Valuing cash balance plans embedded options in `StocVal`

Nathan Esau

June 21, 2015

The default plan consists of the following five employees.

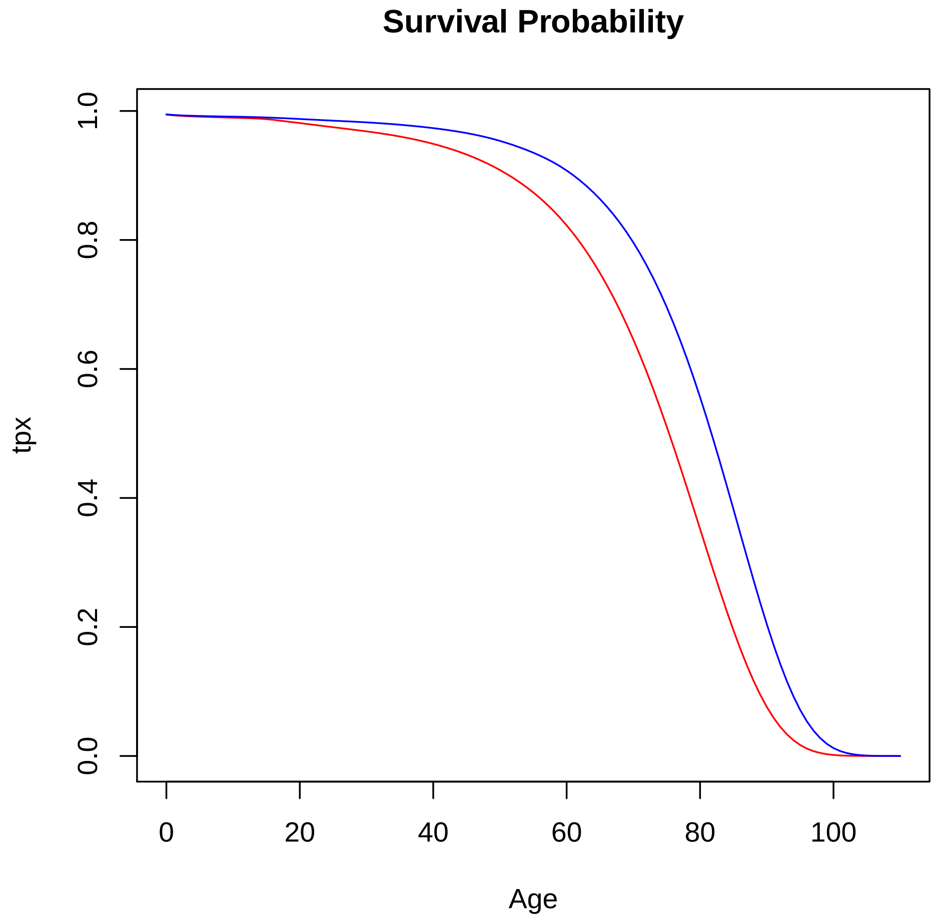
```
> library(StocVal)
> print(demoInfo, row.names=F)
```

```
 employee_id age_entry age_valuation gender current_salary account_value
           1        25            45      M         100000        120000
           2        30            40      M          80000         50000
           3        30            30      F          65000             0
           4        25            50      F         150000        200000
           5        25            60      M         100000         20000
 remaining
        20
        25
        30
        15
         5
```

To demonstrate the various functions, employee 5 will be used. A summary of the embedded option costs for all of the employees will be shown at the end.

The survival probabilities used (mortality) are plotted below. The female mortality is shown in blue. Note that these probabilities are actually discrete (the line joining the points may be deceiving.

```
> tpxm <- cumprod(1 - mortalityInfo$qxm)
> tpxf <- cumprod(1 - mortalityInfo$qxf)
> plot(x=seq(0, 110, 1), y=tpxm, xlab="Age", ylab="tpx",
+   main="Survival Probability", type='l', col='red')
> lines(x=seq(0,110,1), y=tpxf, xlab="Age", ylab="tpx", col='blue', type='l')
```



For employee 5, the relevant decrements are shown below (note that the termination rate refers to the probability of employee 5 retiring early in that year).

```
> employee <- demoInfo[5,]
> surviveInfo.out <- surviveInfo(employee)
> print(surviveInfo.out, row.names=F)
```

```
 Age mortRate termRate probDecr   survBoy
  60  0.01416      0.1 0.112744 1.0000000
  61  0.01549      0.1 0.113941 0.8872560
  62  0.01694      0.1 0.115246 0.7861612
  63  0.01853      0.1 0.116677 0.6955592
  64  0.02026      0.1 0.118234 0.6144035
  65  0.02216      1.0 1.000000 0.5417601
```

First, the deterministic scenario will be shown. Here we will be using the interest rate yields with term length 1 and 10 (these are the output of `determScenario()`).

First, these short rate yields are shown.

```
> print(term1)
```

```
  [1] 0.00303900 0.00982243 0.01866025 0.03011234 0.04046266 0.04509149
  [7] 0.04521804 0.04395812 0.05019802 0.05638154 0.05717695 0.05606745
 [13] 0.05448846 0.05269593 0.05094373 0.04937927 0.04778350 0.04616153
 [19] 0.04460994 0.04322488 0.04210228 0.04133800 0.04102814 0.04126929
 [25] 0.04215893 0.04379584 0.04628052 0.04971578 0.05420726 0.05986410
 [31] 0.05183728 0.05200166 0.05210760 0.05219895 0.05227760 0.05234524
 [37] 0.05240334 0.05245319 0.05249592 0.05253252 0.05256383 0.05259059
 [43] 0.05261346 0.05263297 0.05264962 0.05266381 0.05267590 0.05268618
 [49] 0.05269494 0.05270238 0.05270871 0.05271409 0.05271865 0.05272252
 [55] 0.05272581 0.05272859 0.05273095 0.05273296 0.05273465 0.05273608
 [61] 0.05273730 0.05273833 0.05273919 0.05273993 0.05274055 0.05274107
 [67] 0.05274152 0.05274189 0.05274221 0.05274247 0.05274270 0.05274289
 [73] 0.05274305 0.05274319 0.05274330 0.05274339 0.05274348 0.05274354
 [79] 0.05274360 0.05274365 0.05274369 0.05274372 0.05274375 0.05274378
 [85] 0.05274380 0.05274382 0.05274383 0.05274384 0.05274385 0.05274386
 [91] 0.05274387 0.05274387 0.05274388 0.05274388 0.05274389 0.05274389
 [97] 0.05274389 0.05274390 0.05274390 0.05274390
```

```
> print(term10)
```

```
  [1] 0.03415100 0.03960161 0.04426713 0.04788314 0.05015811 0.05121122
  [7] 0.05164171 0.05189955 0.05212135 0.05156018 0.05024312 0.04873584
 [13] 0.04726386 0.04591931 0.04477842 0.04390179 0.04334503 0.04319527
 [19] 0.04354915 0.04450396 0.04615809 0.04713129 0.04819868 0.04930895
 [25] 0.05040519 0.05142063 0.05227866 0.05289283 0.05316708 0.05299599
 [31] 0.05226531 0.05233797 0.05239687 0.05244745 0.05249086 0.05252806
 [37] 0.05255992 0.05258718 0.05261048 0.05263038 0.05264737 0.05266186
 [43] 0.05267421 0.05268472 0.05269368 0.05270130 0.05270778 0.05271328
 [49] 0.05271796 0.05272193 0.05272530 0.05272816 0.05273058 0.05273264
 [55] 0.05273438 0.05273585 0.05273710 0.05273816 0.05273905 0.05273981
 [61] 0.05274045 0.05274099 0.05274144 0.05274183 0.05274215 0.05274243
 [67] 0.05274266 0.05274286 0.05274302 0.05274316 0.05274328 0.05274338
 [73] 0.05274346 0.05274353 0.05274359 0.05274364 0.05274368 0.05274372
 [79] 0.05274375 0.05274377 0.05274379 0.05274381 0.05274383 0.05274384
 [85] 0.05274385 0.05274386 0.05274387 0.05274387 0.05274388 0.05274388
 [91] 0.05274389         NA         NA         NA         NA         NA
 [97]         NA         NA         NA         NA
```

The embedded options that we will be valuing is a floor on the minimum return obtained by the employee in a given year. The value of this floor and some other plan assumptions, such as the volatility of a long-term treasury bond, a vesting period and a fixed contribution rate are shown below.

```
> planVariablesDF <- data.frame(variable=c("salary_scale", "contrib_rate",
+   "vesting", "floor", "vol"), value=c(planVariables[[1]], planVariables[[2]],
+   planVariables[[3]], planVariables[[4]], planVariables[[5]]))
> print(planVariablesDF, row.names=F)
```

```
    variable value
salary_scale 0.040
contrib_rate 0.050
     vesting 3.000
       floor 0.040
         vol 0.019
```

The account value with and without a floor for employee 5 are shown below.

```
> accountValueNF.out <- accountValueNF(employee, surviveInfo.out)
> print(accountValueNF.out, row.names=F)

 Age   av_boy av_growth   salary contrib_eoy   av_eoy   ben_boy   dis_ben
  60 20000.00  20683.02 100000.0    5000.000 25683.02  2254.880  2254.880
  61 25683.02  26700.11 104000.0    5200.000 31900.11  2596.421  2588.554
  62 31900.11  33312.24 108160.0    5408.000 38720.24  2890.211  2853.427
  63 38720.24  40574.28 112486.4    5624.320 46198.60  3142.370  3045.546
  64 46198.60  48515.84 116985.9    5849.293 54365.13  3356.023  3157.534
  65 54365.13  57149.23      0.0       0.000 57149.23 29452.857 26633.249

> accountValueWF.out <- accountValueWF(employee, surviveInfo.out)
> print(accountValueWF.out, row.names=F)

 Age   av_boy av_growth option_adjust av_growth_adjust   salary contrib_eoy
  60 20000.00  20683.02     116.98000         20800.00 100000.0    5000.000
  61 25800.00  26821.72      10.27851         26832.00 104000.0    5200.000
  62 32032.00  33449.96       0.00000         33449.96 108160.0    5408.000
  63 38857.96  40718.61       0.00000         40718.61 112486.4    5624.320
  64 46342.93  48667.40       0.00000         48667.40 116985.9    5849.293
  65 54516.69  57308.56       0.00000         57308.56      0.0       0.000
   av_eoy   ben_boy   dis_ben
 25800.00  2254.880  2254.880
 32032.00  2608.247  2600.344
 38857.96  2902.161  2865.225
 46342.93  3153.548  3056.379
 54516.69  3366.507  3167.399
 57308.56 29534.968 26707.499
```

We can then calculate the value of our embedded option which provides a floor on the minimum return obtained by employee 5 under the deterministic scenario.

```
> floorOption.cost <- floorOption(accountValueNF.out, accountValueWF.out)
> print(floorOption.cost, row.names=F)

[1] 118.5354
```

There is an alternative way of valuing this option which uses an adapted version of the black scholes formula. I have referred to it as `floorOptionPBO()` since the price of the option equals to the sum of the PBO column, not the difference between the benefit with and without the option.

```
> floorOptionPBO.cost <- floorOptionPBO(demoInfo[5,], accountValueNF.out,
+   accountValueWF.out, surviveInfo.out)
> print(floorOptionPBO.cost, row.names=F)

[1] 717.9956
```

Under the deterministic scenario, the cost of this option for each member of the pension plan are shown below. Totals are shown in the bottom row.

```
> planSummaryDeterm.out <- planSummaryDeterm(demoInfo,planVariables)
> print(planSummaryDeterm.out, row.names=F)

 employee_id pv_benefit_no floor_option floor_option_pbo
           1     197860.29   768.506627       11668.6016
           2     118410.16   324.411306        7622.1005
           3      60133.70     1.229513        3344.8421
           4     304389.46  1294.489400       13715.4442
           5      40533.19   118.535355         717.9956
                 721326.80  2507.172202       37068.9840
```

For the stochastic scenario, we use different yields than under the deterministic scenario. For instance, the first stochastic scenario and the results for employee 5 are shown below. [1]

```
> termStruct.out <- cubic_spline(termStruct);
> termStruct.extension <- curve_extensionNS(termStruct.out);
> credit_spread <- rep(0.01,100);
> hw.out <- hull_white(srinput, termStruct.extension, termofyield=1);
> cholesky.out <- cholesky(cholinput);
> esga.out <- esga(esgin, credit_spread, volterm.equity, inflation_mean,
+ volterm.inflation , cholesky.out, termStruct.extension);
> stochasticScenarios.out <- stochasticScenarios(hw.out, termStruct.extension,
+ srinput, esga.out)
> term1 <- as.numeric(stochasticScenarios.out$term1[1,])
> term10 <- as.numeric(stochasticScenarios.out$term10[1,])
> print(term1)

  [1] 0.003034392 0.012904303 0.032837902 0.043229341 0.048336114 0.031789603
  [7] 0.040162987 0.017866165 0.011808591 0.014972061 0.001000000 0.001000000
 [13] 0.001000000 0.001000000 0.004041130 0.002109245 0.001000000 0.001000000
```

[1] To match the results of the SOA excel sheets replace `stochasticScenarios.out` with `stochasticScenarios.out <- bmarkScenarios(file)`. For more information see the `benchmarkDemo`.

```
 [19] 0.001000000 0.001000000 0.001000000 0.002556337 0.014548016 0.017506026
 [25] 0.010859063 0.020678428 0.026044168 0.036260806 0.014828352 0.018397722
 [31] 0.029569083 0.030541169 0.032647860 0.025009182 0.038636552 0.042422924
 [37] 0.046529013 0.033957077 0.046531519 0.044352689 0.043094674 0.032680243
 [43] 0.018334434 0.040573951 0.027970239 0.033516746 0.041548395 0.047635259
 [49] 0.046702242 0.044671738 0.047436926 0.046477762 0.059350242 0.041049203
 [55] 0.042524462 0.055806549 0.046904705 0.038223741 0.033379629 0.046029007
 [61] 0.054076613 0.069179159 0.052317219 0.046217451 0.062442793 0.063082968
 [67] 0.080994382 0.063720931 0.054327092 0.057388825 0.052865428 0.060774544
 [73] 0.073174217 0.078791431 0.083449166 0.100826882 0.119393206 0.129704239
 [79] 0.110082862 0.097155605 0.100343439 0.108863256 0.088604350 0.100329556
 [85] 0.117183906 0.128904652 0.142753932 0.150833648 0.158445229 0.159783677
 [91] 0.160682669 0.138291943 0.145856084 0.165067808 0.159741565 0.155056960
 [97] 0.138819558 0.001000000 0.001000000 0.001000000 0.000000000
```

```
> print(term10)
```

```
  [1]  0.033580800  0.030737329  0.047407259  0.053207160  0.057344151
  [6]  0.044972769  0.047698735  0.024914368  0.018269304  0.020256197
 [11]  0.008782393  0.008889364  0.009167603  0.009561749  0.012735972
 [16]  0.012289294  0.012925144  0.014893857  0.017317107  0.018845640
 [21]  0.020121820  0.022282043  0.032376076  0.034531706  0.028142276
 [26]  0.034126868  0.035414332  0.039354915  0.027316874  0.030166985
 [31]  0.039190394  0.039971478  0.041670882  0.035494433  0.046507626
 [36]  0.049567346  0.052886083  0.042724917  0.052889255  0.051129229
 [41]  0.050113698  0.041697453  0.030103697  0.048080916  0.037895356
 [46]  0.042380124  0.048873550  0.053795028  0.053042487  0.051402824
 [51]  0.053639323  0.052865465  0.063271272  0.048480316  0.049673923
 [56]  0.060410590  0.053216531  0.046200936  0.042286498  0.052511500
 [61]  0.059016959  0.071224674  0.057596328  0.052666699  0.065781789
 [66]  0.066299778  0.080777590  0.066816414  0.059224074  0.061699196
 [71]  0.058043436  0.064436505  0.074459140  0.078999678  0.082764674
 [76]  0.096810857  0.111817741  0.120152085  0.104292878  0.093844327
 [81]  0.096421143  0.103307642  0.086933086  0.096410390  0.110033430
 [86]  0.119507110  0.130701219  0.137231941 -0.338704990 -0.796795387
 [91] -1.233431849           NA           NA           NA           NA
 [96]           NA           NA           NA           NA           NA
[101]  0.000000000
```

```
> accountValueNF.out <- accountValueNF(employee, surviveInfo.out,
+   oneyear=term1, tenyear=term10)
> accountValueWF.out <- accountValueWF(employee, surviveInfo.out,
+   oneyear=term1, tenyear=term10)
> floorOption.cost <- floorOption(accountValueNF.out, accountValueWF.out,
+   oneyear=term1, tenyear=term10)
```

```
> floorOptionPBO.cost <- floorOptionPBO(employee, accountValueNF.out,
+   accountValueWF.out, surviveInfo.out, oneyear=term1, tenyear=term10)
> print(floorOption.cost)
```

```
[1] 307.7618
```

```
> print(floorOptionPBO.cost)
```

```
[1] 654.4354
```

. We can calculate the option costs for each scenario, plots these costs, and calculate the mean, expected value of these option costs (for employee 5). First, we use the first method of valuing the option.

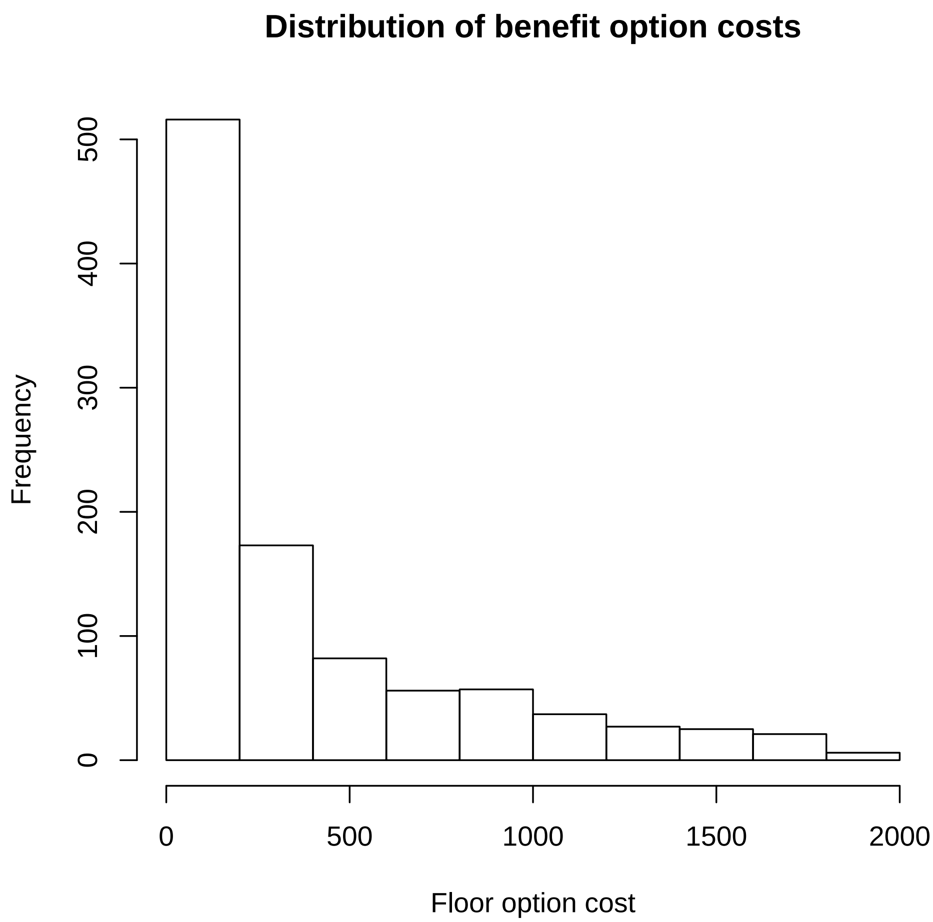
```
> stocfloorOption.cost <- stocfloorOption(employee, surviveInfo.out,
+   stochasticScenarios.out)
> print(stocfloorOption.cost$mean_cost)
```

```
[1] 403.0803
```

```
> print(stocfloorOption.cost$var_cost)
```

```
[1] 179414.5
```

```
> hist(stocfloorOption.cost$costs, main="Distribution of benefit option costs",
+   ylab="Frequency", xlab="Floor option cost")
```



Next, we show the other method of valuing this option.

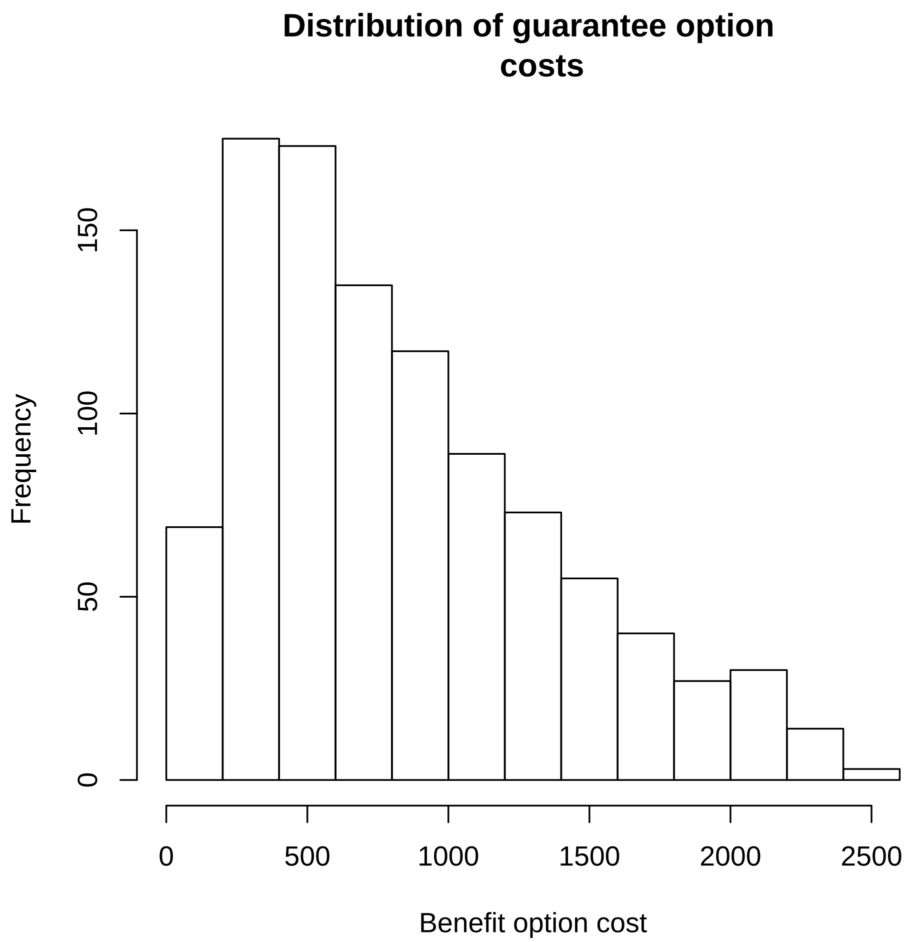
```
> stocfloorOptionPBO.cost <- stocfloorOptionPBO(employee, surviveInfo.out,
+   stochasticScenarios.out)
> print(stocfloorOptionPBO.cost$mean_cost)

[1] 844.6975

> print(stocfloorOptionPBO.cost$var_cost)

[1] 297110.3

> hist(stocfloorOptionPBO.cost$costs, main="Distribution of guarantee option
+   costs", ylab="Frequency", xlab="Benefit option cost")
```

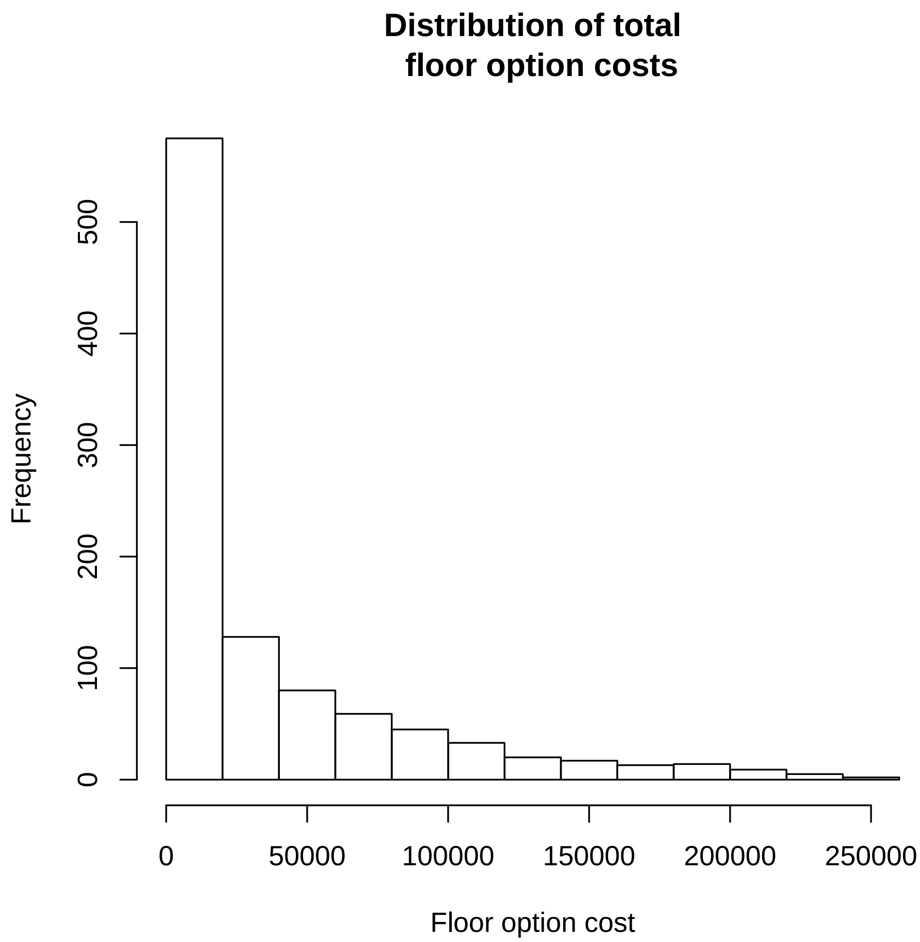


Finally, we show the cost of these options for the entire plan.

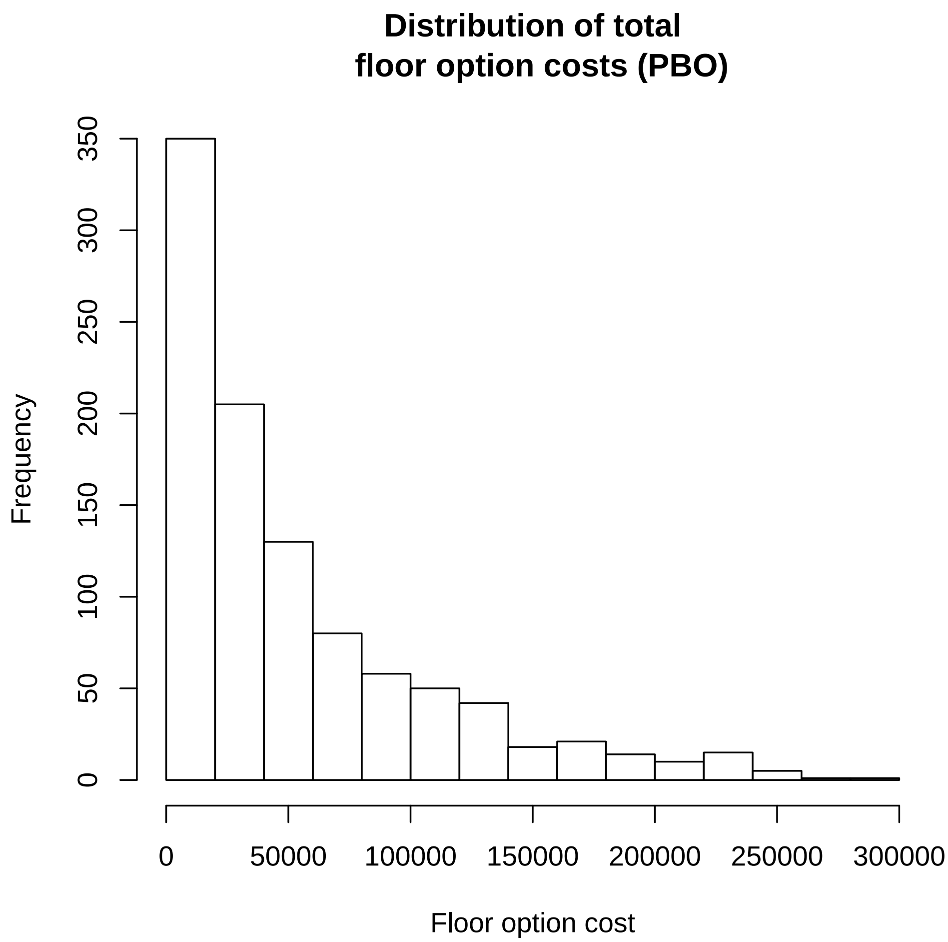
```
> planSummaryStoc.out <- planSummaryStoc(demoInfo, stochasticScenarios.out,
+   planVariables)
> print(planSummaryStoc.out$planSummary, row.names=F)
```

```
 employee_id pv_benefit_no mean_floor_option mean_floor_option_pbo
           1      215957.5        11555.7436            16784.5587
           2      136582.9         7692.9729            10581.8823
           3       75036.3         3682.4548             4813.9392
           4      315204.5        13314.5207            20627.7261
           5       39409.0          403.0803              844.6975
                  782190.2        36648.7724            53652.8037
 var_floor_option var_floor_option_pbo
      244284081.4          315355759.8
      120882182.4          154583024.5
       36798290.2           46788999.4
      304812046.0          407955937.5
         179414.5             297110.3
      706956014.7          924980831.4
```

```
> hist(planSummaryStoc.out$floor_option_costs, main="Distribution of total
+   floor option costs", ylab="Frequency", xlab="Floor option cost")
```



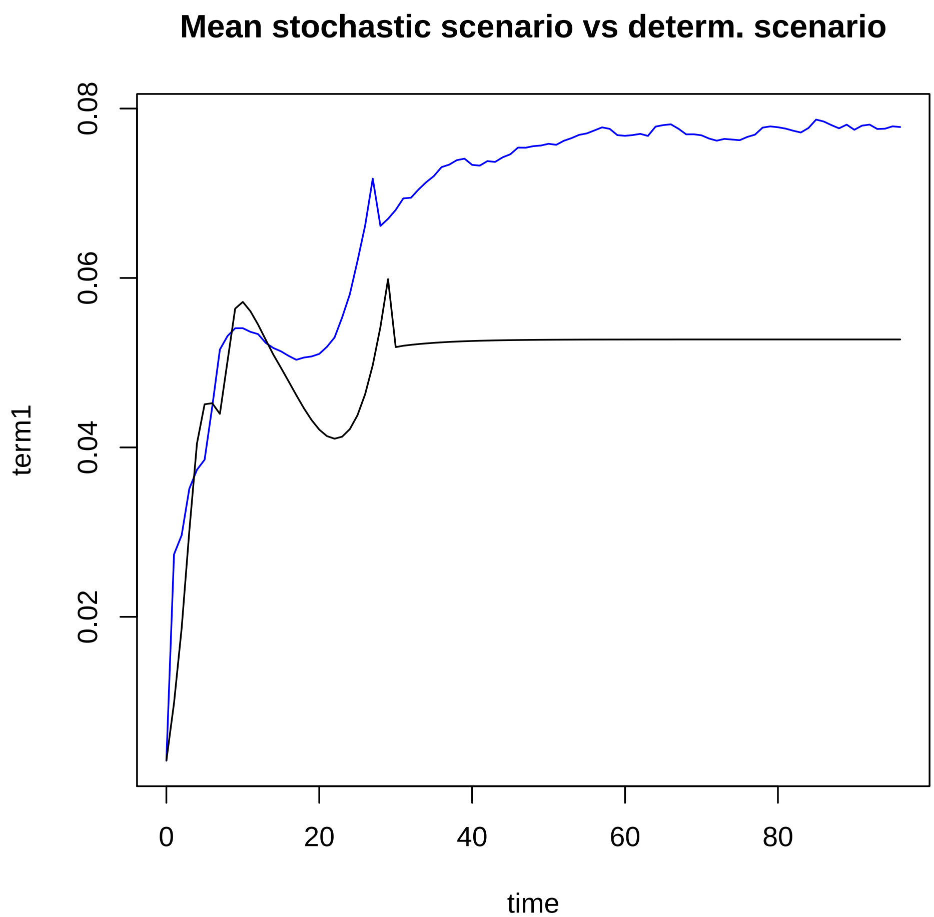

```
> hist(planSummaryStoc.out$floor_option_costs_pbo, main="Distribution of total
+   floor option costs (PBO)", ylab="Frequency", xlab="Floor option cost")
```



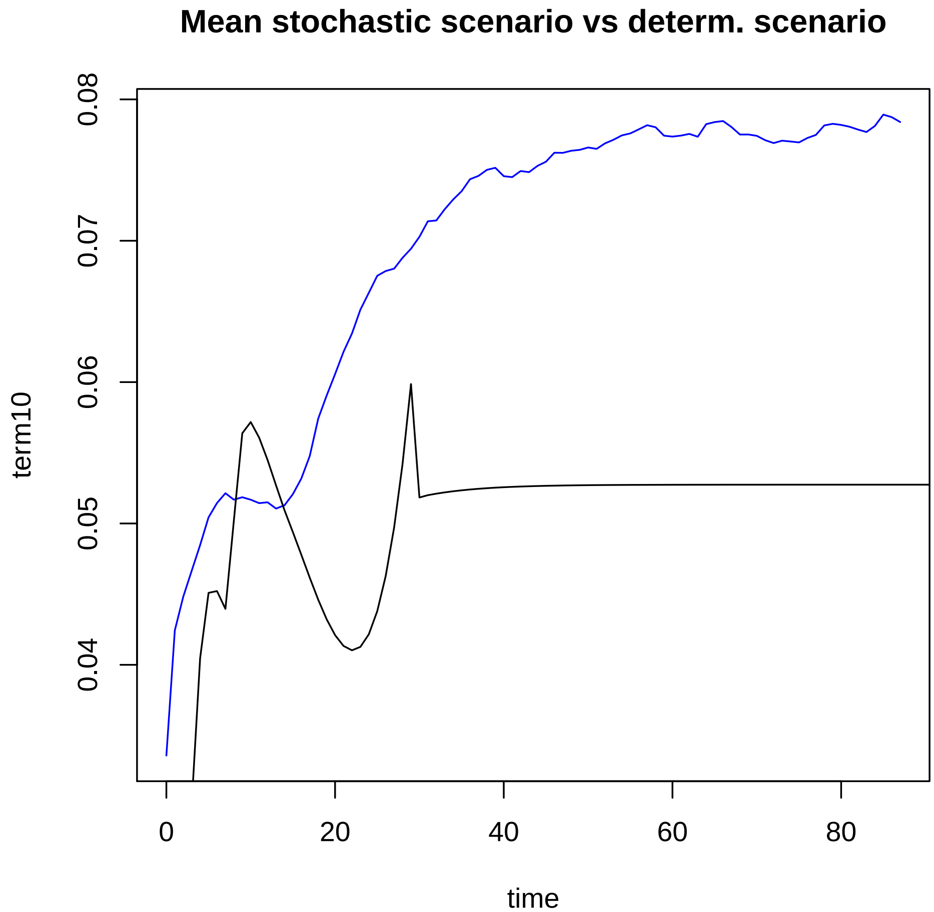
Note that the total cost of the option under the deterministic scenario (2,507) was much less that the total cost of the option under the stochastic scenario ($> 30,000$). This difference is refered to as the time value of financial options and guarantees (TVFOG) and represents the additional option cost caused by market volatility around the baseline projection.

Finally, we can compare the deterministic scenario yield curve to the mean stochastic scenario yield curve. Notice that the determinstic scenario yields are much lower than most of the stochastic scenario yields.

```
> determScenario.out <- determScenario(termStruct.out)
> deterterm1 <- determScenario.out$term1
> deterterm10 <- determScenario.out$term10
> stochasticterm1.mean <- meanScenarios(stochasticScenarios.out$term1)
> stochasticterm10.mean <- meanScenarios(stochasticScenarios.out$term10)
> plot(x=seq(0,96,1), y=stochasticterm1.mean[1:97],
+   main="Mean stochastic scenario vs determ. scenario", type='l', col='blue',
+   xlab="time", ylab="term1")
> lines(x=seq(0,96,1), y=deterterm1[1:97], type='l')
```




```
> plot(x=seq(0,87,1), y=stochasticterm10.mean[1:88],
+   main="Mean stochastic scenario vs determ. scenario", type='l', col='blue',
+   xlab="time", ylab="term10")
> lines(x=seq(0,96,1), y=deterterm1[1:97], type='l')
```


Mean stochastic scenario vs determ. scenario
term10
0.04
0.05
0.06
0.07
0.08
0
20
40
60
80
time