20962 Coursework (R)

Scenario 1: 1,000 people aged 60 exact take out a whole of life assurance policy which pays a sum assured of £20,000 at the end of the year of death, each person pays the insurance company a lump sum of £9,128 for this policy. Hence, the insurance company will have £9,128,000 at time zero and we want to see if the insurance company is expected to run out of money at some stage in the future. Assume an interest rate of 4 %.p.a. and mortality of AM92 (ultimate) throughout this question.

We begin the calculations by loading AM92 in R:

```
> source("Cont1code.R")
> useLifeTable("AM92")
> getLives(age = "all")
```

variables:

```
> x <- 60
> n <- 1000
> S <- 20000
> i <- 0.04
> price <- 9128
> v <- 1/1.04
```

(a) Calculate $20{,}000A_{60}$

```
> a <- S*getEPVWholeLifeAss(x, i)
> a
[1] 9127.996
```

$\therefore 20{,}000A_{60} \approx$ £9128.00

(b) Assuming that deaths actually occur in line with the mortality assumption and starting with 1,000 lives, generate a vector in R showing the expected number of deaths each year until there are expected to be no further deaths.

```
> qx <- c()
> lives <- c()
> deaths <- c()

> for (k in 1:60){
    qx[k] <- (getLives(59+k)- getLives(60+k))/getLives(59+k)
```

#formula used : $q_x = 1 - \frac{l_{x+1}}{l_x} = \frac{l_x - l_{x+1}}{l_x}$

```
      for (j in 2:60){
        lives[1] <- 1000      ## start with 1000 lives aged 60
        lives[j] <- lives[j-1]*(1-qx[j-1]) ##
```

formula used: $l_{x+1} = l_x * (1\text{-}q_x)$

```
          for (w in 1:59){
          deaths[w] <- lives[w]*qx[w]
```

formula used: $d_x = l_x * q_x$

```
        }
      }
    }
```

```
> deaths[is.na(deaths)] <- 0 ##set NaNs as 0
> options(scipen=3) ##show decimals
```

∴ The expected number of deaths each year is:

```
> deaths
 [1]  8.02200538797  8.93672510958  9.94051349983 11.03885121057 12.23357948244 13.52841309911
 [7] 14.92461185679 16.42259568755 18.02032953598 19.71295726457 21.49229558170 23.34714629886
[13] 25.26186425461 27.21679878160 29.18773379718 31.14404656276 33.05032280717 34.86645363405
[19] 36.54596656109 38.04124775213 39.29934269648 40.26856744718 40.89763645434 41.13945272127
[25] 40.95434881866 40.31438311268 39.20420116409 37.62634410026 35.60245457401 33.17413816265
[31] 30.40350174246 27.37113996827 24.17154832313 20.90812700348 17.68666658828 14.60739086472
[37] 11.75768877314  9.20621382312  6.99822177073  5.15379398288  3.66894648864  2.51891406342
[43]  1.66370625325  1.05440635582  0.63949193646  0.37010013033  0.20379626343  0.10643662831
[49]  0.05255611358  0.02445296741  0.01068134904  0.00437160052  0.00165819330  0.00059221189
[55]  0.00019381480  0.00005383744  0.00002153498  0.00000000000  0.00000000000
```

```
> sum(deaths)  ##check if the total of deaths is 1000
[1] 1000
```

(d)

```
> moneypaid <- c()
> moneyleft <- c()
```

```
> for (w in 1:59){
    moneypaid[w] <- S*v^w*deaths[w]
```

The money paid every year according to the number of deaths is calculated as such:
$20000 * 1.04^w *$ number of deaths at year w

```
    moneyleft[w] <- price*n - sum(moneypaid[1:w])
```

#e.g. money left in year 3 = 2198000 (initial) - money paid in y1 - money paid in y2 - money paid in y3

```
  }
> moneyleft[moneyleft<50] = 0     ##regard numbers below £50 as 0
```

∴ The sum of money(£) held by the insurance company at the end of each year is:

```
> moneyleft
 [1] 8973730.66562 8808480.57113 8631738.96504 8443017.83946 8241915.62768
 [6] 8028081.60003 7801252.03193 7561255.43574 7308038.54568 7041691.19389
[11] 6762471.48618 6470820.31229 6167387.89152 5853047.42891 5528909.17906
[16] 5196347.95739 4857004.92747 4512783.68759 4165858.36444 3818627.78661
[21] 3473710.34429 3133879.56535 2802014.60375 2481026.92476 2173773.63741
[26] 1882954.35899 1611021.02734 1360070.26566 1131750.71715  927186.59536
[31]  746917.99615  590870.78061  458365.16959  348157.55838  258516.05960
[36]  187328.77437  132232.98600   90752.46907   60433.28819   38963.71169
[41]   24267.53171   14565.93770    8404.62778    4649.96376    2460.36368
[46]    1241.89147     596.74361     272.76166     118.93924      50.12242
[51]       0.00000       0.00000       0.00000       0.00000       0.00000
[56]       0.00000       0.00000       0.00000       0.00000
```

Note that moneyleft[1] is the same as my answer to part (c), as said in the hint given.

Q2.
new variable introduced:

```
> pricePonzi <- 3000
```

(a)

```
> moneyleftPonzi <- c()

> for (w in 1:59){
    moneyleftPonzi[w] <- pricePonzi*n - sum(moneypaid[1:w])
  }
```

∴ The sum of money(£) held by Ponzi Inc at the end of each year is:

```
> moneyleftPonzi
 [1]  2845730.67  2680480.57  2503738.97  2315017.84  2113915.63  1900081.60
 [7]  1673252.03  1433255.44  1180038.55   913691.19   634471.49   342820.31
[13]    39387.89  -274952.57  -599090.82  -931652.04 -1270995.07 -1615216.31
[19] -1962141.64 -2309372.21 -2654289.66 -2994120.43 -3325985.40 -3646973.08
[25] -3954226.36 -4245045.64 -4516978.97 -4767929.73 -4996249.28 -5200813.40
[31] -5381082.00 -5537129.22 -5669634.83 -5779842.44 -5869483.94 -5940671.23
[37] -5995767.01 -6037247.53 -6067566.71 -6089036.29 -6103732.47 -6113434.06
[43] -6119595.37 -6123350.04 -6125539.64 -6126758.11 -6127403.26 -6127727.24
[49] -6127881.06 -6127949.88 -6127978.78 -6127990.16 -6127994.30 -6127995.73
[55] -6127996.18 -6127996.30 -6127996.34 -6127996.34 -6127996.34
```

From the vector computed above, the amount of money left for Ponzi turns negative at year 14 (see moneyleftPonzi[14]), which is the year Ponzi Inc goes bankrupt.

(b)ii)

```
> livesPonzi <- c()
> deathsPonzi <- c()
> moneypaidPonzi <- c()
> moneyleftPonzinew <- c()

> for (r in 1:50){
    livesPonzi[r] <- sum(lives[1:r]) ##e.g. in year 2, there are 2 batches of ages: 60 with
    lives[1] and 61 with lives[2]
    deathsPonzi[r] <- livesPonzi[r] - livesPonzi[r+1]
    moneypaidPonzi[r] <- S*(sum(v^seq(1:r)*deaths[1:r]))
    moneyleftPonzinew[r] <- r*pricePonzi*n - sum(moneypaidPonzi[1:r]) ## r*pricePonzi*n
    because Ponzi Inc is receiving £3000000 every year
  }
```

∴ the expected amount of money(£) that Ponzi Inc has remaining for the next 50 years is:

```
> moneyleftPonzi
 [1]    2845731    5526211    8029950   10344968   12458884   14358965
 [7]   16032217   17465473   18645511   19559202   20193674   20536494
[13]   20575882   20300930   19701839   18770187   17499192   15883975
[19]   13921834   11612461    8958172    5964051    2638066   -1008907
[25]   -4963133   -9208179  -13725158  -18493088  -23489337  -28690150
[31]  -34071232  -39608362  -45277997  -51057839  -56927323  -62867994
[37]  -68863761  -74901009  -80968575  -87057612  -93161344  -99274778
[43] -105394374 -111517724 -117643263 -123770021 -129897425 -136025152
[49] -142153033 -148280983
```

See that moneyleftPonzinew[1] matches moneyleftPonzi[1] and moneyleftPonzinew[2] matches my answer to part (b)i), as said in the given hint.

(b)iii)
As moneyleftPonzi[24] is the first term in the vector that turns negative, Ponzi Inc goes bankrupt at the end of year 24.