Excercises chapter 4 vectorized

Andreas Slåttelid

Contents

Exercise 4.4 Term insurance in discrete time 2
Exercise 4.6 (Endowment policy in discrete time) 4

Exercise 4.4 Term insurance in discrete time

Required libraries:

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

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

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

```r
upper_summand <- function(n){
  v(n+1)*p_surv(x, x + n)*(1 - p_surv(x + n, x + n + 1))
}

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

premium_yearly <- (B*sum(map_dbl(0:(T-1), upper_summand)))/(sum(map_dbl(0:(T-1),  lower_summand)))

premium_yearly
```

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

Exercise 4.6 (Endowment policy in discrete time)

Required-libraries

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

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

Contractual information provided:

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

```
#0: alive, 1: dead
mu01 <- function(t){
  return(0.0015 + 0.0004*(t-35))
}

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

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

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

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

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:

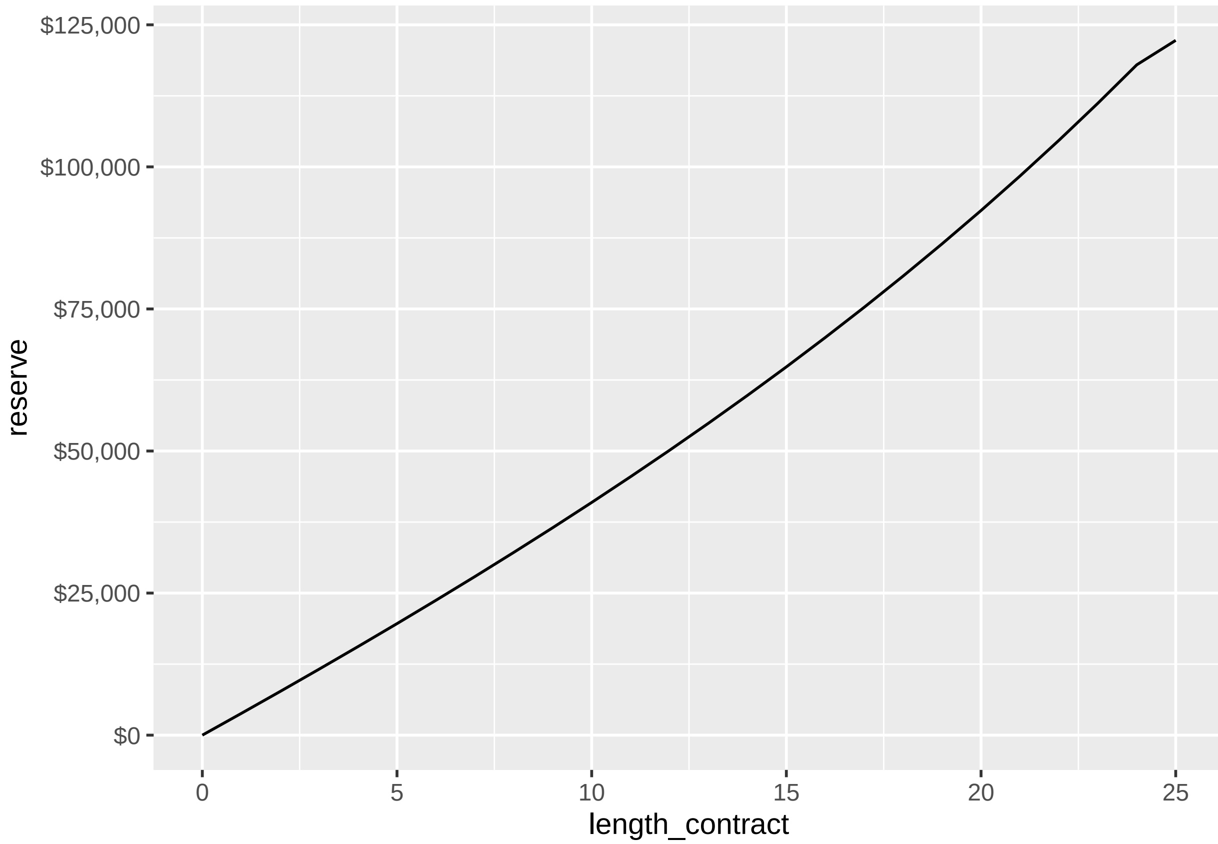
$$\pi = \frac{v(T) p_{**}(x, x+T) E + B \sum_{n=0}^{T-1} v(n+1) p_{**}(x, x+n) p_{*\dagger}(x+n, x+n+1)}{\sum_{n=0}^{T-1} v(n) p_{**}(x, x+n)}$$

```
upper_summand <- function(n){
  v(n+1)*p_surv(x, x + n)*(1-p_surv(x + n, x + n +1))
}

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

upper_sum <- sum(map_dbl(0:(T-1), upper_summand))
```

```r
lower_sum <- sum(map_dbl(0:(T-1), lower_summand))

upper_expression <- v(T)*p_surv(x, x + T)*E + B*upper_sum

yearly_premium <- upper_expression/lower_sum
yearly_premium
```

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

```r
V_star <- function(t){

  #inner summands
  summand_1 <- function(t, n){
    (v(n)/v(t))*p_surv(x + t, x +n)
  }

  summand_2 <- function(t, n){
    (v(n+1)/v(t))*p_surv(t + x, n + x)*(1 - p_surv(x + n, x + n +1))
  }

  ans <- (v(T)/v(t))*p_surv(x + t, x + T)*E -
         yearly_premium*sum(map_dbl(t:(T-1),summand_1, t = t)) + B*sum(
                        map_dbl(t:(T-1), summand_2, t = t))

  return(ans)
}

length_contract <- 0:T
reserve <- map_dbl(length_contract, V_star)

df <- data.frame(length_contract, reserve)
colnames(df) <- c("length_contract", "reserve")

df %>%
  ggplot(aes(x = length_contract, y = reserve)) +
  geom_line() +
  scale_y_continuous(labels = dollar)
```



$125,000
$100,000
$75,000
$50,000
$25,000
$0
reserve
0
5
10
15
20
25
length_contract