Asssignment 4

Hyunkyun Chung(12527742), Jagathiswarra Rao Raja Rao(12542660)
Younjoo Mo(12475440), Thomas Rigny(12914347)

11/10/2020

Question 1

```r
i <- 0.05
A <- 0.00022
B <- 2.7*10^-6
c <- 1.124
s <- exp(-A)
g <- exp(-B/log(c))

x <- 0:100
age=c(0:100)
t=1
Sv=(s^t)*(g^((c^t-1)*c^age))
Sv=data.frame(age,Sv)
lt=matrix(nrow = 101, ncol = 3)
k=1
lt[,2]=1-Sv[,2]
lt[1,1]=10^6
while(k<100){
  lt[k+1,1]=(1-lt[k,2])*lt[k,1]
  k=k+1
}
j=1
while(j<=100){
  lt[j,3]=sum(lt[j:100,1])/lt[j,1]
  j=j+1
}

lt=data.frame(lt)
colnames(lt)=c("lx","qx","ex")
lt[1:101,]
```

```
##              lx           qx        ex
## 1   1000000.00 0.0002228393 85.882171
## 2    999777.16 0.0002231944 84.901091
## 3    999554.02 0.0002235935 83.919821
## 4    999330.52 0.0002240421 82.938366
## 5    999106.63 0.0002245463 81.956727
## 6    998882.28 0.0002251130 80.974910
```

```
## 7    998657.42 0.0002257500 79.992917
## 8    998431.98 0.0002264661 79.010754
## 9    998205.87 0.0002272708 78.028425
## 10   997979.00 0.0002281754 77.045935
## 11   997751.29 0.0002291922 76.063291
## 12   997522.61 0.0002303350 75.080499
## 13   997292.85 0.0002316195 74.097566
## 14   997061.85 0.0002330633 73.114501
## 15   996829.48 0.0002346862 72.131312
## 16   996595.53 0.0002365102 71.148010
## 17   996359.83 0.0002385605 70.164604
## 18   996122.14 0.0002408650 69.181108
## 19   995882.21 0.0002434552 68.197534
## 20   995639.75 0.0002463666 67.213898
## 21   995394.46 0.0002496390 66.230215
## 22   995145.97 0.0002533172 65.246503
## 23   994893.88 0.0002574515 64.262782
## 24   994637.75 0.0002620983 63.279073
## 25   994377.05 0.0002673214 62.295401
## 26   994111.24 0.0002731921 61.311791
## 27   993839.65 0.0002797907 60.328272
## 28   993561.58 0.0002872076 59.344876
## 29   993276.23 0.0002955440 58.361638
## 30   992982.67 0.0003049140 57.378596
## 31   992679.90 0.0003154459 56.395792
## 32   992366.76 0.0003272835 55.413272
## 33   992041.97 0.0003405889 54.431086
## 34   991704.09 0.0003555439 53.449290
## 35   991351.50 0.0003723530 52.467945
## 36   990982.37 0.0003912462 51.487116
## 37   990594.65 0.0004124817 50.506877
## 38   990186.05 0.0004363498 49.527306
## 39   989753.98 0.0004631769 48.548490
## 40   989295.55 0.0004933298 47.570524
## 41   988807.50 0.0005272204 46.593510
## 42   988286.18 0.0005653122 45.617560
## 43   987727.49 0.0006081256 44.642797
## 44   987126.83 0.0006562457 43.669354
## 45   986479.03 0.0007103299 42.697374
## 46   985778.30 0.0007711170 41.727014
## 47   985018.15 0.0008394373 40.758443
## 48   984191.29 0.0009162238 39.791846
## 49   983289.55 0.0010025248 38.827420
## 50   982303.78 0.0010995182 37.865381
## 51   981223.72 0.0012085275 36.905960
## 52   980037.88 0.0013310397 35.949406
## 53   978733.42 0.0014687256 34.995987
## 54   977295.92 0.0016234618 34.045991
## 55   975709.32 0.0017973567 33.099728
## 56   973955.62 0.0019927785 32.157526
## 57   972014.75 0.0022123868 31.219740
## 58   969864.27 0.0024591689 30.286746
## 59   967479.21 0.0027364792 29.358945
## 60   964831.73 0.0030480838 28.436761
```

```
## 61   961890.84 0.0033982113 27.520647
## 62   958622.13 0.0037916077 26.611077
## 63   954987.41 0.0042336000 25.708553
## 64   950944.38 0.0047301652 24.813604
## 65   946446.25 0.0052880089 23.926782
## 66   941441.44 0.0059146520 23.048663
## 67   935873.14 0.0066185277 22.179849
## 68   929679.04 0.0074090890 21.320963
## 69   922790.96 0.0082969290 20.472647
## 70   915134.63 0.0092939131 19.635561
## 71   906629.45 0.0104133270 18.810383
## 72   897188.42 0.0116700384 17.997800
## 73   886718.20 0.0130806770 17.198508
## 74   875119.32 0.0146638316 16.413203
## 75   862286.72 0.0164402661 15.642584
## 76   848110.50 0.0184331558 14.887335
## 77   832477.14 0.0206683441 14.148130
## 78   815271.22 0.0231746205 13.425615
## 79   796377.62 0.0259840198 12.720406
## 80   775684.53 0.0291321407 12.033074
## 81   753087.18 0.0326584844 11.364135
## 82   728492.49 0.0366068080 10.714039
## 83   701824.71 0.0410254900 10.083151
## 84   673032.00 0.0459679011  9.471734
## 85   642094.13 0.0514927715  8.879925
## 86   609030.93 0.0576645431  8.307713
## 87   573911.44 0.0645536903  7.754895
## 88   536863.34 0.0722369904  7.221040
## 89   498081.94 0.0807977153  6.705419
## 90   457838.06 0.0903257142  6.206925
## 91   416483.51 0.1009173439  5.723944
## 92   374453.10 0.1126751990  5.254182
## 93   332261.52 0.1257075806  4.794391
## 94   290493.73 0.1401276327  4.339956
## 95   249787.53 0.1560520640  3.884247
## 96   210807.67 0.1735993612  3.417565
## 97   174211.60 0.1928873917  2.925415
## 98   140608.38 0.2140302860  2.385559
## 99   110513.92 0.2371344912  1.762866
## 100   84307.26 0.2622938963  1.000000
## 101         NA 0.2895839526        NA
```

Question 2

```r
#2.1
n=500000
px=1-lt$qx
df=(1+i)^-c(c(0:20))
q=c(lt$qx[51:69],1)
p=c(1,cumprod(px[51:69]))
P_sing=sum(n*p*q*df[-1])
P_sing
```

```
## [1] 194219.3
```

```r
#2.2
P_ann=P_sing/sum((df[-21]*p))
P_ann
```

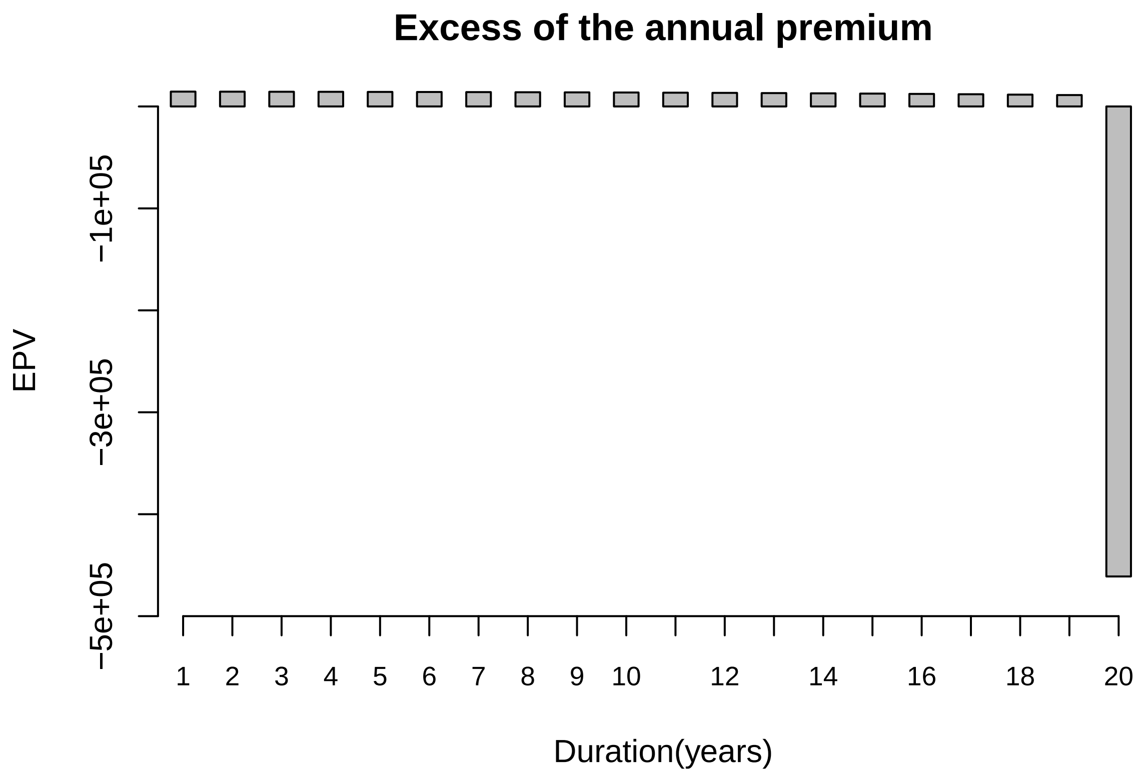
```
## [1] 15122.82
```

(2.3) As with figure 7.1 in the DHW book, we see the EPV drops dramatically at t=20.

```r
#2.3
ben=n*q/(1+i)
pre=rep(P_ann, 20)
excess_term<- c(pre-ben)
excess_term
```

```
##  [1]   14547.33   14489.00   14423.43   14349.75   14266.94   14173.88
##  [7]   14069.31   13951.79   13819.74   13671.36   13504.63   13317.30
## [13]   13106.82   12870.36   12604.72   12306.32   11971.14   11594.69
## [19]   11171.91 -461067.65
```

```r
barplot(height=c(pre-ben),
        main="Excess of the annual premium",
        xlim = c(1,20),
        ylim = c(-500000,1000),
        width= 0.5,
        space = 1,
        xlab = "Duration(years)",
        ylab = "EPV",
        names.arg = sprintf("%d", c(1:20)),
        axes = TRUE,
        axisnames = TRUE,
        axis.lty = 1,
        cex.names = 0.8)
```

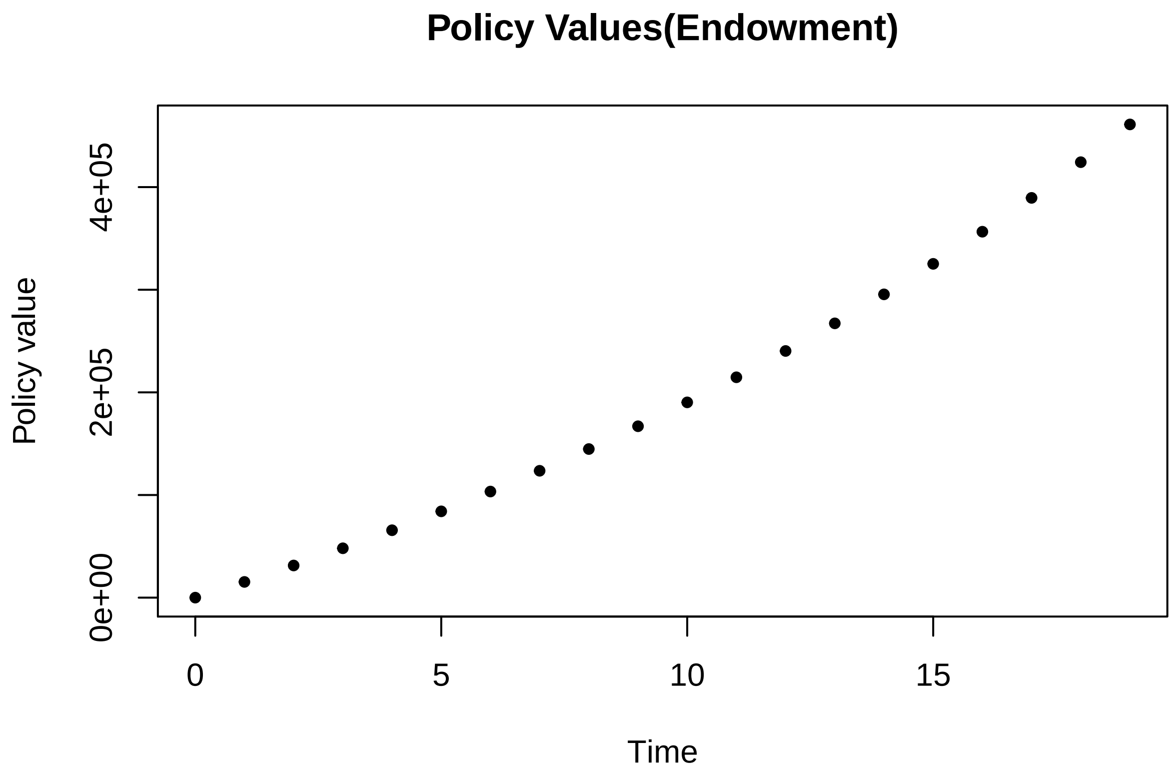


(2.4) As expected, the policy value increases monotonically as time passes.

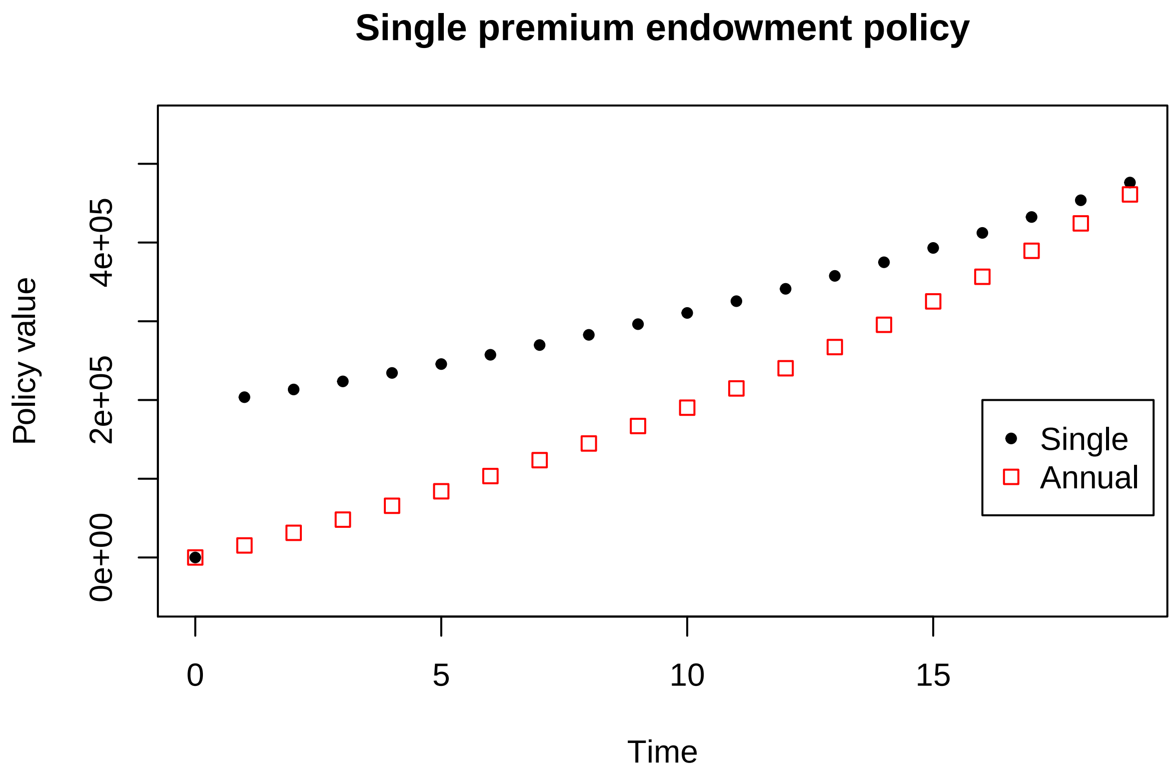
```
#2.4
v=rep(0,20)
j=1
while(j<20){
  v[j+1]=((v[j]+P_ann)*(1+i)-n*q[j])/px[50+j]
  j=j+1
}
Annual_Premium <- v
Annual_Premium
```

```
##  [1]      0.00  15293.18  31312.97  48093.85  65672.40  84087.44 103380.40
##  [8] 123595.64 144780.84 166987.57 190271.83 214694.86 240323.98 267233.71
## [15] 295507.07 325237.24 356529.49 389503.61 424296.86 461067.65
```

```
plot(x=c(0:19), y=v, type = "p",
     xlab = "Time",
     ylab = "Policy value",
     main = "Policy Values(Endowment)",
     pch=20)
```




```
#2.5
v2=rep(0,20)
v2[2]=((P_sing)*(1+i)-n*q[1])/px[51]
k=2
while(k<20){
  v2[k+1]=(v2[k]*(1+i)-n*q[k])/px[50+k]
  k=k+1
}
plot(x=c(0:19), y=v2, type = "p",
     xlab = "Time",
     ylab = "Policy value",
     ylim = c(-51000,550000),
     main="Single premium endowment policy",
     pch=20)
points(x=c(0:19), y=v,pch=0, col="red")
legend(16,200000, c("Single","Annual"), pch = c(20,0),
       col = c("black","red"))
```



Question 3

```
#3.1
q[length(q)]=lt$qx[70]
P_sing=sum(n*p*q*df[-1])
P_sing
```

```
## [1] 20100.41
```

```
#3.2
P_ann=P_sing/sum((df[-21]*p))
P_ann
```

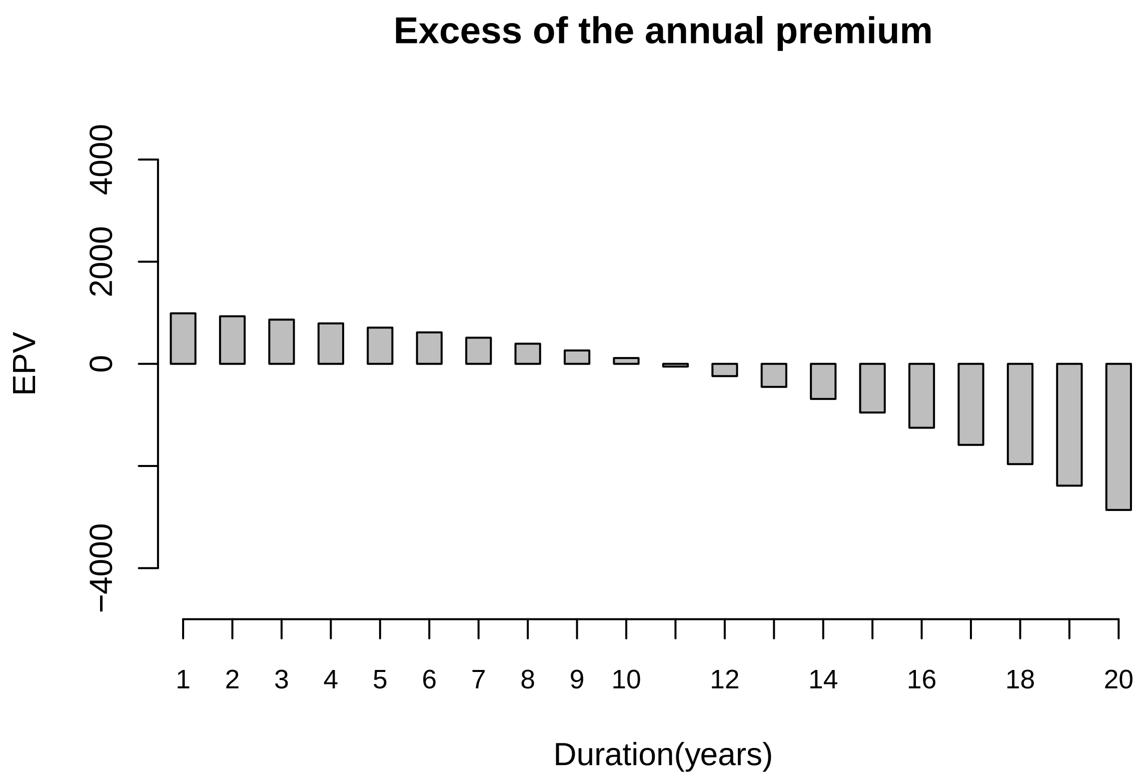
```
## [1] 1565.112
```

(3.3)

```
#3.3
ben=n*q/(1+i)
pre=rep(P_ann, 20)
excess_term2<-c(pre-ben)
excess_term2
```

```
##  [1]   989.6231   931.2839   865.7192   792.0353   709.2282   616.1702
##  [7]   511.5948   394.0795   262.0270   113.6438   -53.0835  -240.4151
## [13]  -450.8877  -687.3473  -952.9871 -1251.3886 -1586.5675 -1963.0253
## [19] -2385.8062 -2860.5606
```

```r
barplot(height=c(pre-ben),
        xlim = c(1,20),
        ylim = c(-5000, 5000),
        width= 0.5,
        space = 1,
        xlab = "Duration(years)",
        ylab = "EPV",
        main="Excess of the annual premium",
        names.arg = sprintf("%d", c(1:20)),
        axes = TRUE,
        axisnames = TRUE,
        axis.lty = 1,
        cex.names = 0.8)
```



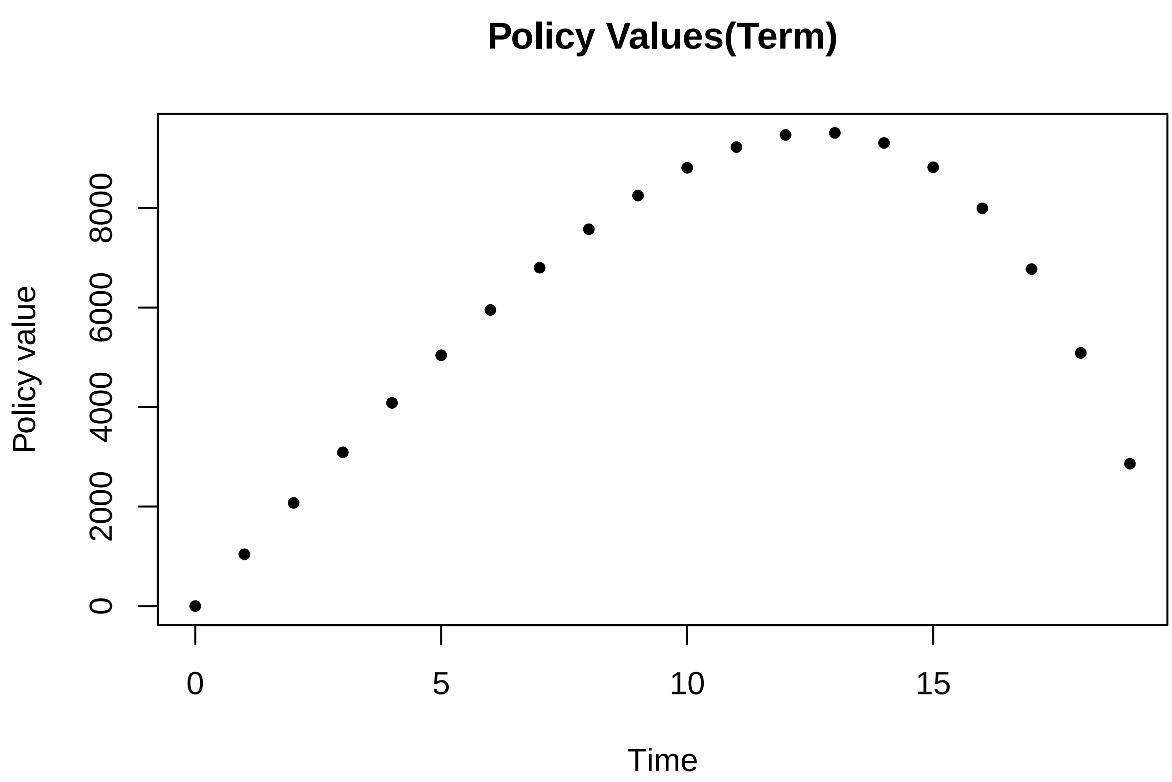
(3.4) As with figure 7.4 in the DHW book, we observe a parabolic curve that opens down.

```r
#3.4
v=rep(0,20)
j=1
while(j<20){
  v[j+1]=((v[j]+P_ann)*(1+i)-n*q[j])/px[50+j]
```

```
  j=j+1
}
Policy_value_term_insurance<-v
v
```

```
##  [1]    0.000 1040.362 2072.987 3090.180 4082.955 5040.852 5951.734 6801.543
##  [9] 7574.029 8250.436 8809.135 9225.203 9469.934 9510.261 9308.088 8819.494
## [17] 7993.791 6772.408 5087.546 2860.561
```

```
plot(x=c(0:19), y=v, type = "p",
     xlab = "Time",
     ylab = "Policy value",
     main = "Policy Values(Term)",
     pch=20)
```




```
#3.5
v2=rep(0,20)
v2[2]=((P_sing)*(1+i)-n*q[1])/px[51]
k=2
while(k<20){
  v2[k+1]=(v2[k]*(1+i)-n*q[k])/px[50+k]
  k=k+1
}
```

```
plot(x=c(0:19), y=v2, type = "p",
     main="Policy values of term insurance",
     xlab = "Time",
     ylab = "Policy value",
     pch=20)
points(x=c(0:19), y=v,pch=0, col="red")
legend(16,21000, c("Single","Annual"), pch = c(20,0),
       col = c("black","red"))
```