont <- v(T)*p_surv(x, x + T)*E + B*integral_upper
below_cont <- integral_lower

premium_cont <- above_cont/below_cont
premium_cont
```

```
## [1] 2268.052
```

We then recall what the premiums were in the different cases:

We then recall what the premiums were:

```r
premium_discrete
```

```
## [1] 2204.58
```

```r
premium_cont
```

```
## [1] 2268.052
```