MIS 381N Project 1

Siqi Chen (sc56839), Danielle Diehl (dsd752), Zhiyi Yang (zy3296), Boying You (by3475)

02/14/2018

```r
library(lpSolve)
```

Question 1

Let the amount purchased for bond1, bond2, ..., bond10 be $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}$. Then the ojective is to minimize the total cost at $t = 0$, which is

$$minimize \quad 102x_1 + 99x_2 + 101x_3 + 98x_4 + 98x_5 + 104x_6 + 100x_7 + 101x_8 + 102x_9 + 94x_{10}$$

subject to the constraints of matching liability schedule, which are

$$\begin{cases} 105x_1 + 3.5x_2 + 5x_3 + 3.5x_4 + 4x_5 + 9x_6 + 6x_7 + 8x_8 + 9x_9 + 7x_{10} & = 12000 \\ 103.5x_2 + 105x_3 + 3.5x_4 + 4x_5 + 9x_6 + 6x_7 + 8x_8 + 9x_9 + 7x_{10} & = 18000 \\ 103.5x_4 + 4x_5 + 9x_6 + 6x_7 + 8x_8 + 9x_9 + 7x_{10} & = 20000 \\ 104x_5 + 9x_6 + 6x_7 + 8x_8 + 9x_9 + 7x_{10} & = 20000 \\ 109x_6 + 106x_7 + 8x_8 + 9x_9 + 7x_{10} & = 16000 \\ 108x_8 + 9x_9 + 7x_{10} & = 15000 \\ 109x_9 + 7x_{10} & = 12000 \\ 107x_{10} & = 10000 \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} & \geq 0 \end{cases}$$

We could rewrite the objective function and contraints in matrix forms $\mathbf{c}^T\mathbf{x}$ and $A\mathbf{x} = \mathbf{b}$ respectively, where

$$\mathbf{c} = \begin{bmatrix} 102 \\ 99 \\ 101 \\ 98 \\ 98 \\ 104 \\ 100 \\ 101 \\ 102 \\ 94 \end{bmatrix}, \quad A = \begin{bmatrix} 105 & 3.5 & 5 & 3.5 & 4 & 9 & 6 & 8 & 9 & 7 \\ 0 & 103.5 & 105 & 3.5 & 4 & 9 & 6 & 8 & 9 & 7 \\ 0 & 0 & 0 & 103.5 & 4 & 9 & 6 & 8 & 9 & 7 \\ 0 & 0 & 0 & 0 & 104 & 9 & 6 & 8 & 9 & 7 \\ 0 & 0 & 0 & 0 & 0 & 109 & 106 & 8 & 9 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 108 & 9 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 109 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 107 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 12000 \\ 18000 \\ 20000 \\ 20000 \\ 16000 \\ 15000 \\ 12000 \\ 10000 \end{bmatrix}$$

Question 2

We could use R to solve for $\mathbf{x}$ in above linear problem,

```r
c <- c(102,99,101,98,98,104,100,101,102,94)
b <- c(12000,18000,20000,20000,16000,15000,12000,10000)
dir <- rep("=", 8)
coupon <- c(5,3.5,5,3.5,4,9,6,8,9,7)
```

```r
maturity <- c(1,2,2,3,4,5,5,6,7,8)
A = matrix(0,8,10)
for (i in 1:10){
  for (j in 1:8){
    if (maturity[i] > j){
      A[j,i] = coupon[i]
    } else if(maturity[i] == j){
      A[j,i] = coupon[i]+100}
  }
}
s <- lp('min',c,A,dir,b)
print(s$solution)
```

```
##  [1]  62.13613   0.00000 125.24293 151.50508 156.80776 123.08007   0.00000
##  [8] 124.15727 104.08986  93.45794
```

Hence, the solution to the linear problem is

$$\mathbf{x} = \begin{bmatrix} 62.14 \\ 0 \\ 125.24 \\ 151.51 \\ 156.81 \\ 123.08 \\ 0 \\ 124.16 \\ 104.09 \\ 93.46 \end{bmatrix}$$

which matches the rounded up optimal solution.

Question 3

To write a function that can construct a portfolio for any set of liabilities and bonds, we need to tranform the inputs to the vectors/matrix $\mathbf{c}, \mathbf{A}, \mathbf{b}$ required for solving the linear problem.

We can easily see that the price vector $\mathbf{P}$ is the vector $\mathbf{c}$ in the objective function, the libaility vector $\mathbf{L}$ is the vector $\mathbf{b}$ on the RHS of the constraints.

Now, we need to construct $\mathbf{A}$, which consists of the cashflow from the bonds we purchased on each liability date. Since we're given the coupon payments and maturity for the bonds, we can use nested for loops to build $\mathbf{A}$. For each bond,

- if the current date is before maturity, then the cashflow will be the coupon payment;
- if the current date is on maturity, then the cashflow will be the coupon plus face value ($100);
- if the current date is after maturity, then the cashflow will be 0 (bond expired).

Hence, we could write the function in R as:

```r
dedicate_g11 <- function(P,C,M,L){
  n = length(P) #number of bonds
  t = length(L) #length of years
  dir=rep("=",t)
  A = matrix(0,t,n)
  for (i in 1:n){
    for (j in 1:t){
```

```
      if (M[i] > j){
        A[j,i] = C[i]
      } else if(M[i] == j){
        A[j,i] = C[i]+100}
    }
  }
  s = lp('min',P,A,dir,L,compute.sens = 1)
  return(s)
}
```

Now, we can use the function to solve the previous problem

```
result = dedicate_g11(c,coupon,maturity,b)
print(result$solution)
```

```
##  [1]  62.13613   0.00000 125.24293 151.50508 156.80776 123.08007   0.00000
##  [8] 124.15727 104.08986  93.45794
```

The result matches our solution in previous question.

Question 4

Assumptions

To solve this problem, following assumptions are made:

- portfolio construction date is 01/03/2017
- all prices are aksed price
- for t-bonds with same maturity date, we consider all of them in our potential portfolio
- coupon payments are paid semi-annually with the amount of half of the annual coupon payments

T-Bond Information

First, we constructed an array with all liability dates.

```
year = c(2017:2022)
dates = c()
for (y in year){
  date1 = paste(c("6/30/",y), collapse = "")
  date2 = paste(c("12/31/",y), collapse = "")
  dates = c(dates,date1,date2)
}
```

Then, we fetched the bond information on 01/03/2017 from the Wall Street Journal (WSJ) Online U.S. Treasury Quotes. We used the information to construct the price vector **P**, the coupon vector **C**, the maturity vector **M**, and the liability vector **L**. (note that maturity are in unit of half years)

```
library(XML)
tbond = "http://www.wsj.com/mdc/public/page/2_3020-treasury-20170103.html#treasuryB?
mod=mdc_pastcalendar"
tbond.table = readHTMLTable(tbond, header=T, which = 3, stringsAsFactor = F)
tbond.table[,2:6] = lapply(lapply(tbond.table[,2:6],as.character),as.double)
```

```
## Warning in lapply(lapply(tbond.table[, 2:6], as.character), as.double): NAs
## introduced by coercion
```

```r
P = c()
C = c()
M = c()
for (d in dates){
  row = which(tbond.table$Maturity == d)
  P = c(P,tbond.table[row,]$Asked)
  C = c(C,tbond.table[row,]$Coupon/2)
  M = c(M, rep(which(dates == d),length(row)))
}

L = c(9000000,9000000,10000000,10000000,6000000,6000000,
      9000000,9000000,10000000,10000000,5000000,3000000)
```

Constructing Portfolio

Now we can use the function we wrote in Question 3 to contruct a dedicated portfolio.

```r
library(pander)
portfolio = dedicate_g11(P,C,M,L)
result = data.frame(cbind(P, C, dates[M]),portfolio$solution)
colnames(result) = c('price','coupon (half-year)', 'maturity','quantity')
pander(result)
```

price	coupon (half-year)	maturity	quantity
99.9922	0.3125	6/30/2017	0
100.0391	0.375	6/30/2017	0
100.8906	1.25	6/30/2017	80800
99.8203	0.375	12/31/2017	0
100.0625	0.5	12/31/2017	0
101.7891	1.375	12/31/2017	81810
99.3516	0.3125	6/30/2018	0
100.4453	0.6875	6/30/2018	0
101.9063	1.1875	6/30/2018	92935
100.0625	0.625	12/31/2018	0
100.2891	0.6875	12/31/2018	0
100.5156	0.75	12/31/2018	94038
99.1563	0.5	6/30/2019	0
100.6797	0.8125	6/30/2019	54744
98.9609	0.5625	12/31/2019	0
100.4141	0.8125	12/31/2019	55189
99.9609	0.8125	6/30/2020	85637
100.8906	0.9375	6/30/2020	0
99.8906	0.875	12/31/2020	86333
102.375	1.1875	12/31/2020	0
96.7266	0.5625	6/30/2021	0
100.9688	1.0625	6/30/2021	97088
100.2656	1	12/31/2021	0
100.7813	1.0625	12/31/2021	98120
100.2891	1.0625	6/30/2022	49162
99.8516	1.0625	12/31/2022	29685

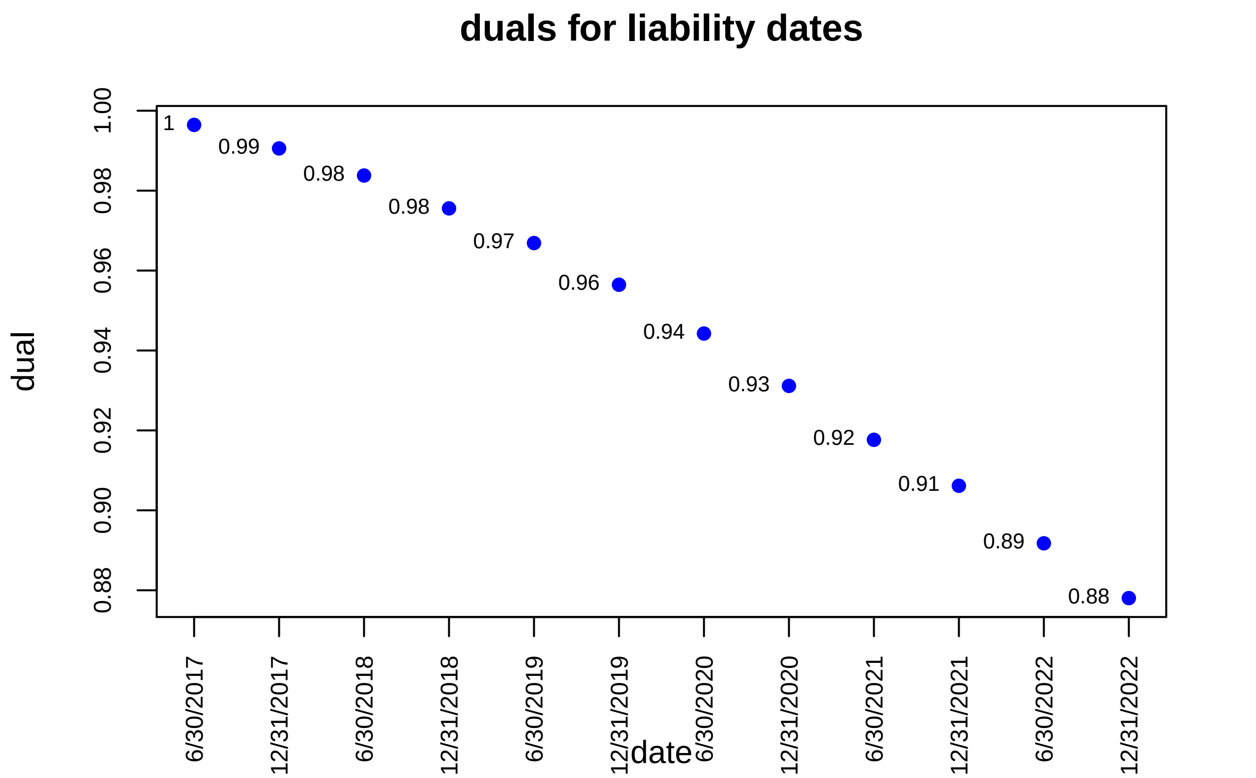
```r
print(portfolio$objval)
```

```
## [1] 91225778
```

Also, the total cost for construncting the portfolio would be **$91,225,778**.

Sensitivity Analysis

```r
duals = portfolio$duals[1:12]
pos = 1:12
plot(duals, xlab = 'date', ylab = 'dual',
     main = 'duals for liability dates',
     xaxt = 'n',cex.axis = 0.75,
     col = 'blue', type="p",pch = 16)
axis(1, at=pos,labels=dates, las=2, cex.axis = 0.75)
text(pos, duals, labels = round(duals,2), pos = 2, cex = 0.7)
```



The plot above shows the dual at each liability date. The values can be interpreted as the present value of $1 increase in liability at $t = 0$ (01/03/2017). For example, if the liability on 12/31/2022 increases by $1, the total cost will increase by $0.88. These values also shows the implied yield rate for the portfolio we constructed.