Life Insurance Mathematics - Week 1 Assignment 1

09/09/2021

Group 3: Vu The Doan (12918687), Shagun Saini (13011448), Ngoc Nguyen (13009842), Aljer Lee Zhen Yee (12563412)

Question 2

Creating the vectors for the cash flow of the loan and the (decreasing) payments:

```
cash_flow_loan <- c(10000, 20000, 30000)

cash_flow_vector <- c(1)

for (i in 1:19) {
  cash_flow_vector[i+1] <- cash_flow_vector[i]*(1-0.05)
}
```

Creating the vector for the discount factors:

```
d_factors <- c(1.06^-3, rep(NULL, 19))

for (i in 1:2) {
  d_factors[i+1] <- d_factors[i]*1.06^-1
}

d_factors[4] <- 1.06^-5*1.07^-1

for (i in 4:7) {
  d_factors[i+1] <- d_factors[i]*1.07^-1
}

d_factors[9] <- 1.06^-5*1.07^-5*1.08^-1

for (i in 9:19) {
  d_factors[i+1] <- d_factors[i]*1.08^-1
}
```

Creating the discount function and calculate the present value of the loan and the payment cash flows:

```
discount <- function(s, t, i) {(1 + i) ^ - (t - s)}

PV_loan <- sum(cash_flow_loan * discount(0, 0:2, i = 0.06))

PV_payments <- sum(cash_flow_vector * d_factors)
```

Calculating the yearly payments and showing the result:

```
k1 <- PV_loan/PV_payments

yearly_payments <- cash_flow_vector*k1; yearly_payments
```

```
##  [1] 8177.153 7768.296 7379.881 7010.887 6660.343 6327.325 6010.959 5710.411
##  [9] 5424.891 5153.646 4895.964 4651.166 4418.607 4197.677 3987.793 3788.403
## [17] 3598.983 3419.034 3248.082 3085.678
```

Question 3

Converting the yearly interest rate to the monthly interest rate, using the EAR (effective annual rate) formula and calculating the monthly payments:

```
i <- 5.5/100

i_monthly <- (1+i)^(1/12) - 1

discount_factors <- (1 + i_monthly) ^ - (1:180)

cash_flow <- rep(1, 180)

PV <- sum(cash_flow * discount_factors)

k2 <- 150000/PV; k2
```

```
## [1] 1214.988
```

It can be seen that this result is slightly different from the Bloomberg Mortgage Calculator's. This is because Bloomberg uses another formula to convert the yearly interest rate to the monthly interest rate.

```
i <- 5.5/100

i_monthly <- i/12

discount_factors <- (1 + i_monthly) ^ - (1:180)

cash_flow <- rep(1, 180)

PV <- sum(cash_flow * discount_factors)

k3 <- 150000/PV; k3
```

```
## [1] 1225.625
```

The result is now similar to Bloomberg's.

Bonus Question

```r
p <- rev(cash_flow*discount_factors*k3)

max <- 12
y <- seq_along(p)
chunks <- split(p, ceiling(y/max))

a <- c(150000)

for (i in 1:15) {
  a[i+1] <- a[i] - sum(chunks[[i]])
}

round(a, 2)
```

```
##  [1] 150000.00 143377.20 136380.83 128989.80 121181.86 112933.49 104219.85
##  [8]  95014.70  85290.30  75017.36  64164.95  52700.38  40589.12  27794.69
## [15]  14278.55      0.00
```

It can be seen that this result is similar to Bloomberg's.

To calibrate the result for a mortgage with the first payment date in September 2017 and final payment date in August 2032, the following code can be used:

```r
b <- c(150000)

for (i in 1:180) {
  b[i+1] <- b[i] - p[i]
}

round(b[c(seq(5, 180, by=12), 181)], 2)
```

```
##  [1] 147832.66 141087.60 133962.08 126434.61 118482.54 110081.91 101207.42
##  [8]  91832.34  81928.43  71465.86  60413.11  48736.91  36402.08  23371.47
## [15]   9605.82      0.00
```

It can be seen that this result is similar to Bloomberg's.