Natalie A. Milevsky and Yosef Y. Bisk

Student Solutions Manual:

Retirement Income Recipes in R

February 21, 2021

Springer Nature

Chapter 1
Solutions

No questions to answer for this chapter.
I assume throughout the book that the reader already defined all the user defined functions in each chapter before attempting the questions at the end of the chapter.

Chapter 2
Solutions

2.1 Original Question

Compute and report values of RGOA assuming $N = 20$, for a growth rate of $g = 1\%, 2\%, 3\%$ under $v = 4\%$. Notice the pattern. What happens when you plug-in $g = 4\%$? Why do you get this result?

Answer

I use the RGOA function from the chapter and I plug in a vector of $g = 1\%, 2\%, 3\%$ in place of a single value of g.

```
> RGOA(c(0.01,0.02,0.03),0.04,20)
[1] 14.91841 16.41348 18.09858
```

I see that as the growth rate increases, the present value of the annuity increases as well, because cash flows are higher. Now I check what happens when I plug in $g = 4\%$.

```
> RGOA(0.04,0.04,20)
[1] NaN
```

Because v=g the denominator of the RGOA function equals zero making the analytic formula indeterminate. R is using equation 2.1 in the chapter to solve the problem, which is why it gets stuck when both v and g equal to 0.04. To solve for the value of the annuity when v=g I must go back to first principles. To obtain the formula for the present value of an ordinary annuity, I use the equation for the sum of a geometric progression.

$$Y_N = X + X^2 + X^3 + \ldots + X^N = \sum_{i=1}^{N} X^i$$

Where X can be any number not equal to zero. Using some manipulation, by multiplying all the terms by X and subtracting it from Y_N I get:

$$Y_N - XY_N = \sum_{i=1}^{N} X^i - \sum_{i=1}^{N} X^{i+1}$$

Other than the first and last term, all the terms in the summations on the right side cancel each other out, so I get:

$$Y_N - XY_N = X - X^{N+1}$$

Rearranging to isolate Y_N and dividing by X gives me:

$$Y_N(1 - X) = X - X^{N+1}$$

Dividing both sides by $1 - X$:

$$Y_N = \frac{X - X^{N+1}}{1 - X}$$

Dividing the numerator and denominator of the fraction by X:

$$Y_N = \frac{1 - X^N}{\frac{1}{X} - 1}$$

Replacing X with $\frac{1+g}{1+v}$ gives the formula for the present value of an ordinary annuity:

$$Y_N = \frac{1 - (1+g)^N\,(1+v)^{-N}}{(v-g)/(1+g)}$$

This equation works when $v \neq g$. However, when $v = g$ I can just sum the terms as in the geometric series:

$$Y_N = \frac{(1+0.04)^1}{(1+0.04)^1} + \frac{(1+0.04)^2}{(1+0.04)^2} + \ldots \frac{(1+0.04)^N}{(1+0.04)^N}$$

Each of the individual terms would equal 1, leaving me with $N \times 1$ which gives me the answer of N. Intuitively this means that if my cash-flows grow at the same rate that the value of money gets discounted over time, then the growth cancels out the discounting, and the value of the annuity is just the dollar sum of all the payments.

2.2 Original Question

Write a function that computes the present value of an ordinary annuity (similar to the `RGOA` function) that only starts growing after $M \leq N$ periods. Call it `RGOA.DM`

So, for the first M periods there is no growth and then cash-flows start growing by g per period, until time period N. Evaluate this function with parameters $M = 10$ years, $N = 20$ years, $g = 2\%$ and $v = 3\%$ and compare with the value of RGOA in which $g = 2\%$, $v = 3\%$ and explain the difference.
Answer
To create this function, I add the value of two annuities together. The first annuity is an M year zero-growth annuity. The second annuity is an $N - M$ year growing annuity, which I discount back M years to time 0.

```
RGOA.DM<-function(g,v,N,M){
  ((1-(1+v)^(-M))/v)+((1-((1+g)^(N-M))*(1+v)^(-N+M))
  /((v-g)/(1+g)))/(1+v)^M
}
```

I use my RGOA.DM function with $N = 20$ years, $M = 10$ years, $g = 2\%$, $v = 3\%$.

```
> RGOA.DM(0.02,0.03,20,10)
[1] 15.58516
```

I now use the ordinary RGOA function with $N = 20$ years, $g = 2\%$, $v = 3\%$.

```
> RGOA(0.02,0.03,20)
[1] 18.08123
```

I see that the RGOA.DM has a lower value. This is because RGOA grows for all 20 years, while RGOA.DM only grows for 10 years, and its growth begins later, so the cash-flow payments are lower in the RGOA.DM.

2.3 Original Question

Write a function that computes the future value of an ordinary annuity, and call it RGOA.FV. Evaluate this function with parameters: $N = 20$ years, $g = 2\%$ and $v = 3\%$.
Answer
I create a function RGOA.FV based off the formula for the future value of an ordinary annuity. The future value annuity factor, given g,v,N is equal to the following sum:

$$FV = (1+g)(1+v)^{N-1} + (1+g)^2(1+v)^{N-2} + ... + (1+g)^N$$

The first term represents the future value at time N of the $1 + g$ cash-flow received after one period, the second term is the future value of $(1 + g)^2$ which is received after the second period and invested for (N-2) periods, and so on. After some rearranging the formula can be expressed as follows:

$$FV = \frac{(1+v)^N - (1+g)^N}{(v-g)/(1+g)}$$

I got this by multiplying the regular `RGOA` function by $(1+v)^N$

$$(1+v)^N \texttt{RGOA} = (1+v)^N \times \frac{1-(1+g)^N(1+v)^{-N}}{(v-g)/(1+g)} = \frac{(1+v)^N-(1+g)^N}{(v-g)/(1+g)}$$

I now code this function which takes in values for g,v and N.

```
> RGOA.FV<-function(g,v,N){(((1+v)^N-(1+g)^N))/((v-g)/(1+g))}
```

I evaluate this function with parameters: $g = 2\%$, $v = 3\%$, $N = 20$.

```
> RGOA.FV(0.02,0.03,20)
[1] 32.65671
```

Alternatively, I can multiply the original RGOA function by $(1+v)^N$ and get the same result.

```
> RGOA(0.02,0.03,20)*(1+0.03)^20
[1] 32.65671
```

I compare the future value to the present value of the same RGOA at time 0.

```
> RGOA(0.02,0.03,20)
[1] 18.08123
```

I see that an annuity that would costs me 18.08 now, will cost me 32.66 if I pay for it in 20 years.

2.4 Original Question

You are $x = 65$ years-old and entitled to an income (cash-flow, pension) of \$1,000 per month for the rest of your life. Assuming $v = 3\%$, what is the present value of that pension if you live to age 95? What is the present value if you live to age 100? What is the extra cost of those five years?

Answer

I use $v = 3/12 = 0.25\%$ as my monthly valuation rate.

```
> x<-65
> v<-3/12
#months until age 95
> N1<-(95-x)*12
#months until age 100
> N2<-(100-x)*12
> 1000*RGOA(0,v,N1)
[1] 237189.4
> 1000*RGOA(0,v,N2)
[1] 259841.4
```

If I die at age 95, the present value of my pension is \$237, 189. but if I die at age 100 then the present value is \$259, 841. The extra cost of those 5 years is \$22, 652, even though I am spending an extra \$60, 000 (\$12, 000 each year). This tells me the significance of the time value of money, and how a small investment today can go a long way in the distant future.

2.5 Original Question

Assume that you have 100 "units" of something (think of dollars, euros, yen or people who are infected with a virus). Every day that number grows (think of interest rates) by 30%. At the end of the first day you have 130 units, at the end of the second day you have $100 \times (1.3)^2$ units, etc. In how many days will you have 300 million units?

Answer

Let U_s be the number of units I start with, and U_f be the number of units I finish with. Let g be my daily growth rate, and t be the number of days that past.

$$U_f = U_s(1+g)^t$$

I rearrange to solve for t.

$$\frac{U_f}{U_s} = (1+g)^t$$

$$t \times ln(1+g) = ln(\frac{U_f}{U_s})$$

$$t = \frac{ln(\frac{U_f}{U_s})}{ln(1+g)}$$

I calculate t using this equation and our values in **R**.

```
> log(300000000/100)/log(1.3)
[1] 56.8451
```

It would take approximately 57 days to have 300 million units if I start with 100 units and grow at 30% a day. Exponential growth accumulates quickly.

Chapter 3
Solutions

3.1 Original Question

Build an expanded version of the `SMCR` function, which you can label `SMCR.PN`, which assumes that you are entitled to a government pension (for example Social Security) of π times your final wage w_R. So, for example, if you final wage is $w_R = 100{,}000$ per year, you might be entitled to a guaranteed pension of $\pi = (0.60)$ of you final 100,000 salary between the ages of (for example) $R = 65$ and $D = 95$. So, please compute some values for the `SMCR.PN` function.

Answer

The original `SMCR` function divides the present value of wealth and all future income by the ordinary annuity factor representing all future spending.

$$\mathtt{SMCR} = \frac{wRGOA(g, v, R - x) + f}{RGOA(0, v, D - x)}$$

To create `SMCR.PN` I add the present value of the pension to the total wealth and present value of future wages.

$$\mathtt{SMCR.PN} = \frac{wRGOA(g, v, R - x) + f + \pi f_p}{RGOA(0, v, D - x)}$$

Where f_p is the present value of a pension whose annual payment is my final wage at retirement. The present value of my pension is πf_p, since the pension I am entitled to pays π times my final wage. To calculate f_p, first I must calculate my final salary w_R:

$$w_R = w(1 + g)^{R-x}$$

I calculate the present value of a pension that will pay w_R each year after retirement.

$$f_p = \frac{w_R \times RGOA(0, v, D - R)}{(1 + v)^{R-x}}$$

I now have all the components I need to code `SMCR.PN` in **R**.

```
SMCR.PN<-function(x,f,w,g,v,R,D,pi){
  #wR is the final salary
  wR<-(w*(1+g)^(R-x))
  #fp is present value of pension
  fp<-wR*RGOA(0,v,D-R)/(1+v)^(R-x)
  consumption<-(w*RGOA(g,v,R-x)+f+fp*pi)/RGOA(0,v,D-x)
  return(consumption)}
```

I test my function using the following parameters: $x = 35$, $f = 0$, $w = 60000$, $g = 0.01$, $v = 0.04$, $R = 65$, $D = 95$, $p = 0.6$.

```
> SMCR.PN(35,0,60000,0.01,0.04,65,95,0.6)
[1] 63617.53
```

Notice that my smooth consumption rate is higher than my initial wage. This means I can borrow money in my early years to finance my spending. I now try the same example but with no pension.

```
> SMCR.PN(35,0,60000,0.01,0.04,65,95,0.0)
[1] 52182.7
```

In the absence of a pension I must reduce my annual spending by $\$11,434.84$. I now check what happens if I start working (and spending and saving) at other ages of $x = 20$ and $x = 45$.

```
#starting at 20 with a pension
> SMCR.PN(20,0,60000,0.01,0.04,65,95,0.6)
[1] 69493.01
#starting at 20 without a pension
> SMCR.PN(20,0,60000,0.01,0.04,65,95,0.0)
[1] 62450.61
#starting at 45 with a pension
> SMCR.PN(45,0,60000,0.01,0.04,65,95,0.6)
[1] 57804.61
#starting at 45 without a pension
> SMCR.PN(45,0,60000,0.01,0.04,65,95,0.0)
[1] 41667.31
```

If I start working earlier, I can allow myself to spend more each year regardless of pension value. Also, the difference the pension makes to my annual spending when I am 45 is larger than when I am 20 ($16,137 at 45 vs $7,042 at 20), because the value of the pension needs to be distributed over a shorter remaining lifetime.

3.2 Original Question

Continuing the above line of thinking, assume the same π pension rate, now create a new function OLCF.PN for the amount of financial capital you should have accumulated at any age x. Compute this multiple assuming you enter the work-force at age $x_0 = 25$, earn $w_{25} = 50000$ in your first year of work, it grows by $g = 1\%$ per year until you retire at age $R = 62$. Assume you (plan to) live to age $D = 95$, the valuation rate is $v = 2.5\%$ and the pension rate is $\pi = 35\%$ of your final salary. Of course, there is an implicit assumption here that you are entitled to keep your entire wage w_x and that you don't pay any tax (or pension contributions) which go towards funding the entitlement of π.

Answer

The OLCF.PN function I create is similar to the OLCF function in the chapter, but with two differences. It uses SMCR.PN defined in the previous question instead of SMCR, and it also subtracts the present value of the pension.

```
OLCF.PN<-function(x,x0,f0,w0,g,v,R,D,pi){
  if (x<=R){
    #consumption with pension
    PVC<-SMCR.PN(x0,f0,w0,g,v,R,D,pi)*RGOA(0,v,D-x)
    #human capital
    HC<-((w0*(1+g)^(x-x0))*RGOA(g,v,R-x))
    #present value of pension
    PVP<-(pi*(w0*(1+g)^(R-x0))*RGOA(0,v,D-R))/(1+v)^(R-x)
    #financial capital saved up
    Fx<-PVC-HC-PVP
    return(Fx)}
  else{
    SMCR.PN(x0,f0,w0,g,v,R,D,pi)*RGOA(0,v,D-x)-
    (pi*(w0*(1+g)^(R-x0))*RGOA(0,v,D-x))}}
```

I define my notation before I explain my calculation. PVC is the present value of all future consumption. HC is Human capital or the present value of future wages. PVP is the present value of the future pension payments. The way OLCF.PN is calculated, depends on the persons age x, if $x \leq R$:

$$\texttt{OLCF.PN} = \texttt{PVC} - \texttt{HC} - \texttt{PVP}$$

If $x > R$ then human capital is zero because I am no longer working, so the financial capital I should have is the present value of my future consumption subtracting the present value of whats left of my pension:

$$\texttt{OLCF.PN} = \texttt{PVC} - \texttt{PVP}$$

I run OLCF.PN for the parameters given in the question:

```
> OLCF.PN(35,25,0,50000,0.01,0.025,62,95,0.35)
[1]  31697.94
```

At age 35 I need to have \$31,698 accumulated, assuming I will receive a pension that is 35% of my final wage. How does that compare to a situation where I have no pension?

```
> OLCF.PN(35,25,0,50000,0.01,0.025,62,95,0.0)
[1] 108695
```

I see that if I had no pension ($\pi = 0$) I would need to have saved more than three times the amount I must save when I have a pension. Having a pension drastically reduces the amount I must save.

3.3 Original Question

Create a function `FCWM.PN`, which computes the multiple of your wage that you should have saved-up at any age x, assuming you are entitled to a pension of π at retirement. Assuming the same $g = 1\%$, $v = 2.5\%$ numbers (which were the basis for the above table), please locate scenarios (i.e. values of w, R or D) where there is no need to accumulate (or save) any money for retirement, because the pension of $\pi = 60\%$ will be enough to finance a constant smooth consumption plan.

Answer

To solve for the wage multiple F_x/w_x while including a pension, I modify the original `FCWM` function to include a pension. From the previous question, I know the equation for F_x including a pension is:

$$\mathtt{F_x = PVC - HC - PVP}$$

The present value of future consumption is:

$$\mathtt{PVC = c(RGOA(0, v, D - x));}$$

Future annual consumption is:

$$\mathtt{c = \frac{w_0(RGOA(g, v, R - x_0)) + PVP_0}{RGOA(0, v, D - x_0)}}$$

At time 0, the present value of the pension I will receive after retirement is:

$$\mathtt{PVP_0 = \frac{\pi w_0 (1 + g)^{(R - x_0)} RGOA(0, v, D - R)}{(1 + v)^{(R - x_0)}}}$$

My human capital is the present value of my future wages:

$$\mathtt{HC} = \mathtt{w_x}(\mathtt{RGOA}(\mathtt{g},\mathtt{v},\mathtt{R}-\mathtt{x}))$$

The present value of my pension is:

$$\mathtt{PVP} = \frac{\pi(\mathtt{w_x}(1+\mathtt{g})^{(\mathtt{R}-\mathtt{x})}\mathtt{RGOA}(0,\mathtt{v},\mathtt{D}-\mathtt{R}))}{(1+\mathtt{v})^{(\mathtt{R}-\mathtt{x})}}$$

Also note that:

$$\frac{w_0}{w_x} = (1+g)^{(x_0-x)}$$

Dividing through by w_x, I are left with the wage multiple:

$$\begin{aligned}
\left(\frac{\mathtt{F_x}}{\mathtt{w_x}}\right) = {} & (1+\mathtt{g})^{(\mathtt{x_0}-\mathtt{x})}\frac{\mathtt{RGOA}(0,\mathtt{v},\mathtt{D}-\mathtt{x})(\mathtt{RGOA}(\mathtt{g},\mathtt{v},\mathtt{R}-\mathtt{x_0}))}{\mathtt{RGOA}(0,\mathtt{v},\mathtt{D}-\mathtt{x_0})} \\
& + \frac{\pi(1+g)^{(R-x)}\mathtt{RGOA}(0,\mathtt{v},\mathtt{D}-\mathtt{R})(\mathtt{RGOA}(\mathtt{g},\mathtt{v},\mathtt{D}-\mathtt{x}))}{(1+\mathtt{v})^{(\mathtt{R}-\mathtt{x_0})}\mathtt{RGOA}(0,\mathtt{v},\mathtt{D}-\mathtt{x_0})} \\
& - \mathtt{RGOA}(\mathtt{g},\mathtt{v},\mathtt{R}-\mathtt{x}) \\
& - \frac{\pi(1+\mathtt{g})^{\mathtt{R}-\mathtt{x}}\mathtt{RGOA}(0,\mathtt{v},\mathtt{D}-\mathtt{R})}{(1+\mathtt{v})^{\mathtt{R}-\mathtt{x}}}
\end{aligned}$$

The first line of the equation is the part of future consumption that is due to wages. The second line is the part of future consumption that is due to the pension. The third line is the present value of future wages. The fourth line is the present value of the pension I will receive. Note that I don't need to consider times when $x > R$ since wages are 0 so FCMW is undefined. I code this up in R:

```
FCMW.PN<-function(x,x0,g,v,R,D,pi){
  #first numerator of PVC
  PVC1N<-((1+g)^(x0-x))*RGOA(0,v,D-x)*RGOA(g,v,R-x0)
  #first denominator of PVC
  PVC1D<-RGOA(0,v,D-x0)
  #second numerator of PVC
  PVC2N<-pi*((1+g)^(R-x))*RGOA(0,v,D-R)*RGOA(g,v,D-x)
  #second denominator of PVC
  PVC2D<-((1+v)^(R-x0))*RGOA(0,v,D-x0)
  PVC<-(PVC1N/PVC1D)+(PVC2N/PVC2D)
  HC<-RGOA(g,v,R-x)
  PVP<-pi*((1+g)^(R-x)*RGOA(0,v,D-R))/(1+v)^(R-x)
  return(PVC-HC-PVP)}
```

To find scenarios where I don't need to save for retirement, I loop through different retirement ages and death ages to find where the value of the wage multiple is zero. If the value is zero, it means that if I retire at age R and die at age D, I do not need to save for retirement since the amount I are receiving from my pension is sufficient.

```
R<-list()
D<-list()
z<-list()
#loop over possible R values
for (i in 30:110){
  #loop over possible D values
  for (j in 70:130){
    n<-FCMW.PN(i,25,0.01,0.025,i,j,0.6)
    #check if FCMW.pn is close to 0, and if R<D
    if (n>-.005 & n<.005 & i<j){
      z <- append(z, n)
      R <- append(R,i)
      D <- append(D,j)}}}
q<-as.matrix(cbind(R,D,z))
```

In this R script I am using a new object called `list` to store values. This object does not appear in the textbook, but its useful when solving problems where lists of values must be stored. I also use two other functions `as.matrix` and `cbind` which help me combine lists into a table so the relationships can be easily presented. I include the column `z` to show how close `FCMW.PN` is to 0. Here is the output.

```
> q
     R   D   z
[1,] 107 108 0.001585261
[2,] 107 109 0.003204122
[3,] 107 110 0.004912605
[4,] 108 109 -0.002752728
[5,] 108 125 -0.002093542
[6,] 108 126 0.001614958
```

I have to work for at least 82 years if my pension is 60% of my final wage ($\pi = 0.6$) and I want to retire without any savings. Working till 107 is obviously ridiculous, so I conclude that it is not feasible to retire without savings assuming these values of x_0, g, v, and π.

3.4 Original Question
You earn $w = \$100{,}000$ per year, the valuation rate is $v = 4\%$, wage growth rate is $g = 0$, you are $x = 35$ and have nothing saved $f = 0$. Please locate values of (R, D) that result in $F_R = \$1{,}000{,}000$. In other words, find situations in which you really do need one million dollars to retire.
Answer

I try to solve this in two ways, either by finding *R* as a function of *D* and plugging in different values, or by simulating many possible *R* and *D* values and seeing which combinations require 1 million at retirement. Trying the first way:
Present value of consumption during retirement must equal $1 million:

$$c^* \frac{(1-1.04^{-(D-R)})}{.04} = 1,000,000 \tag{3.1}$$

After some manipulation:

$$c^* = \frac{40,000}{1-1.04^{-(D-R)}} \tag{3.2}$$

PV of Human Capital must equal PV of consumption, which leads to:

$$c^* \frac{(1-1.04^{-(D-35)})}{.04} = 100,000\frac{(1-1.04^{-(R-35)})}{.04} \tag{3.3}$$

I know $c^* = \frac{40,000}{1-1.04^{-(D-R)}}$, which leads to:

$$\frac{40,000(1-1.04^{-(D-35)})}{.04(1-1.04^{-(D-R)})} = 100,000\frac{(1-1.04^{-(R-35)})}{.04} \tag{3.4}$$

$$\frac{2(1-1.04^{-(D-35)})}{1-1.04^{-(D-R)}} = 5(1-1.04^{-(R-35)}) \tag{3.5}$$

$$1-1.04^{-(D-35)} = 2.5(1-1.04^{-(R-35)})(1-1.04^{-(D-R)}) \tag{3.6}$$

$$1-1.04^{-(D-35)} = 2.5(1-1.04^{-(R-35)}-1.04^{-(D-R)}+1.04^{-(D-35)}) \tag{3.7}$$

$$0 = 1.5-2.5(1.04^{-(R-35)}+1.04^{-(D-R)})+3.5(1.04)^{(-D-35)} \tag{3.8}$$

At this point it is very difficult to isolate *R* since the relationship between *R* and *D* is non-linear, which leads me to try the alternate way of solving this problem, using the OLCF function in **R**.

```
R<-list()
D<-list()
z<-list()
#loop over possible R values
for (i in 48:75){
  #loop over possible D values
  for (j in 80:120){
    n<-OLCF(i,35,0,100000,0.0,0.04,i,j)
    #check if OLCF is close to 1,000,000
    if (n>998000 & n<1002000){
      z <- append(z, n)
      R <- append(R,i)
      D <- append(D,j)}}}
q<-as.matrix(cbind(R,D,z))
```

This script shows the combinations of R and D that require \$1 million to retire (give or take \$2,000). Here is the output.

```
     R  D   z
[1,] 49 103 998621.1
[2,] 49 104 1000999
[3,] 50 91  1000415
[4,] 51 85  998654.1
[5,] 52 82  999713.9
[6,] 65 82  999713.9
[7,] 69 85  998654.1
```

I see that if I retire later I need to die earlier, in order for my optimal savings at retirement to be \$1 million. The intuition is that the more years I will work, the higher my smooth consumption rate will be. A higher smooth consumption rate means I must spend less years in retirement (die earlier) for the present value of my retirement spending to be \$1 million at the time I retire.

3.5 Original Question

Similar to question (3.4), assume ($R = 65, D = 100$) but modify the wage growth rate g and the valuation rate v, and locate parameters for which you need one million dollars to retire.

Answer

Similarly to 3.4, I wrote an **R** script to test different combinations of `v` and `g` using the `OLCF` function.

```
G<-list()
V<-list()
z<-list()
#loop over possible g values
for (g in 1:100*.001){
  #loop over possible v values
  for (v in 1:100*.001){
    if (g!=v){
      n<-OLCF(65,35,0,100000,g,v,65,100)
      #check if OLCF is close to 1,000,000
      if (n>998000 & n<1002000){
        z <- append(z, n)
        G <- append(G,g)
        V <- append(V,v)}}}}
q<-as.matrix(cbind(G,V,z))
```

Here is the output.

```
     G     V     z
[1,] 0.002 0.09  999892.8
[2,] 0.005 0.094 998716
[3,] 0.006 0.095 1000751
[4,] 0.009 0.099 998873.6
[5,] 0.01  0.1   1000730
```

I see that for higher v I need higher g values, assuming I need one million dollars at retirement, $R = 65$, and $D = 100$. The intuition is that if the valuation rate is higher, $1 million at retirement would fund higher annual consumption. For this higher annual consumption during retirement to match my annual consumption during my working years, I must spend more in my working years, which requires my salary to grow more.

Chapter 4
Solutions

4.1 Original Question

Assume that the real projected growth rate of wages is $g = 1.0\%$ and that real valuation (or interest) rate is $v = 2.5\%$. What fraction of the entire group *do not* have enough financial capital to maintain their standard of living? Please use the functions and metrics I designed previously to support your answer.

Answer

To solve this problem, I loop though all the rows in SIMPBS. I check if they are still working (i.e. they are younger than their specific retirement age). If they are still working, I check if their spending exceeds their smooth consumption rate `SMCR`. If they retired, I check if the present value of their future consumption exceeds their financial capital. If either of these conditions are true, then I add them to the count of `overspending` (which I set to 0 before the loop). I then divide the number of overspenders by the length of SIMBPS (my total population), to get the fraction of people that do not have the capital to maintain their standard of living. I assumed everyone will live until age 95, because that is the oldest age in the dataset.

```
#renaming columns for easier use
#renaming columns for easier use
SP<-SIMPBS$SPENDING
WG<-SIMPBS$CRNTWG
AG<-SIMPBS$AGEFAM
RG<-SIMPBS$RETAGE
FC<-SIMPBS$FINCAP
# placeholder to count overspenders
overspending<-0
#loop through all people
for(i in 1:length(FC)){
  #check if they are still working
  if (AG[i]<RG[i]){
    #check if spending exceeds smooth consumption rate
    if(SMCR(AG[i],FC[i],WG[i],0.01,0.025,RG[i],95)<SP[i])
```

```
   #increase the count by 1
   {overspending<-overspending+1}}
 #if they retired
 else{
  #check if PV of future spending exceeds financial capital
  if(FC[i]<SP[i]*RGOA(0,0.025,95-AG[i]))
  #increase the count by 1
  {overspending<-overspending+1}}}
```

Now I divide the number of over-spenders by the number of people in the set. Here are the results:

```
> overspending/length(FC)
[1] 0.8848
```

Meaning 88% of families do not have enough financial capital to maintain their standard of living under my assumptions. Note that because I am assuming the same life expectancy for my entire population, and there are 77 people that are alive at age 95, I can not assume an age of death below 95. However, I still want to work with a value of `D` below 95 so I will choose to assume that those who survived past the expected age of death `D` have enough capital to maintain their spending. This means when I loop through the population to check for overspenders, I do not need to check the people who are older than `D`. I check what happens if I lower the assumed age of death to the Canadian (2019) life expectancy of 82 years:

```
#placeholder to count overspenders
overspending<-0
#loop through all people
for(i in 1:length(FC)){
 #check if they are under 82
 if (AG[i]<82){
  #check if they are working
  if (AG[i]<RG[i]){
   #check if they are overspending and count if yes
   if(SMCR(AG[i],FC[i],WG[i],0.01,0.025,RG[i],82)<SP[i])
  {overspending<-overspending+1}}
  #if they are not working (they retired)
  else{
   #count those whose future consumption exceeds capital
   if (FC[i]<SP[i]*RGOA(0,0.025,82-AG[i])) {
    overspending<-overspending+1}}}}
```

```
> overspending/length(FC)
[1] 0.1736
```

At this lower life expectancy only 17% of the population do not have enough to maintain their standard of living. This result is more comforting than the 88% that were underfunded when I assumed death at age 95. This is consistent with the intuition that higher life expectancy requires greater capital for retirement. Note, that I assumed everyone older than 82 had enough money, and this assumption is probably false in reality.

4.2 Original Question

What fraction of the working population (sample) will not have enough to maintain their standard of living? Segment by age buckets.

Answer

To solve this problem, I create a function called `Frac_bucket` which returns the fraction of an age bucket that will not have enough to maintain their standard of living. The function takes in a lower bucket age limit x_1, and an upper bucket age limit x_2, along with an age of death D. I do not need to worry about people that are older than the death age D, because I can set the upper bucket limit x_2 to be less than D.

```
#function takes bucket of working people between age x1 and x2
#returns fraction that can not maintain standard of living
#assuming they die at age D
Frac_bucket<-function(x1,x2,D){
  #isolate members of the bucket
  SP_b<-SP[AG<=x2 & AG>x1 & AG<RG]
  WG_b<-WG[AG<=x2 & AG>x1 & AG<RG]
  AG_b<-AG[AG<=x2 & AG>x1 & AG<RG]
  RG_b<-RG[AG<=x2 & AG>x1 & AG<RG]
  FC_b<-FC[AG<=x2 & AG>x1 & AG<RG]
  #reset counter to 0
  overspending<-0
  #loop through bucket members
  for (i in 1:length(FC_b)){
    #if they spend more than SMCR add them to the count
    if (SMCR(AG_b[i],FC_b[i],WG_b[i],0.01,0.025,RG_b[i],D)
    <SP_b[i]){overspending<-overspending+1}}
  #divide number of overspenders by bucket population size
  return(overspending/length(SP_b))}
```

I run this function on decade long buckets from age 20 to 70.

```
> Frac_bucket(20,30,85)
[1] 1
> Frac_bucket(30,40,85)
[1] 0.8890357
> Frac_bucket(40,50,85)
[1] 0.6900585
```

```
> Frac_bucket(50,60,85)
[1] 0.2543103
> Frac_bucket(60,70,85)
[1] 0.01362398
```

It looks like younger people are more likely to spend beyond their means. Maybe as people get older they begin thinking about retirement more seriously, and realize they must save more and spend less. Note that I assumed a life horizon of 85. Had I chosen $D = 95$ these fractions would be much higher. I now do that same calculation using $D = 95$ below, and see that the fraction of working people that can not maintain their standard of living is indeed much higher, and the variation between the age buckets is much smaller.

```
> Frac_bucket(20,30,95)
[1] 1
> Frac_bucket(30,40,95)
[1] 0.989432
> Frac_bucket(40,50,95)
[1] 0.9859649
> Frac_bucket(50,60,95)
[1] 0.987069
> Frac_bucket(60,70,95)
[1] 0.972752
```

Its interesting that for this higher life expectancy, the 40-50 bucket is doing better than the 50-60 bucket, contrary to what I saw when $D = 85$. The explanation for this can be that when I assume survival until 95, only a hand-full of people are actually able to maintain their standard of living. Because the numbers are so small they may not be reflective of the general population. I modify the `Frac_bucket` function and create another function called `Frac_bucket2` which returns the size of each bucket along with the number of people that are not over spending.

```
#function takes bucket of working people between age x1 and x2
#returns bucket size and number of non-overspenders
Frac_bucket2<-function(x1,x2,D){
  #isolate members of the bucket
  SP_b<-SP[AG<=x2 & AG>x1 & AG<RG]
  WG_b<-WG[AG<=x2 & AG>x1 & AG<RG]
  AG_b<-AG[AG<=x2 & AG>x1 & AG<RG]
  RG_b<-RG[AG<=x2 & AG>x1 & AG<RG]
  FC_b<-FC[AG<=x2 & AG>x1 & AG<RG]
  #reset counter to 0
  overspending<-0
  #loop through bucket members
  for (i in 1:length(FC_b)){
    #check if they spend more than SMCR
    if (SMCR(AG_b[i],FC_b[i],WG_b[i],0.01,0.025,RG_b[i],D)
     <SP_b[i])
      #add overspenders to the count
      {overspending<-overspending+1}}
  #return bucket size and number of non-overspenders
  return(c(length(SP_b),length(SP_b)-overspending))}
```

I run this function on my buckets, first assuming death at 85 and then at 95.

```
> Frac_bucket2(20,30,85)
[1] 4 0
> Frac_bucket2(30,40,85)
[1] 757  84
> Frac_bucket2(40,50,85)
[1] 855 265
> Frac_bucket2(50,60,85)
[1] 696 519
> Frac_bucket2(60,70,85)
[1] 367 362

> Frac_bucket2(20,30,95)
[1] 4 0
> Frac_bucket2(30,40,95)
[1] 757   8
> Frac_bucket2(40,50,95)
[1] 855  12
> Frac_bucket2(50,60,95)
[1] 696   9
> Frac_bucket2(60,70,95)
[1] 367  10
```

I see that when $D = 95$ there are at most 12 people in a bucket that can actually maintain their spending until age 95. This number is not statistically significant. I would require a bigger population to get a better understanding of the actual probability distribution of a person overspending, given their age bucket, assuming $D = 95$.

4.3 Original Question

What fraction of the population have 20% less than what they need to maintain their standard of living? (Call this financial distress.)

Answer

To solve this problem, I make some modifications to the code in the solution from question 4.1. Taking into account the entire population, but checking to see if the financial capital is less than 80% of what it should be. The financial capital should equal the present value of future consumption minus human capital. Here is the code in R:

```
#placeholder to count financially distressed people
distressed<-0
#loop through all people
for(i in 1:length(FC)){
  #present value of their future consumption
  FCR<-SP[i]*RGOA(0,0.025,95-AG[i])
  #check if they are still working
  if (AG[i]<RG[i]){
    #human capital is PV of future wages
    HC<-WG[i]*RGOA(0.01,0.025,RG[i]-AG[i])}
  #if they are not working set human capital to 0
  else{HC<-0}
  #check if financial capital is below 80% of needed
  if (FC[i]<0.8*(FCR-HC)){
    #count as distressed if true
    distressed<-distressed+1}}
```

The percent of people in financial distress is 51.4% which can be seen in the box below:

```
> distressed/length(FC)
[1] 0.514
```

Even though I saw in question 4.1 that 88% of people are overspending, only about half the population is more that 20% away from where they should be, assuming $D = 95$.

4.4 Original Question

Assume a $g = 2\%$ growth rate and $v = 4\%$ valuation (interest) rate. Does the situation improve? Please describe.

Answer

When the growth and valuation rate change to $g = 2\%$ and $v = 4\%$, the situation drastically improves. Now the percentage of the population that are in financial distress drops from 51% to 7% which can be seen in the script and output below.

```
#placeholder to count distressed
distressed<-0
#loop through all people
for(i in 1:length(FC)){
  #present value of their future consumption
  FCR<-SP[i]*RGOA(0,0.04,95-AG[i])
  #check if they are still working
  if (AG[i]<RG[i]){
    #human capital is PV of future wages
    HC<-WG[i]*RGOA(0.02,0.04,RG[i]-AG[i])}
  #if they are not working human capital is 0
  else{HC<-0}
  #check if financial capital is below 80% of needed
  if (FC[i]<0.8*(FCR-HC)){
    #count as distressed if true
    distressed<-distressed+1}}
```

```
#divide distressed cases by all cases
> distressed/length(FC)
[1] 0.07
```

The fraction of the population that is in financial distress drops down by 44 percentage points from question 4.3 which used a growth rate of $g = 1.0\%$ and a real valuation rate of $v = 2.5\%$. I see that assuming a higher valuation rate and wage growth, improves the situation and reduces the need for savings earlier in life, because financial capital is expected to grow more quickly, and human capital is higher due to the higher growth rate of wages.

4.5 Original Question

For the age groups between $x = 30$ and $x = 40$ that was analyzed in the last section, can you locate a set of g and v parameters in the financial life-cycle model that will actually fit the observed 62.6% salary multiple that families had accumulated on average?

Answer

To solve this problem, I write an **R** script which loops through growth rates and valuation rates and evaluates the mean wage multiple they must have saved. I store the pairs of g and v that result in an average wage multiple that is between 61.6% and 63.6%.

```
#isolate people between ages 30 and 40
SP<-SIMPBS$SPENDING[SIMPBS$AGEFAM<=40 & SIMPBS$AGEFAM>=30]
WG<-SIMPBS$CRNTWG[SIMPBS$AGEFAM<=40 & SIMPBS$AGEFAM>=30]
AG<-SIMPBS$AGEFAM[SIMPBS$AGEFAM<=40 & SIMPBS$AGEFAM>=30]
RG<-SIMPBS$RETAGE[SIMPBS$AGEFAM<=40 & SIMPBS$AGEFAM>=30]
FC<-SIMPBS$FINCAP[SIMPBS$AGEFAM<=40 & SIMPBS$AGEFAM>=30]

#create lists to store g and v values
G<-list()
V<-list()

#loop through g values up to 10%
for (g in 1:100*.001){
  #loop through v values up to 10%
  for (v in 1:100*.001){
    #check that v does not equal g
    if (g!=v){
      #evaluate the population mean wage multiple
      m<-mean((SP/WG)*RGOA(0,v,95-AG)-RGOA(g,v,RG-AG))
      #check if wage multiple is within 1% of 62.6%
      if (m>.616 & m<.636){
        #if it is record the values of g and v
        G <- append(G,g)
        V <- append(V,v)}}}}
#put the g and v value into a matrix
q<-as.matrix(cbind(G,V))
```

Here is the output.

```
> q
     G     V
[1,] 0.001 0.068
[2,] 0.003 0.063
[3,] 0.005 0.058
[4,] 0.007 0.053
[5,] 0.01  0.046
[6,] 0.015 0.035
[7,] 0.016 0.033
[8,] 0.021 0.023
```

I see that for low growth rates like 0.1%, I require high valuation rates like 6.8%, but for higher growth rates, like 2.1%, a lower valuation rate of 2.3% will suffice to warrant a 62.6% salary multiple in financial capital.

Chapter 5
Solutions

5.1 Original Question

Create a modified version of the main script (recipe) that generates monthly trajectories, assumes monthly consumption withdrawals of $c/12$, and (obviously) generates monthly investment returns. Call this function the `PLSM.12` to remind you that everything has been converted to monthly values. Now, assume $\nu = 3.5\%$, $\sigma = 20\%$, which remember are annual values, and compute portfolio longevity for $c = 4, 5, 6, 7$ using a total of $N = 10,000$ simulation runs. Compare the median portfolio longevity to the annual values (which appear in the chapter). Finally, compute the probability (or more accurately the frequency for which) portfolio longevity will be less than 30 years, that is 360 months, for each of these four withdrawal rates. This is the (infamous) *30-year failure* rate. Compare with the results in the chapter and discuss the impact of withdrawal frequency on these probabilities.

Answer

To create a function which generates monthly portfolio trajectories, I modify the original **R** script from the chapter as follows. When `c` or `nu` (ν) appear in the function I divide them by 12 and I divide `sigma` by `sqrt(12)`. Also, instead of storing 100 columns of data representing 100 years, I store 1200 columns representing 1200 months. Finally I divide the end result `PLV` by 12, so that my values are in years.

```
PLSM.12<-function(F,c,nu,sigma,N){
  path<-matrix(nrow=N,ncol=1200)
  PLV<-c()
  for (i in 1:N){
    return<-exp(rnorm(1200,nu/12,sigma/sqrt(12)))
    path[i,1]<-F
    for (j in 2:1200){
      path[i,j]<-path[i,j-1]*return[j]-(c/12)
      if (path[i,j]<=0) {break}}
  PLV[i]=j/12}
  PLV}
```

Now that I have my `PLSM.12` function, I use it to compute median portfolio longevity for consumption values of `c` ranging from 4 to 7:

```
> L4.12<-PLSM.12(100,4,0.035,0.2,10000)
> L5.12<-PLSM.12(100,5,0.035,0.2,10000)
> L6.12<-PLSM.12(100,6,0.035,0.2,10000)
> L7.12<-PLSM.12(100,7,0.035,0.2,10000)
> median(L4.12)
[1] 47.25
> median(L5.12)
[1] 31
> median(L6.12)
[1] 23.58333
> median(L7.12)
[1] 18.91667
```

In the chapter, when withdrawals were annual, the median portfolio longevity values were 50, 34, 25, 20. In contrast, when withdrawals were monthly, I got 47, 31, 24, 19. It appears that for all consumption rates, increasing the withdrawal frequency from annual to monthly reduces the portfolio longevity. This makes sense intuitively, because the models assume consumption happens at the end of each period, so the annual withdrawals model would mark the time of ruin at the end of the year, while the monthly model would mark the time of ruin earlier at the end of the month. I now look at the probability of the portfolio lasting 30 years.

```
> length(L4.12[L4.12<30])/length(L4.12)
[1] 0.3351
> length(L5.12[L5.12<30])/length(L5.12)
[1] 0.4838
> length(L6.12[L6.12<30])/length(L6.12)
[1] 0.6148
> length(L7.12[L7.12<30])/length(L7.12)
[1] 0.7283
```

In the chapter, the *30-year failure* rates were 32%, 46%, 60%, 71% for annual withdrawals of $c = 4, 5, 6, 7$. For monthly withdrawals I got 34%, 48%, 61%, 73%. I see that monthly withdrawals result in higher failure rate than annual withdrawals. This makes sense because if ruin were to happen during the first 11 months of year 30, the annual model would treat is as a 30-year success, while the monthly model would treat it as 30-year failure. I conclude that portfolio longevity is higher, and 30-year failure is less likely, if withdrawals are made annually as opposed to monthly.

5.2 Original Question

Using your monthly PLSM.12 simulation script, assume a starting portfolio value of $F = \$100$, and an annual consumption withdrawal rate of $c = 4$. Now, for the (annual) portfolio return values, assume $SD[\tilde{v}] = 15\%$, but for the expected return $E[\tilde{v}]$ assume a range of values (i.e. run five different simulations) ranging from $v = 1\%, 2\%, 3\%, 4\%, 5\%$. Then, compute the *30-year failure* rates under these five expected return assumptions and locate the value of $E[\tilde{v}]$ for which the *30-year failure* rate is closest to 90%.

Answer

I simulate 5 scenarios based off the range of values for the expected return $E[\tilde{v}] = 1\%, 2\%, 3\%, 4\%, 5\%$.

```
> Lm1<-PLSM.12(100,4,0.01,0.15,10000)
> Lm2<-PLSM.12(100,4,0.02,0.15,10000)
> Lm3<-PLSM.12(100,4,0.03,0.15,10000)
> Lm4<-PLSM.12(100,4,0.04,0.15,10000)
> Lm5<-PLSM.12(100,4,0.05,0.15,10000)
> length(Lm1[Lm1<30])/length(Lm1)
[1] 0.5619
> length(Lm2[Lm2<30])/length(Lm2)
[1] 0.4414
> length(Lm3[Lm3<30])/length(Lm3)
[1] 0.316
> length(Lm4[Lm4<30])/length(Lm4)
[1] 0.2173
> length(Lm5[Lm5<30])/length(Lm5)
[1] 0.1414
```

The higher the value of $E[\tilde{v}]$, the lower the frequency of the portfolio longevity being less than 30 years. The *30-year failure* rate is closest to 90% when $E[\tilde{v}] = 1\%$. It also makes sense intuitively that the highest failure rate will be at the worst expected market performance. For curiosity, I check the *30-year failure* rate when $E[\tilde{v}] = 0\%, -1\%$.

```
> Lm0<-PLSM.12(100,4,0,0.15,10000)
> length(Lm0[Lm0<30])/length(Lm0)
[1] 0.6737
> Lm0n<-PLSM.12(100,4,-0.01,0.15,10000)
> length(Lm0n[Lm0n<30])/length(Lm0n)
[1] 0.7761
```

I see that if $c = 4\%$ even if the market expected return is 0 or -1, the *30-year failure* rate is still below 90%. Using trial-and-error I found that for $E[\tilde{v}] = -2.6\%$ (and $SD[\tilde{v}] = 15\%$) the 30-year failure rate is indeed around 90%. This can be seen in the code below:

```
> Lm0n<-PLSM.12(100,4,-0.026,0.15,10000)
> length(Lm0n[Lm0n<30])/length(Lm0n)
[1] 0.8991
```

This means that I must expect the market to decline more than 2.6% per year, for there to be a 10% probability that the portfolio will outlast 30 years.

5.3 Original Question

Modify your basic (monthly) simulation PLSM.12 so that if-and-when the portfolio value declines by more than three standard deviations in any given month, the consumption withdrawal rate is reduced from $c/12$ to $(0.75) \times c/12$. In words, if the stock market does (very) badly in a particular month, you spend less in that particular month. Note that you will have to implement this with a separate (IF, ELSE) branch within the simulation. Now, using the $\sigma = SD[\tilde{v}] = 15\%$, and the relevant $v = E[\tilde{v}]$ value from the above problem, please compute the *30-year failure* rate in this case. It should be lower. But by how much?

Answer

To answer this question I create a new (IF, ELSE) branch in PLSM.12 which checks whether or not the portfolio value declines by more than three standard deviations in any given month. If it does, then I lower consumption for that month to $(0.75) \times c/12$. Here is the R script.

```
PLSM2.12<-function(F,c,nu,sigma,N){
  path<-matrix(nrow=N,ncol=1200)
  PLV<-c()
  for (i in 1:N){
    return<-exp(rnorm(1200,nu/12,sigma/sqrt(12)))
    path[i,1]<-F
    for (j in 2:1200){
      #check if decline exceeds three standard deviations
      if (return[j]-return[j-1]<(-3)*sigma/sqrt(12)){
        #assume lower consumption if it does
        path[i,j]<-path[i,j-1]*return[j]-(0.75*c/12)}
        #otherwise assume normal consumption
      else{path[i,j]<-path[i,j-1]*return[j]-(c/12)}
      if (path[i,j]<=0) {break}}
    PLV[i]=j/12}
  PLV}
```

I run this new function using $\sigma = SD[\tilde{v}] = 15\%$, and the $E[\tilde{v}]$ values from the previous question $v = 1\%, 2\%, 3\%, 4\%, 5\%$.

```
> Lm1<-PLSM2.12(100,4,0.01,0.15,10000)
> Lm2<-PLSM2.12(100,4,0.02,0.15,10000)
> Lm3<-PLSM2.12(100,4,0.03,0.15,10000)
> Lm4<-PLSM2.12(100,4,0.04,0.15,10000)
> Lm5<-PLSM2.12(100,4,0.05,0.15,10000)
> length(Lm1[Lm1<30])/length(Lm1)
[1] 0.5453
> length(Lm2[Lm2<30])/length(Lm2)
[1] 0.4284
> length(Lm3[Lm3<30])/length(Lm3)
[1] 0.313
> length(Lm4[Lm4<30])/length(Lm4)
[1] 0.2173
> length(Lm5[Lm5<30])/length(Lm5)
[1] 0.1399
```

The *30-year failure* rates I got are 54.53%, 42.84%, 31.3%, 21.73%, 13.99%. In the previous question, when consumption is not reduced in response to poor returns, *30-year failure* rates were 56.19%, 44.14%, 31.6%, 21.73%, 14.14% which are higher. I see only a small decline in the *30-year failure* rate compared with the last question, where the consumption withdrawal was never reduced. This is because a market decline of 3 standard deviations is very rare, so the overall consumption is almost identical between the two questions. Also, notice that for higher values of $E[\tilde{v}]$ the two functions result in increasingly similar output, because a huge market drop is less likely for higher $E[\tilde{v}]$. It could also be that consumption plays a much smaller role in financial ruin when compared with the role of investment returns. If I modify my function to reduce consumption in response to a decline of one standard deviation (instead of three), then I will see much lower *30-year failure* rates. I show this by creating another function `PLSM3.12` which does exactly that.

```
PLSM3.12<-function(F,c,nu,sigma,N){
  path<-matrix(nrow=N,ncol=1200)
  PLV<-c()
  for (i in 1:N){
    return<-exp(rnorm(1200,nu/12,sigma/sqrt(12)))
    path[i,1]<-F
    for (j in 2:1200){
      #check if decline is more than one standard deviation
      if (return[j]-return[j-1]<(-1)*sigma/sqrt(12)){
        #assume lower consumption if it is
        path[i,j]<-path[i,j-1]*return[j]-(0.75*c/12)}
      else{path[i,j]<-path[i,j-1]*return[j]-(c/12)}
      if (path[i,j]<=0) {break} }
    PLV[i]=j/12}
  PLV}
```

Here are the results.

```
> Lm1<-PLSM3.12(100,4,0.01,0.15,10000)
> Lm2<-PLSM3.12(100,4,0.02,0.15,10000)
> Lm3<-PLSM3.12(100,4,0.03,0.15,10000)
> Lm4<-PLSM3.12(100,4,0.04,0.15,10000)
> Lm5<-PLSM3.12(100,4,0.05,0.15,10000)

> length(Lm1[Lm1<30])/length(Lm1)
[1] 0.5027
> length(Lm2[Lm2<30])/length(Lm2)
[1] 0.3821
> length(Lm3[Lm3<30])/length(Lm3)
[1] 0.268
> length(Lm4[Lm4<30])/length(Lm4)
[1] 0.1774
> length(Lm5[Lm5<30])/length(Lm5)
[1] 0.1107
```

Notice that my *30-year-failure* rates are noticeably lower when using `PLSM3.12` than when using `PLSM2.12`. This is because I am lowering my consumption much more frequently in `PLSM3.12`, since declines of one standard deviation are far more prevalent than declines of three standard deviations.

5.4 Original Question

Simulate a vector of 100 historical (continuously compounded) inflation-adjusted investment returns, assuming they are normally distributed with a mean value of $\nu = 3\%$ and a standard deviation $\sigma = 15\%$. Export those numbers from **R** into a csv file using the `write.csv` command. (Google the exact syntax.) Then, import this dataset into **R**, and use those numbers with the `sample` command to bootstrap a Monte Carlo simulation for portfolio longevity.

Answer

I used the following R script to simulate a vector of 100 historical investment returns, and then bootstraped a Monte Carlo simulation for portfolio longevity. The bootstrap generates 10,000 possible portfolio trajectories by randomly selecting annual returns from the vector of 100 historical returns.

```
#simulating returns and writing to csv
write.csv(rnorm(100,0.03,0.15), file = "sample.csv")
#importing data from csv
sample <- read.csv("sample.csv")
PLSM.sample<-function(F,c,N){
  path<-matrix(nrow=N,ncol=100)
  PLV<-c()
  for (i in 1:N){
  #sampling from the imported data
    return<-exp(sample(sample[,"x"], size=100))
    path[i,1]<-F
    for (j in 2:100){
      path[i,j]<-path[i,j-1]*return[j]-c
      if (path[i,j]<=0) {break}}
    PLV[i]=j}
  PLV}
```

The results I obtained using the various `c` values can be seen below:

```
> median(PLSM.sample(100,4,10000))
[1] 37
> median(PLSM.sample(100,5,10000))
[1] 27
> median(PLSM.sample(100,6,10000))
[1] 22
> median(PLSM.sample(100,7,10000))
[1] 19
```

I ran this simulation multiple times. The results were highly inconsistent between simulations. This can be explained by the relatively small sample of only 100 values that I am using in the bootstrap. Yes, my `PLSM.sample` function is producing 10,000 trajectories, but the returns that create those trajectories come from a csv of only 100 random values. I altered the script to simulate, export to csv, and import 10,000 random historical returns, and I used them in a Monte Carlo simulation. My results were much more consistent when I repeated this experiment multiple times using using a bigger sample in the csv.

5.5 Original Question

Instead of the normal distribution with mean ν and standard deviation σ, generate portfolio longevity results, and in particular compute the *30-year failure* rate assuming a uniformly distributed (continuously compounded) return $\tilde{\nu}$, using the `runif(.)` command. Google the exact syntax, but assume monthly returns range from $\min[\tilde{\nu}] = -2\%$ to $\max[\tilde{\nu}] = 2\%$. Discuss and explain the results.

Answer

To solve this problem, I modified the code I used for normally distributed $\tilde{\nu}$ by changing `return<-exp(rnorm(100,nu,sigma))` to `return<-exp(runif(100,min,max))`:

```
#function simulates portfolio longevity
#returns distribution is assumed to be uniform(min,max)
PLSM.U<-function(F,c,N,min,max){
  path<-matrix(nrow=N,ncol=100)
  PLV<-c()
  for (i in 1:N){
    return<-exp(runif(100,min,max))
    path[i,1]<-F
    for (j in 2:100){
      path[i,j]<-path[i,j-1]*return[j]-c
      if (path[i,j]<=0) {break}}
    PLV[i]=j}
  PLV}
```

Using values of c ranging from 4 to 7, I consistently get a *30-year-failure* rate of around 100%. This makes sense because my average annual return is 0, so even with the lowest consumption of 4, my portfolio is not expected to last more than 25 years. The results can be seen below.

```
> U4<-PLSM.U(100,4,10000,-.02,.02)
> U5<-PLSM.U(100,5,10000,-.02,.02)
> U6<-PLSM.U(100,6,10000,-.02,.02)
> U7<-PLSM.U(100,7,10000,-.02,.02)
>
> length(U4[U4<30])/length(U4)
[1] 0.9992
> length(U5[U5<30])/length(U5)
[1] 1
> length(U6[U6<30])/length(U6)
[1] 1
> length(U7[U7<30])/length(U7)
[1] 1
```

If I increase the market volatility by allowing returns to be uniform between $\min[\tilde{v}] = -10\%$ to $\max[\tilde{v}] = 10\%$, I find that many portfolios will survive past the 30 year mark and my 30-year failure rates decline.

```
> U4<-PLSM.U(100,4,10000,-.1,.1)
> U5<-PLSM.U(100,5,10000,-.1,.1)
> U6<-PLSM.U(100,6,10000,-.1,.1)
> U7<-PLSM.U(100,7,10000,-.1,.1)
>
> length(U4[U4<30])/length(U4)
[1] 0.732
> length(U5[U5<30])/length(U5)
[1] 0.9736
> length(U6[U6<30])/length(U6)
[1] 0.9988
> length(U7[U7<30])/length(U7)
[1] 1
```

Chapter 6
Solutions

6.1 Original Question

Assuming the same consumption withdrawal of $c = 5$ per $F = 100$, as well as $\nu = 2.5\%$ and $\sigma = 15\%$ investment values, generate $N = 500$ instead of the (much larger) $N = 50000$ paths. Run the same `lm(.)` and `glm(.)` regressions and discuss the statistical significance of your results. Is 500 enough data points for a Monte Carlo simulation of retirement income?

Answer

Using the code from the chapter and our `PLSM.SR` function, I simulate a sample with N=500 instead of N=50000. I then run lm(.) to create a linear regression model and summarize it.

```
sample<-PLSM.SR(100,5,0.025,0.15,500)
DR1<-sample[,1]
DR2<-sample[,2]
DR3<-sample[,3]
PoLo<-sample[,4]
fit<-lm(PoLo~DR1+DR2+DR3)
summary(fit)
```

The following results were obtained:

```
Call:
lm(formula = PoLo ~ DR1 + DR2 + DR3)

Residuals:
    Min      1Q  Median      3Q     Max
-30.541  -9.673  -2.561   6.105  53.039

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  19.5691     0.9351  20.927   <2e-16 ***
DR1         356.2417    14.0681  25.323   <2e-16 ***
```

```
DR2         233.8209    14.1557  16.518   <2e-16 ***
DR3         123.0652    13.5190   9.103   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1

Residual standard error: 14.95 on 496 degrees of freedom
Multiple R-squared:  0.6602,        Adjusted R-squared:
0.6581
F-statistic: 321.2 on 3 and 496 DF,  p-value: < 2.2e-16
```

The three stars show that each of the coefficients are highly statistically significant (`p<0.001`). We saw the same thing in the chapter. The R^2 value, a measure of the regression model's fitness, is 66%, which is again the same value we got with N=50000 in the chapter.
Now I generate the logistic regression model i.e. the `glm(.)` function.

```
PoLo[PoLo<30]<-0
PoLo[PoLo>=30]<-1
fit<-glm(PoLo~DR1+DR2+DR3, family = "binomial")
summary(fit)
```

The results are:

```
Call:
glm(formula = PoLo ~ DR1 + DR2 + DR3, family = "binomial")

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.24139  -0.30456  -0.01011   0.34943   2.16503

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -3.8047     0.4081  -9.324  < 2e-16 ***
DR1          76.5500     7.6132  10.055  < 2e-16 ***
DR2          51.8129     5.7355   9.034  < 2e-16 ***
DR3          12.4913     3.3653   3.712 0.000206 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 692.95  on 499  degrees of freedom
Residual deviance: 281.81  on 496  degrees of freedom
```

```
AIC: 289.81

Number of Fisher Scoring iterations: 7
```

Here again, all three of the coefficients are statistically significant. I conclude that for this type of analysis, there is no large difference between N=500 paths and N=50000 paths, 500 data points is indeed good enough.

6.2 Original Question

Modify the basic `PLSM.SR` simulation so that you keep track of **five** 7 (instead of 10) year ARIG values, which represents investment returns during the first 35 years of consumption withdrawals. Using the same $c = 5$ per $F = 100$, and $\nu = 2.5\%$, $\sigma = 15\%$ values, run a logistic regression (only), where success is defined as surviving for 35 years. Please report and interpret the coefficients in this regression.

Answer

I generated a modified R script, which now keeps track of every 7 years. The only difference between the new script and the script from the chapter are the lines in which the $\nu\%$ values are stored. Now every 7 values get stored in a separate column in the PLM matrix. I had to extend the PLM matrix to have 6 columns, 5 for each 7-year-period, and 1 for the portfolio longevity. First, I define my new function `PLSM.SR7`.

```
PLSM.SR7<-function(F,c,nu,sigma,N){
  path<-matrix(nrow=N,ncol=100)
  PLM<-matrix(nrow=N,ncol=6)
  for (i in 1:N){
    return<-exp(rnorm(100,nu,sigma))
    PLM[i,1]<-prod(return[1:7])^(1/7)-1
    PLM[i,2]<-prod(return[8:14])^(1/7)-1
    PLM[i,3]<-prod(return[15:21])^(1/7)-1
    PLM[i,4]<-prod(return[22:28])^(1/7)-1
    PLM[i,5]<-prod(return[29:35])^(1/7)-1
    path[i,1]<-F
    for (j in 2:100){
      path[i,j]<-path[i,j-1]*return[j]-c
      if (path[i,j]<=0) {break}}
    PLM[i,6]=j}
  PLM}
```

Then I use the function and run a logistic regression on the output.

```
sample<-PLSM.SR7(100,5,0.025,0.15,50000)
DR1<-sample[,1]
DR2<-sample[,2]
DR3<-sample[,3]
DR4<-sample[,4]
```

```
DR5<-sample[,5]
PoLo<-sample[,6]
PoLo[PoLo<35]<-0
PoLo[PoLo>=35]<-1
fit<-glm(PoLo~DR1+DR2+DR3+DR4+DR5, family = "binomial")
summary(fit)
```

Here are the results:

```
Call:
glm(formula = PoLo ~ DR1 + DR2 + DR3 + DR4 + DR5,
family = "binomial")

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6161  -0.2492  -0.0281   0.1557   3.2987

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -6.3754     0.0663  -96.16   <2e-16 ***
DR1          64.3099     0.6907   93.11   <2e-16 ***
DR2          52.5995     0.5894   89.25   <2e-16 ***
DR3          33.1886     0.4390   75.60   <2e-16 ***
DR4          17.8389     0.3416   52.23   <2e-16 ***
DR5           5.2306     0.3005   17.41   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 66159  on 49999  degrees of freedom
Residual deviance: 21611  on 49994  degrees of freedom
AIC: 21623

Number of Fisher Scoring iterations: 7
```

We see all 5 coefficients are statistically significant. Notice that the first 7 years contributes 12 times more to the probability of achieving a 35 year longevity, compared to the last 7 years.

6.3 Original Question

All else being equal (that is assuming the same values of c/F, v, what happens to the coefficients in the logistic regression when you increase the volatility σ from 15% to 30%. What is the interpretation of the coefficients? Explain intuitively.

Answer

I repeat the same simulation from last question but using $\sigma = 30\%$.

```
sample<-PLSM.SR7(100,5,0.025,0.30,50000)
DR1<-sample[,1]
DR2<-sample[,2]
DR3<-sample[,3]
DR4<-sample[,4]
DR5<-sample[,5]
PoLo<-sample[,6]
PoLo[PoLo<35]<-0
PoLo[PoLo>=35]<-1
fit<-glm(PoLo~DR1+DR2+DR3+DR4+DR5, family = "binomial")
summary(fit)
```

Here is the output>

```
Call:
glm(formula = PoLo ~ DR1 + DR2 + DR3 + DR4 + DR5,
family = "binomial")

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6316  -0.2867  -0.0383   0.1975   3.3651

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.78502    0.03973  -95.27   <2e-16 ***
DR1         29.65069    0.31111   95.31   <2e-16 ***
DR2         23.72245    0.26172   90.64   <2e-16 ***
DR3         14.78597    0.19625   75.34   <2e-16 ***
DR4          8.07079    0.15957   50.58   <2e-16 ***
DR5          2.46596    0.14312   17.23   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 66907  on 49999  degrees of freedom
Residual deviance: 23576  on 49994  degrees of freedom
AIC: 23588

Number of Fisher Scoring iterations: 7
```

When the volatility is increased from 15% to 30%, the coefficients in the regression decrease to approximately half of their previous values. This is because the higher volatility/uncertainty in the investment return makes its way into the wealth path during retirement causing more overall uncertainty in any particular period. Here is a table comparing the coefficients when σ is 15% to when σ is 30%. Notice that the ratio between the coefficients does not change much when we change the market volatility (they all decline proportionally).

Coefficients in the Regression		
Variable	σ=15%	σ=30%
DR1	64.31	29.65
DR2	52.60	23.72
DR3	33.19	14.79
DR4	17.84	8.07
DR5	5.23	2.47

6.4 Original Question

Per the discussion in section 6.9, please modify the basic `PLSM.SR` simulation so that year-by-year investment returns are normally distributed but truncated and upper and lower bounds. In other words, if the random continuously compounded return exceeds 1.5σ, either above or below the mean ν, it is capped. And, while this isn't quite the option-based collar described in section 6.9, it's close enough. Please compute the portfolio 30-year success rate using the usual parameters (from the prior questions) and compare with the uncapped (a.k.a. un-collared) portfolio.

Answer

I modified the `PLSM.SR` function to create a new function `PLSM.SR.cap` which truncates return values that exceed 1.5σ, either above or below the mean ν. I used an `(IF, ELSE)` statement to check each return value, and modify when needed.

```
PLSM.SR.cap<-function(F,c,nu,sigma,N){
  path<-matrix(nrow=N,ncol=100)
  PLM<-matrix(nrow=N,ncol=4)
  for (i in 1:N){
    v<-rnorm(100,nu,sigma)
    for (q in 1:100){
      #check if return exceeds nu by 1.5*sigma
      if (v[q]>nu+1.5*sigma){
        #set return to nu+1.5*sigma
        v[q]<-nu+1.5*sigma}
      #check if return is below nu-1.5*sigma
      else if(v[q]<nu-1.5*sigma){
        #set return to nu-1.5*sigma
        v[q]<-nu-1.5*sigma}}
    return<-exp(v)
    PLM[i,1]<-prod(return[1:10])^(1/10)-1
    PLM[i,2]<-prod(return[11:20])^(1/10)-1
    PLM[i,3]<-prod(return[21:30])^(1/10)-1
    path[i,1]<-F
    for (j in 2:100){
      path[i,j]<-path[i,j-1]*return[j]-c
      if (path[i,j]<=0) {break}}
    PLM[i,4]=j}
  PLM}
```

I ran the `PLSM.SR.cap` and the `PLSM.SR`, and computed both their 30 year success rates.

```
> sample.cap<-PLSM.SR.cap(100,5,0.025,0.15,50000)
> sample<-PLSM.SR(100,5,0.025,0.15,50000)
> PoLo.cap<-sample.cap[,4]
> PoLo<-sample[,4]
>
> #Capped 30 year success
> length(PoLo.cap[PoLo.cap>30])/length(PoLo.cap)
[1] 0.45648
>
> #Uncapped 30 year success
> length(PoLo[PoLo>30])/length(PoLo)
[1] 0.45164
```

We see that the cap slightly improves the 30-year success rate. The difference is small, only around 0.5% which seems almost negligible.

6.5 Original Question Create a modified `PLSM.SR` simulation in which inflation is uniformly distributed between $[-1\%, 3\%]$ in any given year, investment returns are normally distributed with a nominal ν_n, σ, and consumption withdrawals $c = 4\%$ are adjusted every year by realized inflation. Run a simulation with
$\nu_n = 5\%, \sigma = 15\%$ and estimate coefficients using the `lm(.)` command in **R**.
Answer
I create a simulation called `PLSM.SR.inf` by modifying `PLSM.SR` as follows: For each of the N trajectories, I create an `inflation` vector with 100 uniformly distributed values (representing 100 years). I create another variable `c1` which changes every year and denotes inflation adjusted consumption of that year. For each of the N trajectories I initially set `c1=c`. Every year I redefine `c1` by multiplying last year's `c1` by `exp(inflation[i])` which is the inflation factor for that year. I then subtract `c1` instead of `c` to reduce the portfolio's value by the inflation adjusted consumption.

```
PLSM.SR.inf<-function(F,c,nu,sigma,N){
  path<-matrix(nrow=N,ncol=100)
  PLM<-matrix(nrow=N,ncol=4)
  for (i in 1:N){
    return<-exp(rnorm(100,nu,sigma))
    #generate random uniform vector of annual inflation
    inflation<-runif(100,-0.01,0.03)
    #define a new variable to track changing consumption
    #initialize that variable to equal c
    c1<-c
    PLM[i,1]<-prod(return[1:10])^(1/10)-1
    PLM[i,2]<-prod(return[11:20])^(1/10)-1
    PLM[i,3]<-prod(return[21:30])^(1/10)-1
    path[i,1]<-F
    for (j in 2:100){
      #multiply last consumption by current inflation factor
      c1<-c1*exp(inflation[j])
      path[i,j]<-path[i,j-1]*return[j]-c1
      if (path[i,j]<=0) {break}}
    PLM[i,4]=j}
  PLM}
```

I run the simulation and fit a regression model.

```
sample<-PLSM.SR.inf(100,4,0.05,0.15,50000)
DR1<-sample[,1]
DR2<-sample[,2]
DR3<-sample[,3]
PoLo<-sample[,4]
fit<-lm(PoLo~DR1+DR2+DR3)
summary(fit)
```

Here is the model summery.

```
Call:
lm(formula = PoLo ~ DR1 + DR2 + DR3)

Residuals:
    Min      1Q  Median      3Q     Max
-76.758 -14.429  -2.611  14.484  68.271

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  22.4349     0.1775  126.39   <2e-16 ***
DR1         413.5651     1.7195  240.52   <2e-16 ***
DR2         294.0551     1.7148  171.48   <2e-16 ***
DR3         165.6425     1.7152   96.57   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 19.16 on 49996 degrees of freedom
Multiple R-squared:  0.6596,       Adjusted R-squared:  0.6596
F-statistic: 3.229e+04 on 3 and 49996 DF,  p-value: < 2.2e-16
```

In the table below I compare the coefficients and intercept I got here in the ’inflation model’ with the values we got in the chapter when we used `PLSM.SR(100,5,0.025,0.15,50000)`.

Coefficients of Regression Model		
Variable	Coefficient with inflation	Coefficient in the chapter
(Intercept)	22.43	20.38
DR1	413.57	368.06
DR2	294.06	232.11
DR3	165.64	119.67

It looks like our inflation-adjusted-consumption model behaves similarly to the model in the chapter. The ratio between the coefficients is roughly the same for both models, this tells me that the sequence-of-returns risk is similar as well.

Chapter 7
Solutions

7.1 Original Question

Referring again to Figure 7.7, download the 1960 Canadian cohort (not period!) death rate vector, $q_x[1960]$, and compute the fraction of those born in the year 1960, who survived to the age of 50 (in the year 2010.) Using our notation, that is S_{50}/S_0. Compare the values to the corresponding numbers for the 1925 and 1940 cohort.

Answer

To solve this problem, I first had to download and import the 1960, 1940, and 1925 cohort death rate vectors. I imported the data from an Excel file, so your process might be a bit different if you have a txt file.

I use an expression `Q[!is.na(Q$Age),]` which gives me the rows of `Q` where `Q$Age` is not `NA`. Exclamation points are used in **R** to reverse a logical statement. The term `is.na(.)` returns true if a value is `NA` and false otherwise. So `!is.na(Q$Age)` returns a vector with `TRUE` where `Q$Age` is not `NA`, and `FALSE` where `Q$Age` is `NA`. Now `Q[!is.na(Q$Age),]` returns the rows of Q where `Q$Age` is not `NA`.

```
#load the Excel library and import the data
library(readxl)
Q<-read_excel("./HMD_CAD_q.xlsx", na = ".")
#clean up the data (remove NAs)
Q<-Q[!is.na(Q$Age),]
#create a table for each of the three cohorts
Q1960<-Q[Q$Year==1960,]
Q1925<-Q[Q$Year==1925,]
Q1940<-Q[Q$Year==1940,]
```

Once I had the death rates for each of the three cohorts, I convert them to life tables, for males and females separately.

```
#convert death rates to life table (female and male)
SF1960<-100000*cumprod(1-Q1960$Female)
SM1960<-100000*cumprod(1-Q1960$Male)
#repeat for 1940 cohort
SF1940<-100000*cumprod(1-Q1940$Female)
SM1940<-100000*cumprod(1-Q1940$Male)
#repeat for 1925 cohort
SF1925<-100000*cumprod(1-Q1925$Female)
SM1925<-100000*cumprod(1-Q1925$Male)
```

I then divide the surviving population at age 50 by 100,000 (the population at age 0). I repeat this process for the 1940 and 1925 cohort, and compare the results.

```
#get age 50 surviving fraction
#1960
> SF1960[50]/100000;SM1960[50]/100000
[1] 0.9429828
[1] 0.9060208
#1940
> SF1940[50]/100000;SM1940[50]/100000
[1] 0.8885221
[1] 0.8388726
#1925
> SF1925[50]/100000;SM1925[50]/100000
[1] 0.8197243
[1] 0.7580081
```

The results comparing the 50 year survival rate for the cohorts born in 1925, 1940, and 1960 are in the table below:

	50 year cohort survival rate		
	1925 cohort	1940 cohort	1960 cohort
Female	0.82	0.89	0.94
Male	0.76	0.84	0.91

The probability of living past 50 increases as time goes on, and women are more likely than men to survive that long.

7.2 Original Question

Estimate the slope and intercept of $\ln[q_x]$ for the 1940 cohort (both males and females), using data from the age of 35 to 70. You can either do this using the built-in least squares (a.k.a. linear model) function in **R**, called `lm(.)`, or look-up the formula for the slope and intercept (in any textbook on statistics) and compute these values using brute-force. Please discuss some of the methodological concerns with such a (small) set of numbers.

Answer

I run a regression model on the values of $\ln[q_x]$ for the 1940 cohort (both males and females). Its best not to reinvent the wheel, so I use the `lm(.)` function instead of looking up regression formulas. First I look at the male cohort.

```
fit<-lm(log(Q1940$Male)[c(35:70)]~c(35:70))
summary(fit)
```

Here is the output for the male set:

```
 Call:
lm(formula = log(Q1940$Male)[c(35:70)] ~ c(35:70))

Residuals:
     Min       1Q   Median       3Q      Max
-0.13562 -0.04202  0.01017  0.03595  0.16110

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -8.963466   0.058995 -151.94   <2e-16 ***
c(35:70)     0.070925   0.001102   64.34   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.06871 on 34 degrees of freedom
Multiple R-squared:  0.9919,       Adjusted R-squared: 0.9916
F-statistic:  4140 on 1 and 34 DF,  p-value: < 2.2e-16
```

The slope for the male linear regression model for $ln[q_x]$ is 0.071. The intercept is −8.963. Now I look at the female cohort:

```
fit<-lm(log(Q1940$Female)[c(35:70)]~c(35:70))
summary(fit)
```

Here is the output for the female set:

```
Call:
lm(formula = log(Q1940$Female)[c(35:70)] ~ c(35:70))

Residuals:
     Min       1Q   Median       3Q      Max
-0.13002 -0.03089  0.01252  0.03170  0.06324

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) -9.7160695  0.0387028  -251.0   <2e-16 ***
c(35:70)     0.0751446  0.0007232   103.9   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.04508 on 34 degrees of freedom
Multiple R-squared:  0.9969,        Adjusted R-squared: 0.9968
F-statistic: 1.08e+04 on 1 and 34 DF,  p-value: < 2.2e-16
```

The slope and intercept for $ln[q_x]$ of the female cohort is 0.075 and −9.716 respectively. I see that women have a higher slope than men but a lower intercept. This would imply that women's mortality rate is lower than men at early ages (closer to 35), but that difference narrows as they get older.

The sample size for each regression model is 35 points. This would be considered a fairly small sample, however given the very high R^2 value of over 99%, I conclude that my sample size is big enough. Another possible issue is that the model inputs are all between $x = 35$ and $x = 70$. This may be problematic when trying to extrapolate mortality rates at ages above $x = 70$.

7.3 Original Question

Forecast or estimate the fraction of the 1960 cohort (who are 50 years-old in the year 2010), who will survive to the age of 100, based on the methodology described in the chapter. How does it compare to the 1925 cohort?

Answer

Since I need to forecast survival rates in the next few questions, I create a reusable function `LT(q)` which takes in a vector of death rates and extrapolates the next 65 values, using the methodology shown in the chapter. The function then converts that extended death rate vector into a life table. I use `sum(!is.na(q))` to identify the length of the original vector, by counting non `NA` values. The expression `!is.na(q)` returns a vector of `TRUE` and `FALSE` values. The `TRUE` values appear where the `q` vector has non `NA` values. `TRUE` is treated as the number 1 and `FALSE` as 0, so `sum(!is.na(q))` is the number of non `NA` values in the `q` vector.

```
#this function takes a vector of death rates q
#runs regression and extrapolates life table forecast
LT<-function(q){
  #identify last available death rate
  l<-sum(!is.na(q))
  #run regression on 36 to 70
  fit<-lm(log(q)[c(36:70)]~c(36:70))
  #get regression slope
  m<-fit$coefficients[2]
  #forecast 65 death rates after last value
  q.new<-q[l]*exp(c(1:65)*m)
  #append forecasted values to original death rate vector
  q<-append(q,q.new,after=l)
  #convert death rates to life table
  LT<-100000*cumprod(1-q)
  return(LT)}
```

Notice that I begin the regression on the 36th value because that is the death rate of people between their 35th and 36th birthday. I can now use this function to estimate what fraction of those alive in 2010 will survive to age 100.

```
> LT(Q1960$Female)[101]/LT(Q1960$Female)[51]
[1] 0.2525011
> LT(Q1960$Male)[101]/LT(Q1960$Male)[51]
[1] 0.2753618
> LT(Q1940$Female)[101]/LT(Q1940$Female)[71]
[1] 0.2239078
> LT(Q1940$Male)[101]/LT(Q1940$Male)[71]
[1] 0.114978
> LT(Q1925$Female)[101]/LT(Q1925$Female)[86]
[1] 0.09768422
> LT(Q1925$Male)[101]/LT(Q1925$Male)[86]
[1] 0.03903571
```

The results are summarized in the table below:

Survival to age 100 conditional on being alive in 2010			
	1925 cohort	1940 cohort	1960 cohort
Female	0.10	0.22	0.25
Male	0.04	0.11	0.28

I see that the more recent the cohort's birth year, the more likely they are to live to 100. I also see that for the 1960 cohort, the percentage of 50 year old males that will survive to age 100 is higher than that of the females. This is consistant with what I saw in the previous question that females have a higher slope for thier log mortality rates, but a lower intercept.

7.4 Original Question

Please estimate values of S_{90}/S_{60}, which is the 30-year survival rate conditional on being age $x = 60$, for the 1925, 1940 and 1960 cohort.

Answer

For this question I use the life-table function I generated in the last question. For each cohort, I divide the fraction of survivors to age 90 by the fraction of survivors to age 60.

```
> LT(Q1960$Female)[91]/LT(Q1960$Female)[61]
[1] 0.567688
> LT(Q1960$Male)[91]/LT(Q1960$Male)[61]
```

```
[1] 0.5468121
> LT(Q1940$Female)[91]/LT(Q1940$Female)[61]
[1] 0.5042549
> LT(Q1940$Male)[91]/LT(Q1940$Male)[61]
[1] 0.3577371
> LT(Q1925$Female)[91]/LT(Q1925$Female)[61]
[1] 0.3446002
> LT(Q1925$Male)[91]/LT(Q1925$Male)[61]
[1] 0.183985
```

I summarized the results in the table below.

	S_{90}/S_{60}		
	1925 cohort	1940 cohort	1960 cohort
Female	0.34	0.50	0.57
Male	0.18	0.36	0.55

I see that the 30-year survival rate conditional on being age $x = 60$ is always higher for women than for men, and increases for later birth cohorts.

7.5 Original Question

Pick one other country (anyone) from the Human Mortality Database (HMD), which should all look quite similar to Figure 7.7, and download one year (cohort) death rates, create a 1940 cohort life table, and compare projected survival rates S_{90}/S_{60}, conditional on age 60.

Answer

Following the same steps as question 7.1, I downloaded the 1940 cohort death rates from Australia.

```
library(readxl)
Q1940_A <- read_excel("./HMD_AUS_q.xlsx",na=".")
```

I then use the `LT(q)` function I created in question 7.3, and obtain the projected conditional survival rates similarly to question 7.4.

```
> #Australia 1940 female and male cohorts
> LT(Q1940_A$Female)[91]/LT(Q1940_A$Female)[61]
[1] 0.5575774
> LT(Q1940_A$Male)[91]/LT(Q1940_A$Male)[61]
[1] 0.3911386
> #Canada 1940 female and male cohorts
> LT(Q1940$Female)[91]/LT(Q1940$Female)[61]
[1] 0.5042549
> LT(Q1940$Male)[91]/LT(Q1940$Male)[61]
[1] 0.3577371
```

My results are summarized in the table below:

1940 Cohort 30 Year Survival Rates (from age 60 to 90)		
	Canada	Australia
Female	0.50	0.56
Male	0.36	0.39

I see that 60 year old Australians born in 1940 are more likely to survive past age 90 than their Canadian counterparts.

Chapter 8
Solutions

8.1 Original Question

Simulate $N = 10,000$ random lives using the Gompertz model conditional on age $x = 40$, with parameters ($m = 85, b = 10$) and compute the fraction of this group who die between the ages of $y = 75$ and $y = 95$. Please repeat this experiment five (5) times to obtain five (5) different estimates of this number. Finally, compare those numbers to the analytic result using the `TPXG(.)` function. Discuss.

Answer

Correction: The part of this question in the textbook "with plus/minus one dispersion coefficient (not standard deviation)" is a mistake and was removed in the solution manual above.

In this question I compare the results of using a simulation to generate random lives using the Gompertz model and the analytic results using the `TPXG(.)` function. The **R** script can be seen below. First I generate $10,000$ random lifetimes (conditional on age 40) using the `GRAN(.)` function and the parameters given in the question. I tally how many of the results are between 35 and 55 and divide by $10,000$ (the group size) to get the fraction of the group that die between ages 75 and 95. I put this script inside a loop that runs five times. I can also choose to just re-run the middle three lines five times and avoid looping.

```
#repeat 5 times
for (i in 1:5){
  #generate a sample of 10,000 random lifetimes
  z<-GRAN(10000,40,85,10)
  #count the fraction that die between age 75 and 95
  y<-length(z[(z>=35) & (z<=55)])/10000
  #print results
  print(y)}
```

I will present the output along with the analytical solution in a table shortly. Remember that since this is a random simulation, your simulation results will probably vary slightly.

To solve the question analytically, I use the `TPXG(.)` function and subtract the probability of dying past 95 from the probability of dying past 75, this gives me the fraction expected to die between those two ages:

```
#probability of surviving past 75
analytic75<-round(TPXG(40,35,85,10),4)
#probability of surviving past 95
analytic95<-round(TPXG(40,55,85,10),4)
#probability of dying between 75 and 95
f<-analytic75-analytic95
```

The results are summarized in the table below:

Fraction of group who die between 75 and 95	
Trial	Fraction
GRAN1	0.6327
GRAN2	0.6352
GRAN3	0.6207
GRAN4	0.6338
GRAN5	0.6357
TPXG	0.6332

It looks like the results of running 10,000 simulations are very close to the analytic result. I conclude that simulation results are a good proxy for the real value.

8.2 Original Question

Use the simulation "trick" (inverting the survival probability p, and extracting t) that I described in section #8.5 of this chapter, to create a script that simulates exponential (i.e. not Gompertz) remaining lifetimes, with parameter λ. Now use that script to simulate $N = 10,000$, with $\lambda = 1/50$, which is a mortality hazard rate of 2%. Compute the fraction of random lives that died between time $T = 35$ and time $T = 55$, and compare with the analytic expression. Discuss.

Answer

Exponential lifetimes have a constant mortality hazard rate so the age x is not relevant. Here is the survival probability equation for surviving another t years (at any age).

$$\theta(t) = e^{-\lambda t}$$

Inverting this equation gives me:

$$t = \frac{ln(\theta)}{-\lambda}$$

I generate a sample of 10,000 possible θ values, using runif, that are uniformly distributed between 0 and 1, and extract 10,000 t values from that sample, using the above equation. I then measure what percentage of t values are between 35 and 55.

```
> #function simulates N exponential remaining lifetimes
> ERAN<-function(N,lambda){log(runif(N))/(-lambda)}
> #simulate sample
> z<-ERAN(10000,0.02)
> #measure fraction that dies between age 35 and 55
> length(z[(z>=35) & (z<=55)])/10000
[1] 0.1647
```

I see the analytic expression result is very close.

```
> #analytic expression P(T>35)-P(T>55)
> exp(-0.02*35)-exp(-0.02*55)
[1] 0.1637142
```

The results for exponential remaining lifetimes are around 16% which is much lower than the 63% I got using a Gompertz lifetime in the previous question.

Note: Under an exponential remaining lifetime assumption, the probability of a person age zero surviving to 35 is identical to the probability of a person age 40 surviving to 75. The exponential assumption is age agnostic, it does not care how old I am, I am always equally likely to die within a given time period. This is due to the memoryless property of exponential distribution, which you can look up online.

8.3 Original Question

Jack is 50, Jill is 45 and they have a daughter Joanne, who is 25. What is the probability that Joanne will outlive both Jack and Jill? Assume they all "die" according to the Gompertz law, with $m = 80$ for Jack, $m = 90$ for Jill and $m = 95$ for Joanne, and assume that $b = 10$ for all three of them. Please use simulation based techniques if you can't figure-out and solve the relevant 3-dimensional integral.

Answer

Given that I am already using simulations throughout this chapter, it seems like a more natural route than solving the integral. I use the GRAN(.) function to simulate 10,000 random ages of death for Jack, Jill and Joanne.

```
#create Jack sample
zdat.Jack<-(50+GRAN(10000,50,80,10))
#create Jill sample
zdat.Jill<-(45+GRAN(10000,49,90,10))
#create Joanne sample
zdat.Joanne<-(25+GRAN(10000,25,95,10))
```

Next, I write a script which counts the number of times where Joanne's age of death is greater than both Jack and Jill's. I do this by creating a new variable called `outliveboth` which is originally set to zero and gets incremented by one whenever Joanne's age of death is greater than both Jack and Jill.

```
#initialize counter
outliveboth<-0
#loop through sample
for (i in 1:10000){
  #check if Joanne outlives both parents
  if (zdat.Joanne[i]>=zdat.Jack[i] && zdat.Joanne[i]>=
      zdat.Jill[i]){
    #increment counter if she does
    outliveboth<-outliveboth+1}}
```

I then divide `outliveboth` by my total sample size of 10,000 to get the probability of Joanne outliving both her parents.

```
> #get fraction where Joanne outlives parents
> outliveboth/10000
[1] 0.6449
```

I see that there is about a 64% chance Joanne will outlive both Jack and Jill.

8.4
Original Question
Simulate two samples of $N = 10,000$ random Gompertz lifetimes, conditional on age $x = 40$, but one with $(m = 75, b = 13)$ and the other with $(m = 95, b = 7)$, and combine them into one (larger) vector of 20,000 numbers. Now, create one cohort life table (yes, for this combined mixture), extract the one q_x vector, from the age of $x = 40$ to the age of $x = 100$. Plot the log one-year death rate (similar to Figure 8.5), and run a simple linear regression model, recall the `fit<-lm(.)` command, to estimate the slope of log one-year death rate. How good is the fit over the range $(40, 100)$? How good is the fit over the range $(70, 100)$? Discuss your results, and compare with Figure 8.5.
Answer
To solve this problem, first I create the two samples of $N = 10,000$ random Gompertz lifetimes, and combine them into one larger vector.

```
#create sample 1
a<-GRAN(10000,40,75,13)
#create sample 2
b<-GRAN(10000,40,95,7)
#combine the sample
zdat<-append(b,a,after=10000)
```

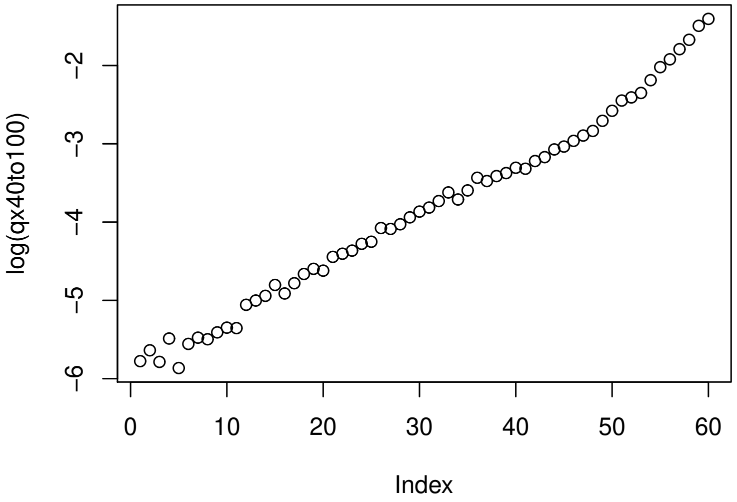
I convert the lifetime samples into a cohort life table and extract the q_x vector.

```
#create cohort life table
zdat.CLT<-LTLD(zdat)
#isolate life table values for x=40 to x=100
z1.CLT<-zdat.CLT[1:61]
#extract qx vector from life table
qx40to100<-(-diff(z1.CLT))/z1.CLT[-length(z1.CLT)]
```

Next, I plot $ln(q_x)$.

```
#plotting ln(q_x)
plot(log(qx40to100))
```

The figure below shows the resulting plot from the regression over x=40 to x=100.



Now I create the linear model and evaluate the coefficients and goodness of fit.

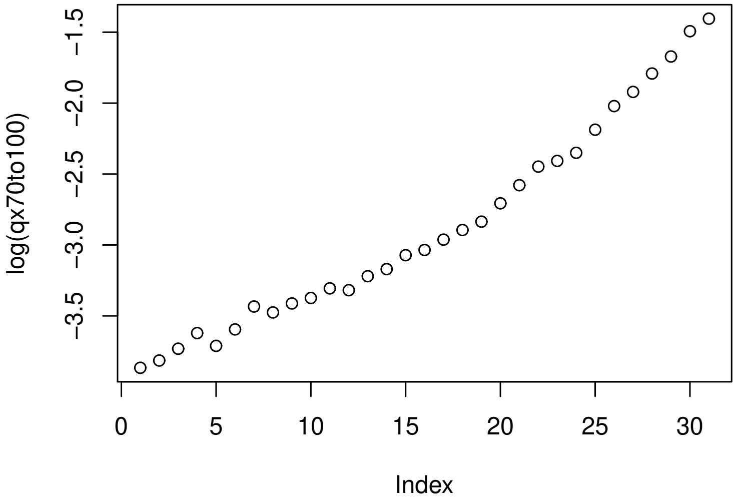
```
> #run simple linear regression
> fit<-lm(log(qx40to100)~c(1:60))
> #get intercept and slope
> fit$coefficients
(Intercept)     c(1:60)
```

```
 -5.9626410   0.0693883
> #get goodness of fit
> summary(fit)$r.squared
[1] 0.9874493
```

I repeat this process over the range (70,100).

```
> #repeat with x=70 to x=100
> z2.CLT<-zdat.CLT[31:61]
> qx70to100<-(-diff(z2.CLT))/z2.CLT[-length(z2.CLT)]
> #plotting ln(q_x)
> plot(log(qx70to100))
> #run simple linear regression
> fit<-lm(log(qx70to100)~c(1:30))
> #get intercept and slope
> fit$coefficients
(Intercept)     c(1:30)
-4.05723294  0.07941847
> #get goodness of fit
> summary(fit)$r.squared
[1] 0.9718257
```

The figure below shows the resulting plot from the regression over x=70 to x=100.



The (40,100) range has a slightly better fit than the (70,100) range, as reflected in the higher R-squared value. This makes sense since the (70,100) range sample starts with fewer people, as many have died between age 40 and 70, so the sample is smaller and hence there is more variance. Both linear regression models are statistically significant, and have very similar slopes and intercepts, with the 70 to 100 having a slightly higher slope and intercept. When I compare these two models and plots with the model in the chapter in Figure 8.5, I see that Figure 8.5 has a better linear fit and also has a higher R-squared value of 0.9987. At first I thought this can be explained by the larger sample size in the chapter of 100,000 versus 20,000 here. I reran the script here using two samples of 100,000 and the goodness of fit did not improve. Then I tried simulating my blended sample at age zero and looking at 120 years of mortality rates instead of sixty years, but the goodness of fit was still not as good as the 0.9987 obtained in the chapter. Finally I tried moving the modal `m` values closer to each other from (75,95) to (84,86). That did the trick and raised my R-squared values close to 0.998. I conclude that combining two Gompertz distributed samples does not result in a Gompertz distribution if the original two samples have different modal parameters.

8.5 Original Question

Finally, some finance and economics. Imagine that you are $x = 40$ years-old and offered a *retirement income* “product” that pays you \$100, 000 (in real, inflation adjusted dollars) if-and-when you reach that age of $y = 80$. And, if you never reach that age, you get nothing. Assume that long-term (continuously compounded) interest rate is 1.25%, so the present value (for 40 years) of this \$100, 000 is approximately \$60, 000. Of course, there is a chance you will not survive to age 80, so you shouldn’t be charged \$60, 000. So, please use the `TPXG(.)` function to estimate the survival probability, and use that information to “value” this product. As far as parameter estimates are concerned, solve this problem assuming two cases. The (what I would call) *sick male* one, with $(m = 75, b = 13)$ and the *healthy female* one with $(m = 95, b = 7)$.

Answer

To determine the value of the product, I first calculate the survival probabilities for the two cases using the `TPXG(.)` function. I then determine the expected value of the product by multiplying the survival probability by \$60, 000, which is the present value of the survival contingent payment. Note that this product is a lump sum bullet payment, not an annuity.

```
> #survival probablity for sick male
> tpx_m<-TPXG(40,40,75,13)
> #survival probabilty for healthy female
> tpx_f<-TPXG(40,40,95,7)
> #product value for sick male
> 60000*tpx_m
[1] 14776.23
> #product value for sick female
> 60000*tpx_f
[1] 53378.73
```

The value of the product is $53, 378.73 for the healthy female and and $14, 776.23 for the unhealthy male. It makes sense that healthy people who are more likely to live longer will have greater benefit and value from their retirement products.

Chapter 9
Solutions

9.1 Original Question

Simulate $N = 10,000$ random Gompertz lifetimes, with a modal value of $(m = 88)$ and dispersion value of $(b = 10)$, as well as random portfolio longevity values with $\nu = 2.5\%$ and $\sigma = 15\%$. Call these vectors zdat.hl and zdat.pl. Compute the lifetime ruin probability φ, for initial withdrawal rates of $\xi = 4\%$ using the method explained in the chapter. Very simple. But in addition, please compute the *expected number of shortfall years*, in the event you are alive when the money runs out. How? Think very carefully about the dots under the diagonal in Figure 9.4, and then measure the magnitude by which L_{ξ} falls short of T_x.

Answer

Similarly to what was done in the chapter, I simulate 10,000 random Gompertz lifetimes with the provided parameters, as well as 10,000 random portfolio longevity values.

```
#simulate Gompertz random lifetimes
zdat.hl<-GRAN(10000,65,88,10)
#simulate random portfolio longevity values
zdat.pl<-PLSM(1,0.04,0.025,0.15,10000)
```

To compute the lifetime ruin probability, I iterate through the random Gompertz lifetime vector zdat.hl and the random portfolio longevity vector zdat.pl, and increment the ruined variable to keep count of every time the zdat.hl value is greater than the zdat.pl value. For every time the zdat.hl value is greater than the zdat.pl value, I add the years by which it is greater to the shortfall variable to get the total number of shortfall years.

```
#initialize count of ruined
ruined<-0
#initialize shortfall years
shortfall<-0
#loop through lifetime and portfolio samples
for (i in 1:10000){
  #if lifetime exceeds portfolio longevity
  if(zdat.hl[i]>=zdat.pl[i]){
    #increment count of ruined
    ruined<-ruined+1
    #add shortfall years to total shortfall count
    shortfall<-shortfall+(zdat.hl[i]-zdat.pl[i])}}
```

I divide the count of ruined individuals by the sample size to get the lifetime probability of ruin.

```
> #lifetime ruin probability
> ruined/length(zdat.hl)
[1] 0.1486
```

I divide the total number of shortfall years by the number of ruined individuals to get the expected number of shortfall years. Note that the expected number of shortfall years is conditioned on being ruined, because I am dividing the total number of shortfall years by the number of ruined people, and not by the total sample size of 10,000.

```
> #expected number of shortfall years
> shortfall/ruined
[1] 6.67663
```

The lifetime ruin probability φ, for an initial withdrawal rate of $\xi = 4\%$ is 14.86%, and the expected number of shortfall years is 6.68. Note, I assumed that $x = 65$, if I assumed a younger age of retirement, then the ruin probability and expected shortfall would be higher.

9.2 Original Question
Going back to the `LRPG(.)` function, locate the spending rate ξ that leads to *at most* a 3% lifetime ruin probability at retirement ages $x = 55, 65, 75$, under a $\nu = 2.5\%$ real rate of return assumption, $\sigma = 15\%$, and Gompertz ($m = 88, b = 10$).
Answer
I create a function that takes in an age x and calculates the highest ξ that results in a lifetime ruin probability below 3% for that given age. The function begins by creating a placeholder for the solution `xi_top`, and setting it to zero. It then loops through possible values of ξ between 0% and 10% in ascending order. If a value of ξ results in a lifetime ruin probability below 3%, it replaces `xi_top` with that ξ. The final value of `xi_top` is the highest ξ that has a ruin probability below 3%.

```
#function returns highest allowed xi for a given age
highest_spending<-function(age){
  #initialize maximum xi value at 0
  xi_top<-0
  #loop through increasing xi values
  for (xi in 1:10000*.00001){
    #if the ruin probability is below 3%
    if (LRPG(0.025,xi,age,88,10)<0.03){
      #replace xi_top with the latest xi
      xi_top<-xi}}
  print(xi_top)}
```

I run my function for ages 55,65,75.

```
> highest_spending(55)
[1] 0.03673
> highest_spending(65)
[1] 0.04225
> highest_spending(75)
[1] 0.05189
```

The results are summarized in the table below.

Approximate Spending Rate Leading to 3% LRP	
Retirement Age	Initial Withdrawal Rate
55	3.67%
65	4.22%
75	5.19%

If I retire at 55 then I can spend at most 3.67% of my initial retirement wealth each year, assuming I want to keep my probability of ruin below 3%. If I wait till age 75 to retire, then I can spend up to 5.19% a year.

9.3 Original Question
I introduced three different estimates for the lifetime ruin probability, denoted by $\varphi := \Pr[T_x \geq L_\xi]$. The first revolved around simulating `zdat.pl`, for portfolio longevity and `zdat.hl` for human longevity and comparing vectors. The second method revolved around simulating the stochastic present value of the wealth trajectory until the time of death, and counting the number of negative values via the `VARPHI.SM` function. The third and final method was based on the moment matching approximation to the stochastic present value, which (although rushed) was captured in the `VARPHI.MM` function. Compare the accuracy (and speed) of these methods by creating a table of φ values for initial withdrawal rates: $\xi = 2\%, 4\%, 6\%$, assuming two pairs: $(\nu = 2.5\%, \sigma = 10\%)$ and $(\nu = 4\%, \sigma = 25\%)$. Discuss results.
Answer
I begin by writing a function `method1(.)` that estimates φ using the first method similarly to question 9.1. For the second and third method, I use the functions defined in the chapter.

```
#create simulation function
method1<-function(N,x,m,b,xi,nu,sigma){
  #simulate random lifetimes
  zdat.hl<-GRAN(N,x,m,b)
  #simulate random portfolio longevities
  zdat.pl<-PLSM(1,xi,nu,sigma,N)
  #fraction where lifetime exceeds portfolio longevity
  return(sum(zdat.hl>=zdat.pl)/length(zdat.hl))}
```

Now, I estimate the lifetime ruin probabilities for the initial withdrawal rates $\xi = 2\%, 4\%, 6\%$ using all three methods with $N = 1,000,000$, $\nu = 0.025$, $\sigma = 0.1$, $x = 55$, $m = 88$ and $b = 10$. I assume the first method that simulates and compares the values of `zdat.pl` and `zdat.hl` is the most accurate since I have a very large sample size. I measure the speed of these methods by recording the start system-time before I run the simulation, and then taking the difference between the system-time right after it finishes running. System-time is a reading of the computer's internal clock, and is a very useful tool for measuring run time.

```
#set sample size and initialize parameters
N<-1000000; x<-55; m<-88; b<-10;
nu<-0.025; sigma<-0.1
#loop though our xi values
for (xi in c(0.02,0.04,0.06)){
  #record start time
  tic<-Sys.time()
  #run first method
  m1<-round(method1(N,x,m,b,xi,nu,sigma),5)
  #measure run time
  toc<-round(Sys.time()-tic,5)
  #print parameter, method name, varphi, and run time
  print(c(’xi=’,xi,’m1=’,m1,toc))
  #record start time
  tic<-Sys.time()
  #run second method
  m2<-round(VARPHI.SM(N,x,m,b,xi,nu,sigma),5)
  #measure run time
  toc<-round(Sys.time()-tic,5)
  #print parameter, method name, varphi, and run time
  print(c(’xi=’,xi,’m2=’,m2,toc))
  #record start time
  tic<-Sys.time()
  #run third method
  m3<-round(VARPHI.MM(x,m,b,xi,nu,sigma),5)
  #measure run time
  toc<-round(Sys.time()-tic,5)
  #print parameter, method name, varphi, and run time
  print(c('xi=',xi,'m3=',m3,toc))}
```

Here is the output for $\nu = 0.025$ and $\sigma = 0.1$. The last number in each row is the time (in seconds) it took for that method to run.

```
[1] "xi="      "0.02"     "m1="      "0.00889"  "41.99483"
[1] "xi="     "0.02"    "m2="     "0.01076" "4.18273"
[1] "xi="     "0.02"    "m3="     "0.01571" "0.01561"
[1] "xi="      "0.04"     "m1="      "0.24787"  "31.24326"
[1] "xi="     "0.04"    "m2="     "0.28423" "3.99255"
[1] "xi="     "0.04"    "m3="     "0.24883" "0"
[1] "xi="      "0.06"     "m1="      "0.62079"  "24.04333"
[1] "xi="     "0.06"    "m2="     "0.66854" "3.9703"
[1] "xi="     "0.06"    "m3="     "0.63393" "0"
```

I then repeat the process but with $\nu = 0.04$ and $\sigma = 0.25$

```
N<-1000000; x<-55; m<-88; b<-10;
nu<-0.04; sigma<-0.25
for (xi in c(0.02,0.04,0.06)){
  tic<-Sys.time()
  m1<-round(method1(N,x,m,b,xi,nu,sigma),5)
  toc<-round(Sys.time()-tic,5)
  print(c('xi=',xi,'m1=',m1,toc))
  tic<-Sys.time()
  m2<-round(VARPHI.SM(N,x,m,b,xi,nu,sigma),5)
  toc<-round(Sys.time()-tic,5)
  print(c('xi=',xi,'m2=',m2,toc))
  tic<-Sys.time()
  m3<-round(VARPHI.MM(x,m,b,xi,nu,sigma),5)
  toc<-round(Sys.time()-tic,5)
  print(c('xi=',xi,'m3=',m3,toc))}
```

Here is the output for $\nu = 0.04$ and $\sigma = 0.25$.

```
[1] "xi="      "0.02"     "m1="      "0.08787"  "41.26497"
[1] "xi="     "0.02"    "m2="     "0.09496" "3.95648"
[1] "xi="     "0.02"    "m3="     "0.08467" "0.1024"
[1] "xi="      "0.04"     "m1="      "0.28652"  "33.11393"
[1] "xi="     "0.04"    "m2="     "0.30728" "3.92965"
[1] "xi="    "0.04"   "m3="    "0.3067" "0"
[1] "xi="      "0.06"     "m1="      "0.47909"  "29.99073"
[1] "xi="     "0.06"    "m2="     "0.51032" "3.89696"
[1] "xi="     "0.06"    "m3="     "0.53719" "0"
```

The resulting φ estimates are summarized in the tables below.

LRP values for $\nu = 2.5\%$ and $\sigma = 10\%$

ξ	Method1	VARPHI.SM	VARPHI.MM
2%	0.00889	0.01076	0.01571
4%	0.24787	0.28423	0.24883
6%	0.62079	0.66854	0.63393

LRP values for $\nu = 4\%$ and $\sigma = 25\%$

ξ	Method1	VARPHI.SM	VARPHI.MM
2%	0.08787	0.09496	0.08467
4%	0.28652	0.30728	0.3067
6%	0.47909	0.51032	0.53719

I see the second and third method have similar accuracy, as they differ from the first method by roughly the same amount. The speed of these methods differ greatly. The first method is about ten times slower than the second method, and the third method is virtually instantaneous, as it does not use simulation. Notice that as ξ increases, the lifetime ruin probability increases.

9.4 Original Question

Advanced: Modify the `VARPHI.SM` from ($\Delta t = 1/52$) to ($\Delta t = 1/12$), and compare numerical results for standard (x, m, b) and (ξ, ν, σ) values.

Answer

I create a new function `VARPHI.SM.12` where I change the ($\Delta t = 1/52$) value to ($\Delta t = 1/12$) by replacing 52 with 12 in the `VARPHI.SM` function.

```
VARPHI.SM.12<-function(N,x,m,b,xi,nu,sigma){
  V<-c()
  #replaced 52 with 12 and renamed 'wks' to 'months'
  months<-round(GRAN(N,x,m,b)*12,0)+1
  for (i in 1:N){
    #replaced 52 with 12 and 'wks' with 'months'
    sB<-sigma*sqrt(1/12)*cumsum(rnorm(months[i]))
    #replaced 52 with 12 and 'wks' with 'months'
    Z<-exp((nu/12)*c(1:(months[i]))+sB)
    #replaced 52 with 12
    V[i]<-sum((1/Z)/12)}
  sum(V>=1/xi)/length(V)}
```

The results of both the `VARPHI.SM` and `VARPHI.SM.12` functions, taking various parameters can be seen below.

```
> VARPHI.SM(10000,65,88,10,0.04,0.025,0.10)
[1] 0.1143
```

```
> VARPHI.SM.12(10000,65,88,10,0.04,0.025,0.10)
[1] 0.1168
>
> VARPHI.SM(10000,65,88,10,0.04,0.04,0.25)
[1] 0.1915
> VARPHI.SM.12(10000,65,88,10,0.04,0.04,0.25)
[1] 0.1864
>
> VARPHI.SM(10000,65,88,10,0.05,0.025,0.10)
[1] 0.262
> VARPHI.SM.12(10000,65,88,10,0.05,0.025,0.10)
[1] 0.2665
>
> VARPHI.SM(10000,65,88,10,0.05,0.04,0.25)
[1] 0.2874
> VARPHI.SM.12(10000,65,88,10,0.05,0.04,0.25)
[1] 0.2827
>
> VARPHI.SM(10000,65,88,10,0.06,0.025,0.10)
[1] 0.4252
> VARPHI.SM.12(10000,65,88,10,0.06,0.025,0.10)
[1] 0.4261
>
> VARPHI.SM(10000,65,88,10,0.06,0.04,0.25)
[1] 0.3699
> VARPHI.SM.12(10000,65,88,10,0.06,0.04,0.25)
[1] 0.3772
```

It looks like the monthly and weekly models yield roughly the same results.

9.5 Original Question

Advanced: Modify the PLSM simulation from yearly to monthly and compare results with the answers to question 9.4.

Answer

I use the PLSM.12 function I created in question 5.1.

```
PLSM.12<-function(F,c,nu,sigma,N){
  path<-matrix(nrow=N,ncol=1200)
  PLV<-c()
  for (i in 1:N){
    return<-exp(rnorm(1200,nu/12,sigma/sqrt(12)))
    path[i,1]<-F
    for (j in 2:1200){
      path[i,j]<-path[i,j-1]*return[j]-(c/12)
      if (path[i,j]<=0) {break}}
  PLV[i]=j/12}
  PLV}
```

I create a new function called `method1.12` which is the same as the `method1` function from question 9.3 but I replace `PLSM` with `PLSM.12`.

```
#create simulation function
method1.12<-function(N,x,m,b,xi,nu,sigma){
  #simulate random lifetimes
  zdat.hl<-GRAN(N,x,m,b)
  #simulate random portfolio longevities USING MONTHLY MODEL
  zdat.pl<-PLSM.12(1,xi,nu,sigma,N)
  #fraction where lifetime exceeds portfolio longevity
  return(sum(zdat.hl>=zdat.pl)/length(zdat.hl))}
```

I now run `method1.12` alongside `VARPHI.SM.12` and `VARPHI.SM` like in question 9.4.

```
> VARPHI.SM(10000,65,88,10,0.04,0.025,0.10)
[1] 0.1151
> VARPHI.SM.12(10000,65,88,10,0.04,0.025,0.10)
[1] 0.1231
> method1.12(10000,65,88,10,0.04,0.025,0.10)
[1] 0.1131
>
> VARPHI.SM(10000,65,88,10,0.04,0.04,0.25)
[1] 0.1936
> VARPHI.SM.12(10000,65,88,10,0.04,0.04,0.25)
[1] 0.1917
> method1.12(10000,65,88,10,0.04,0.04,0.25)
[1] 0.1967
>
> VARPHI.SM(10000,65,88,10,0.05,0.025,0.10)
[1] 0.2632
> VARPHI.SM.12(10000,65,88,10,0.05,0.025,0.10)
[1] 0.269
> method1.12(10000,65,88,10,0.05,0.025,0.10)
[1] 0.2589
>
> VARPHI.SM(10000,65,88,10,0.05,0.04,0.25)
[1] 0.2828
> VARPHI.SM.12(10000,65,88,10,0.05,0.04,0.25)
[1] 0.2857
> method1.12(10000,65,88,10,0.05,0.04,0.25)
[1] 0.2774
```

```
>
> VARPHI.SM(10000,65,88,10,0.06,0.025,0.10)
[1] 0.4254
> VARPHI.SM.12(10000,65,88,10,0.06,0.025,0.10)
[1] 0.427
> method1.12(10000,65,88,10,0.06,0.025,0.10)
[1] 0.4227
>
> VARPHI.SM(10000,65,88,10,0.06,0.04,0.25)
[1] 0.3652
> VARPHI.SM.12(10000,65,88,10,0.06,0.04,0.25)
[1] 0.3723
> method1.12(10000,65,88,10,0.06,0.04,0.25)
[1] 0.3615
```

It appears that `method1.12` usually results in slightly lower ruin probabilities than the other two functions tested here, yet still very close.

Chapter 10
Solutions

10.1 Original Question

Please use integration techniques described in this chapter to value an *immediate* life annuity (factor) at the age of $x = 50$, that pays \$1 of income for life until age $y = 85$, but then if-and-when you reach the age of $y = 85$, the payment is reduced by 50% to only \$0.50, from that point onward. Assume the usual ($m = 88$, $b = 10$) Gompertz parameters, and valuation rates of $v = 1\%$ as well as $v = 5\%$. How does the value (or annuity factor) compare to a basic, simple *immediate* annuity that pays \$1 for life?

Answer

To solve this problem, I use the integration technique for valuing an immediate life annuity but I split the integral into two parts, one from 0 to 35, and another from 35 to infinity. For the first 35 years (from age 50 to 85) \$1 is paid annually, and after that \$0.50 is paid every year.

```
> #v=0.01
> #set parameters
> x<-50;y<-35;m<-88;b<-10;v<-0.01
> #define function to integrate
> integrand<-function(t){exp(-v*t+exp((x-m)/b)*(1-exp(t/b)))}
> #integrate over first 35 years
> a<-integrate(integrand,0,y)$value
> #integrate over 35 to infinity and multiply by 0.5
> b<-(integrate(integrand,y,Inf)$value)*0.5
> #add both parts to get total value
> a+b
[1] 26.59798
```

The value of the annuity with the deferred 50% reduction is \$26.60 under a 1% valuation rate. Below is the value of the basic annuity with no reduction, under the same valuation rate.

```
> #value of original annuity
> integrate(integrand,0,Inf)$value
[1] 31.1367
```

I see that reducing the payout to $0.50 per year after age 85 lowers the value of the annuity from $31.14 to $26.60, which is a reduction of about 15%.

I repeat this process for $v = 5\%$.

```
> #v=0.05
> x<-50;y<-35;m<-88;b<-10;v<-0.05
> #define function to integrate
> integrand<-function(t){exp(-v*t+exp((x-m)/b)*(1-exp(t/b)))}
> #integrate over first 35 years
> a<-integrate(integrand,0,y)$value
> #integrate over 35 to infinity and multiply by 0.5
> b<-(integrate(integrand,y,Inf)$value)*0.5
> #add both parts to get total value
> a+b
[1] 15.24508
```

And here is the value of the regular annuity under the same 5% valuation rate.

```
> #value of original annuity
> integrate(integrand,0,Inf)$value
[1] 16.98737
```

Under a 5% valuation rate, the new product is only 10% cheaper than the basic annuity. Here is a table summarizing the values.

	Value of Immediate Life Annuity		
	Basic Annuity	New Product	Price Difference
$v = 0.01$	31.14	26.60	15% Reduction
$v = 0.05$	16.99	15.25	10% Reduction

I see that at higher valuation rates, the new product's value is closer to that of the basic annuity. This makes sense intuitively, since the value of payments in 35 years is much lower when I assume higher valuation rates, so cutting those distant payments in half has a smaller effect on the overall value.

10.2 Original Question

At the very start of this chapter, I simulated Gompertz lifetimes and effectively valued an *immediate* life annuity by computing the average `RGOA(.)` value for each of those $N = 10,000$ random lifetimes. That was another way of computing the actuarial present value (APV). Using the same idea, please use simulation techniques to compute the APV of a *deferred* annuity, issued at age $x = 50$, which begins paying at age $y = 80$, and assuming a $\rho = 1\%$ cost of living adjustment. In other words, I would like you to replicate the payout numbers displayed in Table 10.1, assuming the same Gompertz parameters.

Answer

I loop through the values of $m = 84, 88, 92, 96$ and simulate 10,000 random lifetimes for each m value using our `GRAN(.)` function. First I calculate the expected age at death for each m by averaging the remaining lifetimes and adding 50. Note that expected age at death is not used to calculate the value of the deferred annuity, but it does appear in table 10.1, so I calculate it here.

Next I compute the APV of the deferred annuity. First I subtract 30 from all the remaining random lifetimes, and replace any resulting negative numbers with zeros to get the number of years the deferred annuity actually pays each person. Now I loop through the valuation rates $1\%, 2\%, 3\%, 4\%$ and take the average present value of the annuity deferred by thirty years, which I calculate using the `RGOA(.)` function divided by $(1 + v)^{30}$. I divide $\$1,000$ by the average present value of the annuity, and then divide the result by 12, to get the monthly payments. Here is the code.

```
#loop through modal values
for (m in c(84,88,92,96)){
  #simulate 10000 Gompertz random lifetimes
  zdat<-GRAN(10000,50,m,10)
  #expected age at death
  print(mean(zdat)+50)
  #number of years receiving payout
  zdat<-zdat-30; zdat[zdat<0]<-0
  #loop through interest rates
  for (v in c(0.01,0.02,0.03,0.04)){
    #average pv of deferred payments
    pv<-mean(RGOA(0.01,v,zdat)/(1+v)^30)
    #monthly income
    m.income<-(1000/pv)/12
    print(c(m,v,m.income))}}
```

Here is the output which I use to replicate Table 10.1.

```
[1] 79.61198
[1] 84.000  0.010 27.042
[1] 84.00000  0.02000 38.54984
[1] 84.00000  0.03000 54.67208
[1] 84.00000  0.04000 77.15196
[1] 83.05014
[1] 88.00000  0.01000 17.69942
```

```
[1] 88.0000  0.0200 25.5126
[1] 88.00000  0.03000 36.55918
[1] 88.00000  0.04000 52.09365
[1] 87.00995
[1] 92.00000  0.01000 12.16734
[1] 92.00000  0.02000 17.75628
[1] 92.00000  0.03000 25.73651
[1] 92.00000  0.04000 37.06116
[1] 90.76901
[1] 96.000000  0.010000  9.121442
[1] 96.00000  0.02000 13.49968
[1] 96.00000  0.03000 19.81988
[1] 96.00000  0.04000 28.87807
```

Deferred: monthly income at 80 per $1000 premiums at 50				
Gompertz mode	$m = 84$	$m = 88$	$m = 92$	$m = 96$
$E[T_{50}] + 50$	79.61	83.05	87.01	90.77
$v = 1\%$	$27.04	$17.70	$12.17	$9.12
$v = 2\%$	$38.55	$25.51	$17.76	$13.50
$v = 3\%$	$54.67	$36.56	$25.74	$19.82
$v = 4\%$	$77.15	$52.09	$37.06	$28.88

My results are close to those in the chapter.

10.3 Original Question
The derivation in Section 10.10 provides an explicit expression for the value of an *immediate* life annuity under an (expanded) Gompertz-Makeham law. Assuming ($m = 88, b = 10$), please value an immediate annuity at age $x = 65$, under $v = 1\%$ and $v = 3\%$, assuming that Makeham constant $\lambda = 3/1000$. This means that every year *three-out-of-a-thousand* people die accidentally. So, how much of a difference does accidental deaths have on annuity valuations at advanced ages?
Answer
I code the explicit expression from the chapter as a function in **R**. First I must define the Gamma function.

```
#define gamma function
G<-function(alpha,beta){
  integrand<-function(t){(t^(alpha-1)*exp(-t))}
  integrate(integrand,beta,Inf)$value}
```

Now I can define my annuity factor function `a(x,m,b,r,u,l)`. Note that I replaced v with r, so that my variable names match the form of the expression in the chapter. I also use l as a shorthand for λ (lambda), the Greek equivalent to l.

```
#annuity factor under Gompertz-Makeham
a<-function(x,m,b,r,u,l){
  #standardize age
  eta<-exp((x-m)/b)
  #define Gamma function parameters
  alpha<-(-(r+l)*b)
  beta<-eta*exp(u/b)
  #set denominator of function
  denominator<- (eta^(-(r+l)*b))*exp(-eta)
  #put it all together
  b*G(alpha,beta)/denominator}
```

I now run this function with the parameters from the question using $v = 1\%$ and $v = 3\%$.

```
> # v=1%
> a(65,88,10,0.01,0,3/1000)
[1] 17.28952
> # v=3%
> a(65,88,10,0.03,0,3/1000)
[1] 13.97794
```

I see that the increase in valuation rate makes the annuity cheaper, because it lowers the value of future payments. Now I check how much of an impact accidental death has on the value of the annuity, by dividing the annuity factor when $\lambda = 3/1000$ by the annuity factor when $\lambda = 0$. I subtract this fraction from 1 to get the percentage decrease caused by accidental death.

```
> # cost discount from accidental death (v=1%)
> 1-a(65,88,10,0.01,0,3/1000)/a(65,88,10,0.01,0,0)
[1] 0.03348231
> # cost discount from accidental death (v=3%)
> 1-a(65,88,10,0.03,0,3/1000)/a(65,88,10,0.03,0,0)
[1] 0.02989935
```

The assumption that 3 in 1000 people die accidentally every year reduces the value of the immediate life annuity by about 3%. For a higher valuation rate, accidental death has less impact on the value of the annuity.

10.4 Original Question
Value a 100% joint life annuity, under a $v = 3\%$ valuation rate, when $x_1 = 65$, $x_2 = 60$, but assuming that $m = 88$ and $b = 10$ for both lives. (I.e. they are homogeneous lives.) Can you simplify the expression for the annuity value?
Answer
First I solve this problem using the ING function from the chapter.

```
x1<-65;m1<-88;b1<-10
x2<-60;m2<-88;b2<-10
v<-0.03
ING<-function(t){
  exp(-v*t)*(exp(exp((x1-m1)/b1)*(1-exp(t/b1)))+
             exp(exp((x2-m2)/b2)*(1-exp(t/b2)))-
             exp(exp((x1-m1)/b1)*(1-exp(t/b1)))*
             exp(exp((x2-m2)/b2)*(1-exp(t/b2))))}
```

I integrate `ING` from zero to infinity to get the value of the joint life annuity.

```
> integrate(ING,0,Inf)$value
[1] 18.50969
```

The value of this 100% joint life annuity is \$18.51 per \$1 of lifetime income.

Now to simplify the expression, I replace `m1` and `m2` with `m`, and `b1` and `b2` with `b`. I also name the parts of the function that are reused multiple times so that the expression is much shorter.

```
ING1<-function(t){
  z<-(1-exp(t/b))
  a<-exp(exp((x1-m)/b)*z)
  b<-exp(exp((x2-m)/b)*z)
  exp(-v*t)*(a+b-a*b)}
```

Integrating this simpler function yields the same result, as expected.

```
> integrate(ING1,0,Inf)$value
[1] 18.50969
```

10.5 Original Question

In the above "problem", assume the couple request a 15-year period certain on their life annuity. What is the annuity factor in that case?

Answer

I use the ING function from the previous question, along with the script from the chapter in section 10.8 to add a 15-year period certain.

```
> x1<-65;m1<-88;b1<-10
> x2<-60;m2<-88;b2<-10
> v<-0.03;PC<-15
> (1/v)*(1-exp(-v*PC))+integrate(ING,PC,Inf)$value
[1] 18.65357
```

The resulting annuity factor is $18.65. Note that the annuity factor only increases by 0.14 when I add a 15-year period certain, because the probability of **both** people dying within 15 years is very small.

Chapter 11
Solutions

11.1 Original Question

How would you modify the IDDR(.) algorithm to account for a (married) couple for which the Gompertz parameters are (m_1, b_1) and (m_2, b_2) respectively. Assume that their time and risk preferences (ρ, γ) are identical, and they desire an *intelligent draw down* strategy that continues until the second member of the couple dies. To be clear, the pension continues at full 100% of π until the second person dies.

Answer

To solve this problem I must replace the individual survival probability in the IDDR(.) function with the couple's joint survival probability. Survival probability appears three times in the IDDR(.) function, once explicitly in the k3 term as TPXG(.), and twice it is embedded within the GTLA(.) function. I replace GTLA(.) with GTLA.2(.) which is the actuarial present value of a joint temporary life annuity. The new GTLA.2(.) function integrates the ING(.) function similarly to what was done in section 10.9 in the previous chapter.

```
#see section 10.9 in chapter
GTLA.2<-function(x1,x2,m1,m2,b1,b2,r,tau){
  ING<-function(t){
    exp(-r*t)*(exp(exp((x1-m1)/b1)*(1-exp(t/b1)))+
                 exp(exp((x2-m2)/b2)*(1-exp(t/b2)))-
                 exp(exp((x1-m1)/b1)*(1-exp(t/b1)))*
                 exp(exp((x2-m2)/b2)*(1-exp(t/b2))))}
  integrate(ING,0,tau)$value}
```

The tPx.2(.) function which replaces TPXG(.) uses the joint survival probability portion of the ING(.) function.

```
#see section 10.9 in chapter
tPx.2<-function(x1,x2,m1,m2,b1,b2,t){
  exp(exp((x1-m1)/b1)*(1-exp(t/b1)))+
     exp(exp((x2-m2)/b2)*(1-exp(t/b2)))-
     exp(exp((x1-m1)/b1)*(1-exp(t/b1)))*
     exp(exp((x2-m2)/b2)*(1-exp(t/b2)))}
```

Now I have everything I need to modify `IDDR(.)` and create `IDDR.2(.)`.

```
#here is the resulting function
IDDR.2<-function(Fx,pi,x1,x2,m1,m2,b1,b2,r,rho,gam){
  k<-(r-rho)/gam
  WDT<-function(tau){
    K1<-((Fx/pi+1/r)*exp(r*tau)-1/r)/
      (
      #replace GTLA with GTLA.2
      GTLA.2(x1-b1*log(gam),x2-b2*log(gam)
             ,m1,m2,b1,b2,r-k,tau)
       *exp(r*tau))
    K2<-exp(k*tau)
    #replace TPXG with tPx.2
    K3<-tPx.2(x1,x2,m1,m2,b1,b2,tau)^(1/gam)
    K1*K2*K3-1}
  tau<-uniroot(WDT,lower=0,upper=100)$root
  Cx<-((Fx+pi/r)*exp(r*tau)-pi/r)/
    #replace GTLA with GTLA.2
    (GTLA.2(x1-b1*log(gam),x2-b2*log(gam),m1,m2,b1,b2,r-k,tau)
     *exp(r*tau))
  (Cx-pi)/Fx}
```

I take my new function for a test drive and compare its output for various scenarios.

```
> #base test
> IDDR.2(100,0.04,65,60,85,90,12,8,0.02,0.03,4)
[1] 0.03797715
> #reduce longevity risk aversion
> IDDR.2(100,0.04,65,60,85,90,12,8,0.02,0.03,2)
[1] 0.04401157
> #raise pension
> IDDR.2(100,0.06,65,60,85,90,12,8,0.02,0.03,2)
[1] 0.04408307
> #lower rho
> IDDR.2(100,0.04,65,60,85,90,12,8,0.02,0.01,2)
[1] 0.03757515
```

Reducing my couple's longevity risk aversion raised their spending. Increasing their pension slightly raised their non pension spending as well. Lowering their subjective discount factor ρ reduced their spending. All of these results are consistent with economic and financial intuition that reducing fear of running out of money or raising the availability of funds through pension will cause spending to rise, and reducing the utility drop resulting from postponing spending will make spending now less valuable in comparison, and result in more saving and less spending now.

11.2 Original Question

Assuming that F_{65} = \$100 at the age of $x = 65$, with Gompertz parameter ($m = 88$, and $r = 2\%$, please investigate the relationship between the dispersion coefficient b, the pre-existing pension π and the *intelligent drawdown* rate `IDDR(.)`.

Answer

To solve this problem, I create three vectors, each with a different dispersion coefficient b. For each vector I iterate through different pension values ranging from 1 to 100 in increments of 10. At each iteration I apply the `IDDR(.)` function and appended the results to the respective vector.

```
#Pension values to loop through
s<-c(1,10,20,30,40,50,60,70,80,90,100)
p1<-c()
for (i in s){
  #b=5
  p1<-append(p1,IDDR(100,i,65,88,5,0.02,0.02,1)*100)}
p2<-c()
for (i in s){
  #b=10
  p2<-append(p2,IDDR(100,i,65,88,10,0.02,0.02,1)*100)}
p3<-c()
for (i in s){
  #b=15
  p3<-append(p3,IDDR(100,i,65,88,15,0.02,0.02,1)*100)}
```

Here is the output.

```
> p1
 [1]  6.377066  7.793895  8.726438  9.468860 10.074357
 [6] 10.546692 11.080017 11.427953 11.918449 12.295531
[11] 12.651097
> p2
 [1]  6.794214  9.352943 11.128742 12.482910 13.649742
 [6] 14.679244 15.702202 16.577952 17.258080 18.001769
[11] 18.889284
> p3
 [1]  6.640543  9.668014 11.753737 13.388118 14.831763
 [6] 16.001966 17.207753 18.250559 19.222611 20.137462
[11] 21.004174
```

The results displaying the relationship between the dispersion coefficient b, the pre-existing pension π, and the intelligent drawdown rate `IDDR(.)` can be seen in the table below.

IDDR(.) values for various π and b			
π	$b = 5$	$b = 10$	$b = 15$
1	6.37	6.79	6.64
10	7.79	9.35	9.67
20	8.73	11.13	11.75
30	9.47	12.48	13.39
40	10.07	13.65	14.83
50	10.55	14.68	16.00
60	11.08	15.70	17.21
70	11.43	16.58	18.25
80	11.92	17.26	19.22
90	12.30	18.00	20.14
100	12.65	18.89	21.00

I see that the higher the pension the stronger the impact of b. Similarly the higher b, the greater the impact of raising pension. This makes sense, because the more pension I will get, the more comfortable I am spending my non-pension money even if I am uncertain about my longevity. i.e. higher values of π make me less worried about running out of money, so if I add higher values of b on top of big pension, now the main worry is dying early with plenty of money, so I must spend more.

11.3 Original Question

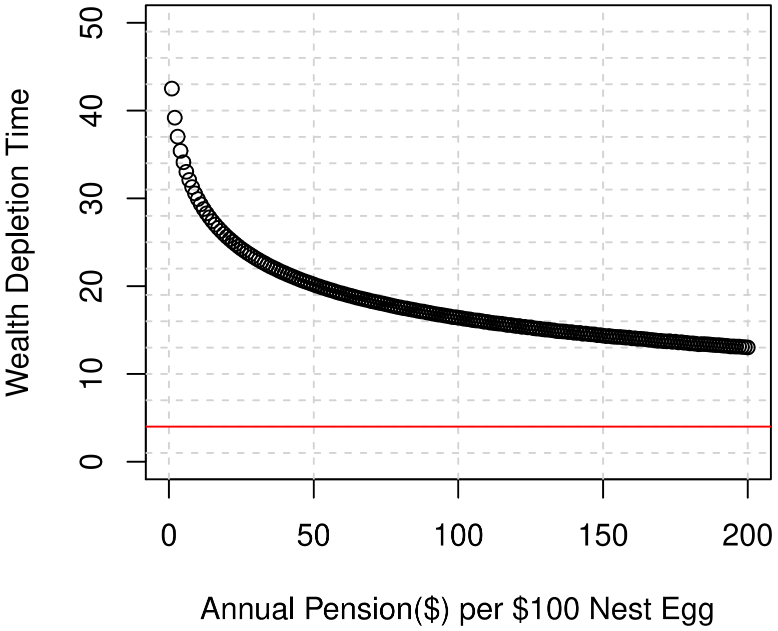
The wealth depletion time is a function that is embedded inside the `IDDR(.)` function. Assuming that $F = \$100$, that $x = 65$, ($m = 88, b = 10$) and $r = \rho = 2\%, \gamma = 4$, please plot WDT as a function of $\pi = \$1$ to $\pi = \$200$. Describe how the wealth depletion time is impacted by the pre-existing pension income. Explain.

Answer

I generate an **R** script which plots WDT as a function of $\pi = \$1$ to $\pi = \$200$.

```
#define WDT function
WDT<-function(tau){
  K1<-((Fx/pi+1/r)*exp(r*tau)-1/r)/
    (GTLA(x-b*log(gam),tau,r-k,m,b)*exp(r*tau))
  K2<-exp(k*tau)
  K3<-TPXG(x,tau,m,b)^(1/gam)
  K1*K2*K3-1}
#define parameters
Fx<-100;x<-65;m<-88;b<-10;r<-0.02;rho<-0.02;gam<-4
k<-(r-rho)/gam
#plot the roots of WDT which is wealth depletion time
plot(c(0,200),c(0,50),type="n",xlab="Annual Pension($)
per $100 Nest Egg",
     ylab="Wealth Depletion Time")
grid(ny=18,lty = 20)
for (i in 1:200){
  pi<-i
  points(i,uniroot(WDT,lower = 1,upper = 100)$root,
  col="black",pch=1)}
abline(h=4,col="red")
```

The plot that is generated can be seen in the figure below.



Based on the figure, it is clear that the higher the pension, the earlier I can deplete my liquid assets, since the higher pension is guaranteed to provide for me.

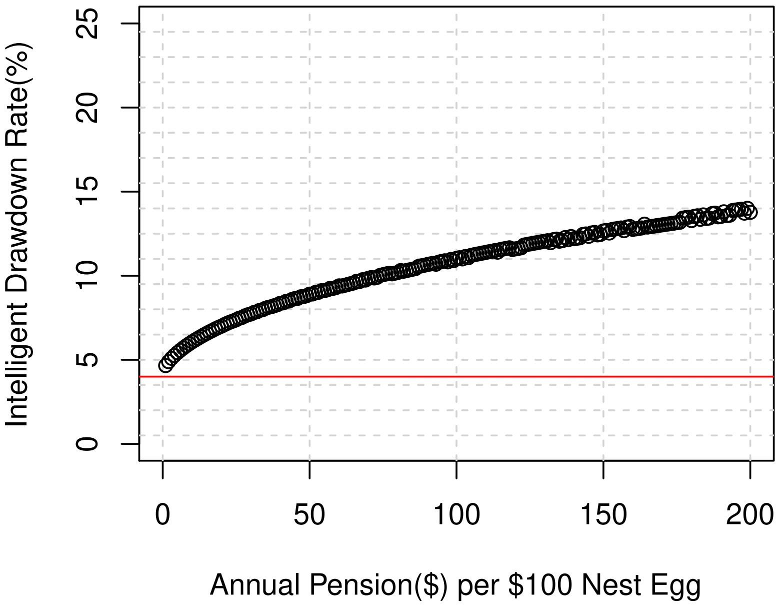
11.4 Original Question

Assume that you are at age $x = 65$, that you have $F_x = \$100$, that $r = \rho = 2\%, \gamma = 4$, and $(m = 88, b = 10)$. Plot the relationship between π and the *intelligent drawdown* rate, within the range of $\pi = \$1$ to $\pi = \$200$.

Answer
I create a script in **R** to plot the relationship between π and the *intelligent drawdown* rate, within the range of π = \$1 to π = \$200.

```
plot(c(0,200),c(0,25),type="n",xlab="Annual Pension($)
per $100 Nest Egg",
     ylab="Intelligent Drawdown Rate(%)")
grid(ny=18,lty=20)
for(i in 1:200){
  points(i,IDDR(100,i,65,88,10,0.02,0.02,4)*100,
  col="black",pch=1)}
abline(h=4,col="red")
```

The plot that is generated can be seen in the following figure.



Increasing annual pension raises the intelligent drawdown rate, similarly to the previous question. When I have more pension, I spend more from my liquid assets.

11.5 Original Question

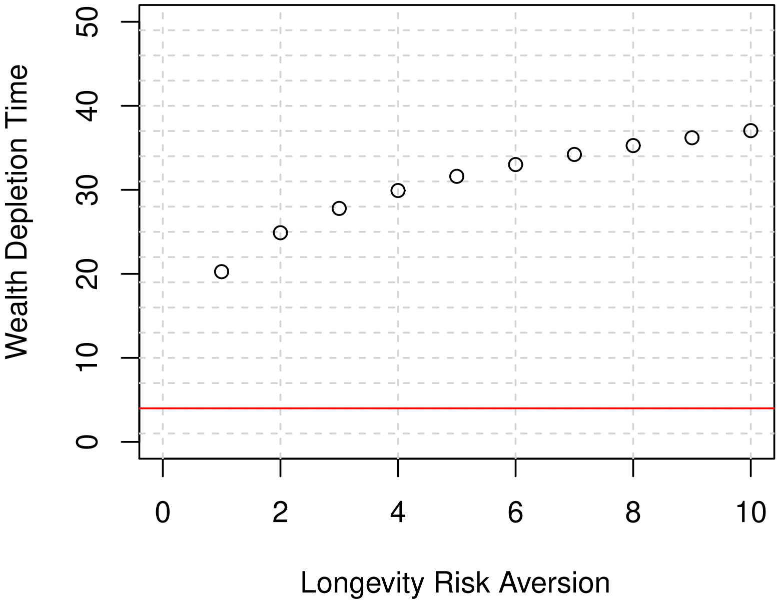
Assuming that $F = \$100$, that $x = 65$, $(m = 88, b = 10)$ and $r = \rho = 2\%, \pi = 10$, please plot WDT as a function of $\gamma = 1$ to $\gamma = 10$. Describe how the wealth depletion time is impacted by longevity risk aversion. Explain.

Answer

I plot WDT as a function of $\gamma = 1$ to $\gamma = 10$, using the parameters above.

```
Fx<-100; x<-65; m<-88; b<-10; r<-0.02; rho<-0.02; pi<-10
plot(c(0,10),c(0,50),type="n",xlab="Longevity Risk
Aversion",ylab="
     Wealth Depletion Time")
grid(ny=18,lty=20)
for(i in 1:10){
  gam=i
  points(i,uniroot(WDT,lower=1,upper=100)$root,
  col="black",pch=1)}
abline(h=4,col="red")
```

Based on this script the following figure was generated.



I see that increasing my longevity risk aversion raises my wealth depletion time, since it causes me to reduce spending.

Chapter 12
Solutions

12.1 Original Question

Compute the contribution rate, s^*, for a copper-plated pension plan (CPP versus GPP) that replaces 35% of your *consumable* income. In other words, if your real *disposable* income is \$100, and the contribution rate is s^*, the CPP would provide you with $(0.35) \times \$100 \times (1 - s^*)$ for your entire life, starting at retirement age R. Assume that you enter the system at age $x = 25$, contribute for 40 years until age $R = 65$, and the plan provides a pension annuity until death, which is modeled as a Gompertz variable with $(m = 88, b = 10)$. Provide s^* numbers for $r = 1\%$ and $r = 4\%$. Note that wage growth g is zero.

Answer

Similarly to section 12.2 from the chapter, I use the formula which calculates the collective GPP contribution rate, but now I multiply the `GDLA` term by 0.35 to account for the 35% of consumable income which the CPP provides.

$$s^* \times GTLA = 0.35 \times (1 - s^*) \times GDLA$$

$$s^* \times GTLA + 0.35 \times s^* \times GDLA = 0.35 \times GDLA$$

$$s^* = \frac{0.35 \times GDLA}{GTLA + 0.35 \times GDLA}$$

I set my parameters and calculate s^* under a 1% valuation rate.

```
> x<-25;R<-65;m<-88;b<-10;r<-0.01
> 0.35*GDLA(x,R,r,m,b)/
   (GTLA(x,R-x,r,m,b)+0.35*GDLA(x,R,r,m,b))
[1] 0.1052559
```

I see that I must save 10.5% of my annual income to afford a copper-plated pension plan, this allows me to spend 89.5% of my income, assuming the valuation rate is 1%. What happens if I assume a 4% valuation rate?

```
> r<-0.04
> 0.35*GDLA(x,R,r,m,b)/
  (GTLA(x,R-x,r,m,b) +0.35*GDLA(x,R,r,m,b))
[1] 0.0407009
```

Under a 4% valuation rate I only need to save 4.1% of my income compared to the 10.5% of income I must save at a 1% valuation rate. Assuming a higher valuation rate reduces the required pension contribution rate.

12.2 Original Question

Building on the CPP described in the prior question, i.e. a plan that replaces 35% of your consumable income, now assume wage growth $g > 0\%$ per year. In other words, at retirement your wage is $\$100 \times (1+g)^{R-x}$, and the CPP replaces 35% of that final wage times $(1 - s^*)$. Remember (again) that your consumable income is *net* of the CPP contributions. Compute s^* values for $r = 1\%$ and $r = 4\%$.

Answer

The main difference between this question and the previous question is that here the temporary life annuity that reflects wages is an increasing annuity, and the deferred annuity that reflects pension is a multiple of higher annual wage (the final wage before retirement). To model the increasing wages, I create a modified version of GTLA which I call GTLA.growth. This returns the actuarial present value of a temporary life annuity that grows at a rate of g every year. I do this by multiplying the value of the rgrid by $(1+g)^t$.

```
GTLA.growth<-function(x,tau,v,m,b,g){
  if (tau==0){0}
  else {
    dt<-1/52; grid<-ceiling(tau/dt); t<-(1:grid)*dt
    pgrid<-exp(exp((x-m)/b)*(1-exp(t/b)))
    #multiply rgrid by growth factor
    rgrid<-exp(-v*t)*(1+g)^t
    sum(pgrid*rgrid)*dt}}
```

I multiply GDLA by $0.35 \times (1+g)^{R-x}$, since the pension payments will be $35\% \times (1 - s^*)$ of the final wage. Now I can compute s^* values for $r = 1\%$ and $r = 4\%$ using multiple values of g.

```
#define parameters
x<-25;R<-65;m<-88;b<-10
#loop through r values
for (r in c(0.01,0.04)){
  #loop through g values
  for (g in c(0.005,0.01,0.015,0.02)){
    #calculate s
    s<-0.35*GDLA(x,R,r,m,b)*(1+g)^(R-x)/
      (GTLA.growth(x,R-x,r,m,b,g) +
        0.35*GDLA(x,R,r,m,b)*(1+g)^(R-x))
    print(c(s,r,g))}}
```

In the output, the first number in each row is the value of s^*, the second number is the valuation rate, and the third number is the income growth rate g.

```
[1] 0.1156581 0.0100000 0.0050000
[1] 0.1265248 0.0100000 0.0100000
[1] 0.137809 0.010000 0.015000
[1] 0.1494587 0.0100000 0.0200000
[1] 0.04585933 0.04000000 0.00500000
[1] 0.05146229 0.04000000 0.01000000
[1] 0.05751381 0.04000000 0.01500000
[1] 0.06401272 0.04000000 0.02000000
```

I see that raising g increases the percentage of income I must save each year. This is intuitive since g has a bigger impact on the deferred pension income than it does on my annual income. All my pension payments get multiplied by $(1+g)^{R-x}$, but my annual income payments are multiplied by lower powers of $(1+g)$. If I want to grow my pension by more than the growth of my wages, then I must save more. Note, the pension income does not grow by g every year, it just starts at a higher level and stays there.

12.3 Original Question
Assume (hypothetically) that you are $x = 65$ years-old, have $F = \$100$ in liquid wealth and are entitled to a pension annuity of $\pi = \$50$. What level of longevity risk-aversion γ will lead to a wealth depletion time (WDT) of exactly $\tau = 20$ years, that is you live on your pension (only) if you get to age $(x + \tau) = 85$. For this question, assume that $r = \rho = 4\%$ and then do it again for $r = \rho = 1\%$. Explain the difference.
Answer
I start by calculating the fraction of my balance sheet that is pensioned ψ. I then create my own function myfunc, which only takes one parameter γ. Now myfunc uses the function WDT.PSI from the chapter, to calculate wealth depletion time as a function of ψ and subtracts 20 from it. Note that WDT.PSI uses the other previously defined parameters, but myfunc only takes in γ. I designed myfunc to only take γ as a parameter so that I can use uniroot to find the value of γ that makes myfunc zero.

```
#set parameters
Fx<-100; pi<-50; x<-65;m<-88; b<-10;
r<-0.04; rho<-0.04;
```

```
#define psi
psi<-GILA(x,r,m,b)/(Fx/pi+GILA(x,r,m,b))
#define a function of gamma that returns WDT-20
myfunc<-function(gam){
  WDT.PSI(psi,x,m,b,r,rho,gam)-20}
```

I find the root of myfunc, which is the value of γ that makes WDT equal 20.

```
#find gamma value that makes myfunc = 0
> uniroot(myfunc,lower=1,upper=10)$root
[1] 3.410369
```

I see that for $r = \rho = 4\%$ my wealth depletion time is 20 years if my longevity risk aversion value is 3.4. Now I repeat the process for $r = \rho = 1\%$.

```
#set parameters
r<-0.01; rho<-0.01;
#define psi
psi<-GILA(x,r,m,b)/(Fx/pi+GILA(x,r,m,b))
#define a function of gamma that gives WDT-20
myfunc<-function(gam){
  WDT.PSI(psi,x,m,b,r,rho,gam)-20}
```

```
#find gamma value that makes myfunc = 0
> uniroot(myfunc,lower=1,upper=10)$root
[1] 4.157554
```

Under a 1% valuation rate, I must have a γ value of 4.2 in order for my wealth depletion time to be 20 years. I see that for a lower r and ρ I require higher longevity risk aversion to deplete my wealth at the same time. This makes sense because lower valuation rates increase the value of the pension, thus increasing ψ. A higher ψ reduces wealth depletion time, so a higher γ is required to keep the same wealth depletion time.

12.4 Original Question
Assume that you are $x = 40$ years old, plan to retire at age $R = 70$, and haven't yet saved anything for retirement, so that $F_{40} = \$0$. You are trying to create a personal gold-plated pension plan, and trying to figure out how much to contribute, s^*, so that you will have flat and constant consumption until age $D = 95$. Assume that $r = 4\%$. Now, what would that contribution rate be, if you became part of a collective (DB) pension plan in which the average participant lives to age $D = 90$. If you earn $w = \$100{,}000$ per year, how much more would you be able to consume (per year) if you joined such a program? Assume no growth, so that $g = 0\%$, for simplicity.
Answer

To solve for s^*, under the personal gold-plated pension plan, I use equation 12.2 from the chapter.

$$s^* = \frac{\texttt{RGOA}(\texttt{0},\texttt{r},\texttt{N2})}{\texttt{RGOA}(\texttt{0},\texttt{r},\texttt{N1}) \times (1+r)^{\texttt{N1}} + \texttt{RGOA}(\texttt{0},\texttt{r},\texttt{N2})}$$

I set my parameters and code this equation.

```
x<-40;R<-70;D<-95;r<-0.04;w<-100000;g<-0;
s1<-RGOA(g,r,D-R)/(RGOA(g,r,R-x)*(1+r)^N1+RGOA(g,r,D-R))
```

Now assuming I became part of a collective (DB) pension plan in which the average participant lives to age $D = 90$, for this I use equation 12.4 from the chapter.

$$s^* = \frac{\texttt{GDLA}(.)}{\texttt{GDLA}(.) + \texttt{GTLA}(.)}$$

Setting the Gompertz parameters and coding this equation.

```
m<-90;b<-10;
s2<-GDLA(x,R,r,m,b)/(GTLA(x,R-x,r,m,b)+GDLA(x,R,r,m,b))
```

Now I look at the the resulting contribution rate s^* from both of these scenarios and the difference in how much I can consume each year.

```
> s1
[1] 0.2178598
> s2
[1] 0.1570675
> (s1-s2)*w
[1] 6079.237
```

I see that under a personal pension plan I must save 22% of my annual income, while in a collective plan I only need to save 16%. The collective pension plan allows me to spend an extra $6,079 a year.

12.5 Original Question
Solve the above question (12.4), assuming that your wage grows (in real terms) by $\rho = 1\%$. How does the solution and answer change?
Answer
If I maintain a flat and constant consumption while my wages are growing, then the percentage of my wages that I save (or spend) will differ every year. To deal with this, I will calculate my flat consumption value which will be constant across all years. For the personal pension, I can use the `RGOA` function. I divide the increasing annuity factor representing the present value of my future wages, by the ordinary annuity factor representing a fixed level of consumption until age 95.

```
> x<-40;R<-70;D<-95;r<-0.04;w<-100000;g<-0.01
> c1<-RGOA(g,r,R-x)/RGOA(0,r,D-x)
> c1*w
[1] 88996.59
> 1-c1
[1] 0.1100341
```

Under a personal pension plan I can spend $88,997 a year until I am 95. In my first year working I must save 11% of my income. To see how this compares to the collective pension plan I use my `GTLA.growth` function defined in question 12.2.

```
GTLA.growth<-function(x,tau,v,m,b,g){
  if (tau==0){0}
  else {
    dt<-1/52; grid<-ceiling(tau/dt); t<-(1:grid)*dt
    pgrid<-exp(exp((x-m)/b)*(1-exp(t/b)))
    #multiply rgrid by growth factor
    rgrid<-exp(-v*t)*(1+g)^t
    sum(pgrid*rgrid)*dt}}
```

I set my Gompertz parameters and divide the actuarial present value of my wages, by the lifetime annuity factor representing my lifetime consumption.

```
> m<-90;b<-10;
> c2<-GTLA.growth(x,R-x,r,m,b,g)/GILA(x,r,m,b)
> c2*w
[1] 95166.72
> 1-c2
[1] 0.04833279
```

If I were part of a collective pension plan then I would be able to consume $95, 167 a year, and in my first year I would need to save 4.8% of my salary. Now I look at the difference between the personal plan and the collective plan.

```
> w*(c1-c2)
[1] -6170.128
```

I see that the collective pension plan allows me to spend $6, 170 more each year than what I could spend on a personal pension plan. When I compare this to the previous question I see that the growth in income does not change the benefit I get from switching to a collective plan, which is around $6, 100 in both cases. The income growth does however require me to save less for retirement, regardless of whether I take the personal pension or collective pension route. Note that in this question I assumed smooth lifetime consumption, unlike in question 12.2 where consumption grew each year until retirement, and where I saved a fixed percentage of my annual income and spent the rest.

Chapter 13
Solutions

13.1 Original Question

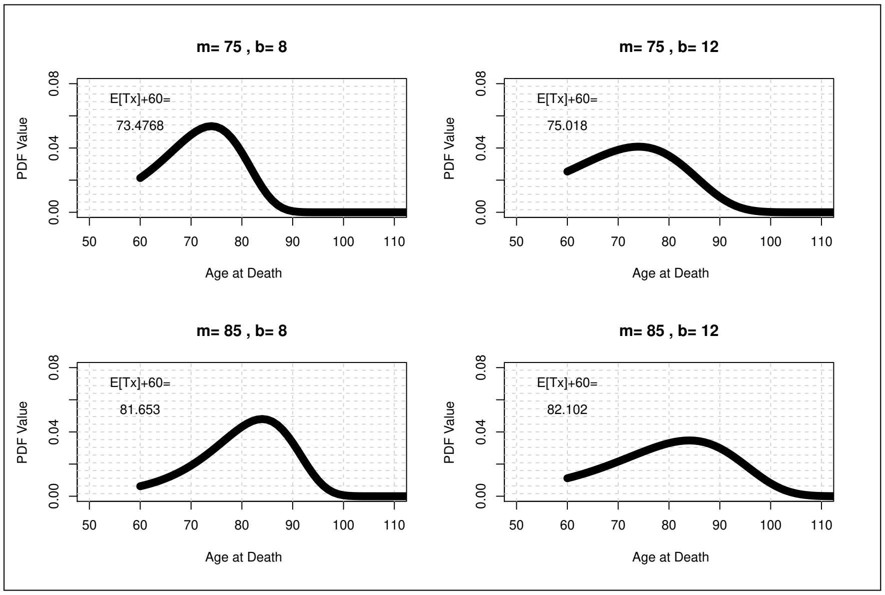
Create a series of four (4) plots of the probability density function (PDF) of Gompertz-Makeham, using the `par(mfrow=c(2,2))` command, for $x = 60$. Use the parameter values $\lambda = 3 \times 10^{-5}$, and $m = 75, m = 85$ and $b = 8, b = 12$. Compute $E[T_x]$ for all four combinations, using `GMLE(.)`, and discuss the impact of m and b on life expectancy.

Answer

I start by using `par(mfrow=c(2,2)` to create my 2×2 grid of plots. I then loop through both m values and both b values and calculate and plot the PDF for each (m, b) pair. I get the values of the probability density function point-by-point using `GMPDF`. For each (m, b) pair I also use `GMLE(.)` to compute life expectancy.

```
#set up 2x2 for plots
par(mfrow=c(2,2))
#loop through m values
for (m in c(75,85)){
  #loop through b values
  for (b in c(8,12)){
    #get PDF values for this (m,b)
    for (i in 1:60){
      x[i]<-GMPDF(60,i,0.00003,m,b)}
    #set plot size and labels
    plot(c(50,110),c(0,0.08),type="n",main=paste("m=",m,", b="
    ,b), xlab=" Age at Death",ylab="PDF Value")
    #add gridlines for clarity
    grid(ny=18,lty=20)
    #plot the PDF values
    lines(60:(59+length(x)),x,col="black",type="l",lwd=8)
    #compute and print life expectancy
    text(60,0.07,"E[Tx]+60=")
    text(60,0.054,round(GMLE(60,0.00003,m,b),4)+60)}}
```

The resulting figure of the four plots can be seen below.



From the output I see that raising m obviously lifts life expectancy, but the interesting thing is that raising b and keeping m constant, also lifts life expectancy.

13.2 Original Question

Derive an expression for the number of (calendar) years it takes your mortality rate to triple under a pure Gompertz law (i.e. assuming the Makeham constant $\lambda = 0$), assuming $b = 1/g = 10$.

Answer

From section 13.2 in the chapter I know:

$$\lambda_x = (1/b)e^{(x-m)/b}$$
$$\lambda_{x+z} = (1/b)e^{(x+z-m)/b}$$

Now I must solve for z in the following equation:

$$\lambda_{x+z}/\lambda_x = 3$$

When I do this I get:

$$\begin{aligned} e^{(x+z-m)/b}/e^{(x-m)/b} &= 3 \\ e^{z/b} &= 3 \\ z/b &= ln(3) \\ z &= bln(3) \end{aligned}$$

Assuming $b = 1/g = 10$, it takes $10 \times ln(3) \approx 11$ years for my hazard rate to triple under a pure Gompertz law.

13.3 Original Question

Using the *compensation* law of mortality, and the `CLAM(.)` function, please compute the range of possible value of m, assuming that $b \in [8, 12]$, that $x^* \in [95, 105]$ and that $\ln[\lambda^*] \in [-2, 0]$. In other words, what are highest and lowest possible values of m?

Answer

I solve for the range of m values using the `CLAM(.)` function while looping through the other parameter ranges given in the question. One thing to notice is that the question gives a range for $ln(\lambda^*)$ but the `CLAM(.)` function takes in λ^* itself. I accommodate this discrepancy by looping through values of λ^* ranging from $e^{(-2)}$ to e^0.

```
#create placeholder for m values
m<-c()
#loop through b
for (b in 8:12){
  #loop through x_star
  for(x_star in 95:105){
    #loop through e^ln(lambda*)
    for(lam_star in exp(-2:0)){
      #calculate m and append to list
      m<-append(m,CLAM(b,x_star,lam_star))}}}
```

I find the highest and lowest possible values of m.

```
> min(m)
[1] 65.18112
> max(m)
[1] 104.3645
```

The value of m ranges from 65.18 to 104.36.

13.4 Original Question

Please download one-year death rates from the HMD, for the period 2011 in Australia and Lithuania (male or female), and use the algorithm in section #13.8 to estimate the best fitting Gompertz-Makeham values of $(\lambda, \ln[h_0], g)$. Convert the numbers from the *demographic* reference frame (h, g) to the *probabilistic* reference frame using (m, b). Finally, compare your estimates to those in Table 13.1 or Table 13.2

Answer

I begin by downloading the 2011 period data for Australia and Lithuania as `AUS` and `LIT`. I create a function `Algo` that takes in a vector of death rates, runs the algorithm in section 13.8, and prints a list containing $(\lambda, ln[h_0], g)$. I can reuse this function on the female and male vectors from Australia and Lithuania. I add a line at the end of the algorithm that converts $\ln[h_0]$ and g to m and b.

```
Algo<-function(qx){
  x<-35:95
  lambda_max<-min(qx)*10^5-1
  gap<-c()
  for (i in 1:lambda_max){
    lambda<-i*10^(-5)
    y<-log(log(1/(1-qx))-lambda)
    fit<-lm(y~x)
    gap[i]<-sigma(fit)}
  lambda_makeham<-which.min(gap)*10^(-5)
  y<-log(log(1/(1-qx))-lambda_makeham)
  fit<-lm(y~x)
  g<-as.numeric(fit$coefficients[2])
  lnh<-as.numeric(fit$coefficients[1]-log((exp(g)-1)/g))
  print(c(lambda_makeham*10^5,lnh,g))
  #print m and b
  print(c(-(lnh+log(1/g))/g,1/g))}
```

I run `algo` on my death rate vectors and get values for $(\lambda, \ln[h_0], g)$ and (m, b).

```
> Algo(Aus$Female[36:96])
[1]  32.0000000 -12.3885604   0.1134517
[1] 90.01345  8.81432
> Algo(Aus$Male[36:96])
[1]  66.0000000 -11.8607350   0.1120088
[1] 86.346425  8.927872
> Algo(LIT$Female[36:96])
[1]  41.00000000 -10.64275246   0.09732001
[1] 85.41925 10.27538
> Algo(LIT$Male[36:96])
[1]  1.00000000 -8.14330310  0.07338789
[1] 75.37084 13.62623
```

I summarize my results in the table below.

Country & Sex	Makeham: λ	Gompertz: m	Gompertz: b
Australia Female	$32{\times}10^{-5}$	90.01	8.81
Australia Male	$66{\times}10^{-5}$	86.35	8.93
Lithuania Female	$41{\times}10^{-5}$	85.42	10.28
Lithuania Male	$1{\times}10^{-5}$	75.37	13.63

Comparing to the values in table 13.1 and 13.2 in the chapter which I summarize in the table below.

Country & Sex	Makeham: λ	Gompertz: m	Gompertz: b
Australia Female	$30{\times}10^{-5}$	90.40	9.00
Australia Male	$63{\times}10^{-5}$	86.78	9.16
Lithuania Female	$38{\times}10^{-5}$	85.70	10.45
Lithuania Male	$1{\times}10^{-5}$	75.75	14.01

I see that my m and b values are less than 0.5 away from the values in the chapter. My λ values are within 3×10^{-5}.

13.5 Original Question

Assuming that $x^* = 100$ and that $\ln[\lambda^*] = -1$, which recall are the assumed coordinates of the mortality plateau, use the `LRAG(.)` function to compute the longevity-risk-adjusted age of an $x = 55$ year-old Russian versus Australian. Please average the two corresponding (i.e. the Russian and the Australian) values of λ and g, to obtain the required (average) $\hat{\lambda}$ and $\hat{g}$, which you need for the formula. Who is "older" the 55-year-old Russian or the 55-year-old Australian? Explain.

Answer

I start by setting my parameters that are given in the question $x^* = 100$, $\lambda^* = e^{-1}$ and $x = 55$.

```
#set global parameters
x_star<-100
lam_star<-exp(-1)
x<-55
```

I then get the λ, m, and b for males and females in Australia and Russia from Table 13.1 and 13.2 in the textbook. I will calculate a separate $\hat{\lambda}$ and $\hat{g}$ for males and for females.

```
#Russian male lambda
lam.rm<-212*10^(-5)
#Australian male lambda
lam.am<-63*10^(-5)
#Russian male g = 1/b
g.rm<-1/13.99
#Australian male g = 1/b
g.am<-1/9.16
```

```
#lambda hat for males
lam_hat.m<-mean(c(lam.rm,lam.am))
#g hat for males
g_hat.m<-mean(c(g.rm,g.am))
```

```
#Russian female lambda
lam.rf<-153*10^(-5)
#Australian female lambda
lam.af<-30*10^(-5)
#Russian female g = 1/b
g.rf<-1/9.67
#Australian female g = 1/b
g.af<-1/9.00
#lambda hat for females
lam_hat.f<-mean(c(lam.rf,lam.af))
#g hat for females
g_hat.f<-mean(c(g.rf,g.af))
```

I now use the LRAG function to calculate the longevity-risk-adjusted age for the chronologically 55-year-old Russian and Australian males and females.

```
> #Russian male longevity risk adjusted age
> LRAG(x,x_star,g_hat.m,g.rm,lam_hat.m,lam.rm,lam_star)
[1] 64.93433
> #Australian male longevity risk adjusted age
> LRAG(x,x_star,g_hat.m,g.am,lam_hat.m,lam.am,lam_star)
[1] 42.04454
> #Russian female longevity risk adjusted age
> LRAG(x,x_star,g_hat.f,g.rf,lam_hat.f,lam.rf,lam_star)
[1] 58.12207
> #Australian female longevity risk adjusted age
> LRAG(x,x_star,g_hat.f,g.af,lam_hat.f,lam.af,lam_star)
[1] 50.72654
```

I see that from a longevity-risk-adjusted perspective, the Russian 55-year-old male is 23 years "older" than the Australian male, and the Russian female is 8 years "older" than the Australian female.

Chapter 14
Solutions

14.1 Original Question

Per the discussion around the Partial Differential Equation (14.9), create an **R** script that solves for the value of the RCLA using the numerical scheme which is described in great detail in the reference cited as [14]. This course assumes you have experience solving a PDE. If not, skip right ahead to the next problem.

Answer

To be solved in version 2.0.

14.2 Original Question

Using the **R** script for the `RCLA(.)` value, confirm the values listed in Table 14.1 and compute the relevant numbers at the age of $x = 60$ and $\sigma = 25$, and confirm they all sit in between the values listed in the table.

Answer

I use the `RCLA` function given in the chapter, with $x = 60$ and $\sigma = 25$, along with the same m, b, v, N parameters used in table 14.1. I run the function five times, once for each ξ value in the table.

```
x1<-round(RCLA(60,88,10,0.03,0.25,0.01,50000),2)
x2<-round(RCLA(60,88,10,0.04,0.25,0.01,50000),2)
x3<-round(RCLA(60,88,10,0.05,0.25,0.01,50000),2)
x4<-round(RCLA(60,88,10,0.06,0.25,0.01,50000),2)
x5<-round(RCLA(60,88,10,Inf,0.25,0.01,50000),2)
```

Here is my output. Since this uses simulated random numbers, your output will likely differ slightly.

```
> x1;x2;x3;x4;x5
[1] 4.00
[1] 5.59
[1] 7.06
[1] 8.38
[1] 21.14
```

I summarize my values of the ruin-contingent life annuities (RCLA) in the table below.

	$\xi = 3\%$	$\xi = 4\%$	$\xi = 5\%$	$\xi = 6\%$	$\xi = \infty$
Age x=60	4.00	5.59	7.06	8.38	21.14

Comparing these to the values in table 14.1 where $\sigma = 30\%$.

	$\xi = 3\%$	$\xi = 4\%$	$\xi = 5\%$	$\xi = 6\%$	$\xi = \infty$
Age x=55	7.12	8.95	10.46	11.85	24.46
Age x=65	3.39	4.57	5.64	6.62	17.88

I see that all my values sit between the values in Table 14.1.

14.3 Original Question

Modify the `RCLA(.)` code so that it assumes *weekly* and not monthly withdrawals, and then recompute the numbers in Table 14.1. Are the RCLA values higher or lower when the withdrawals are taken out weekly versus monthly? Is there a consistent pattern? Explain intuitively why this might be the case.

Answer

I made the following modifications to the `RCLA(.)` code so that it assumes *weekly* and not monthly withdrawals.

```
RCLA.weeks<-function(x,m,b,xi,sigma,r,N){
  #replace months with weeks
  wks<-52*(110-x)
  path<-matrix(nrow=N,ncol=wks)
  PLV<-c()
  for (i in 1:N){
    #sigma is now divided by root 52 instead of root 12
    k2<-sigma/sqrt(52)
    #r is divided by 52 instead of 12
    k1<-r/52-(0.5)*k2^2
    return<-exp(rnorm(wks,k1,k2))
    path[i,1]<-1
    for (j in 2:wks){
      #xi is divided by 52 instead of 12
      path[i,j]<-path[i,j-1]*return[j]-xi/52
      if (path[i,j]<=0) {break}}
      #the deferment time is converted from weeks to years
    PLV[i]=GDLA(x,x+(j-2)/52,r,m,b)}
  mean(PLV)}
```

I then use my `RCLA.weeks` function to recompute the values from table 14.1.

```
#set fixed parameters
m<-88
```

```
b<-10
r<-0.01
N<-50000
```

```
#loop through changing parameters
for (sigma in c(0.15,0.30)){
  for (x in c(55,65)){
    for (xi in c(0.03,0.04,0.05,0.06,Inf)){
      val<-round(RCLA.weeks(x,m,b,xi,sigma,r,N),2)
      print(c(sigma,x,xi,val))}}}
```

Here is my output. The first three values in each row are σ, x, and ξ, they are followed by the RCLA value for those parameters.

```
[1]  0.15 55.00  0.03  3.08
[1]  0.15 55.00  0.04  5.35
[1]  0.15 55.00  0.05  7.59
[1]  0.15 55.00  0.06  9.53
[1]  0.15 55.00   Inf 24.46
[1]  0.15 65.00  0.03  1.03
[1]  0.15 65.00  0.04  2.14
[1]  0.15 65.00  0.05  3.40
[1]  0.15 65.00  0.06  4.68
[1]  0.15 65.00   Inf 17.88
[1]  0.30 55.00  0.03  7.06
[1]  0.30 55.00  0.04  8.91
[1]  0.30 55.00  0.05 10.45
[1]  0.30 55.00  0.06 11.79
[1]  0.30 55.00   Inf 24.46
[1]  0.30 65.00  0.03  3.34
[1]  0.30 65.00  0.04  4.55
[1]  0.30 65.00  0.05  5.61
[1]  0.30 65.00  0.06  6.59
[1]  0.30 65.00   Inf 17.88
```

These values are summarized in the tables below.

Value of RCLA assuming weekly withdrawals for $\sigma = 15\%$

Purchase Age	$\xi = 3\%$	$\xi = 4\%$	$\xi = 5\%$	$\xi = 6\%$	$\xi = \infty$
Age x=55	3.08	5.35	7.59	9.53	24.46
Age x=65	1.03	2.14	3.40	4.68	17.88

Value of RCLA assuming weekly withdrawals for $\sigma = 30\%$

Purchase Age	$\xi = 3\%$	$\xi = 4\%$	$\xi = 5\%$	$\xi = 6\%$	$\xi = \infty$
Age $x = 55$	7.06	8.91	10.45	11.79	24.46
Age $x = 65$	3.34	4.55	5.61	6.59	17.88

Here is a table showing the differences between my values and the values in the chapter in table 14.1.

Weekly withdrawals RCLA - Ordinary RCLA for $\sigma = 15\%$					
Purchase Age	$\xi = 3\%$	$\xi = 4\%$	$\xi = 5\%$	$\xi = 6\%$	$\xi = \infty$
Age x=55	-0.01	0-.05	-0.03	-0.06	0
Age x=65	0	0.01	-0.05	-0.03	0

Weekly withdrawals RCLA - Ordinary RCLA for $\sigma = 30\%$					
Purchase Age	$\xi = 3\%$	$\xi = 4\%$	$\xi = 5\%$	$\xi = 6\%$	$\xi = \infty$
Age $x = 55$	-0.06	-0.04	-0.01	-0.06	0
Age $x = 65$	-0.05	-0.02	-0.03	-0.03	0

When I compare my values here (weekly withdrawals), with the values from table 14.1 (monthly withdrawals), the numbers are very close, but usually a little bit lower. The intuition is that the ordinary RCLA begins paying at the beginning of the year when ruin occurs, but if we assume weekly withdrawals from the reference portfolio, then the RCLA only begins paying at the beginning of the week of ruin. The later we expect the RCLA to begin paying, the lower its value. Note that when $\xi = \infty$, our reference portfolio and its withdrawals becomes irrelevant and the RCLA is just a regular life annuity, and can be computed using the `GILA` function.

14.4 Original Question

Modify the `RCLA(.)` code to value a modified version that begins paying $1 for life, as soon as the underlying reference index, which I denoted by $\tilde{F}_t$, loses 80% of its value. In other words, the payoff is not triggered by *ruin*, but by *very bad markets*. Please generate numbers similar to Table 14.1, and compare results. How much *more* expensive is such a product? Why?

Answer

I modify the `RCLA` function and create `RCLA.loses80`. This function checks to see if the portfolio value is below 20%, and begins paying the deferred annuity at the time that this is true.

```
RCLA.loses80<-function(x,m,b,xi,sigma,r,N){
  mnths<-12*(110-x)
  path<-matrix(nrow=N,ncol=mnths)
  PLV<-c()
  for (i in 1:N){
    k2<-sigma/sqrt(12)
    k1<-r/12-(0.5)*k2^2
    return<-exp(rnorm(mnths,k1,k2))
    path[i,1]<-1
    for (j in 2:mnths){
      path[i,j]<-path[i,j-1]*return[j]-xi/12
      #look for drop below 20% instead of ruin
      if (path[i,j]<=0.2) {break}}
    PLV[i]=GDLA(x,x+(j-2)/12,r,m,b)}
  mean(PLV)}
```

I use this function to generate values similar to the values in table 14.1.

```
#set fixed parameters
m<-88
b<-10
r<-0.01
N<-50000
```

```
#loop through changing parameters
for (sigma in c(0.15,0.30)){
  for (x in c(55,65)){
    for (xi in c(0.03,0.04,0.05,0.06,Inf)){
      val<-round(RCLA.loses80(x,m,b,xi,sigma,r,N),2)
      print(c(sigma,x,xi,val))}}}
```

Here is my output. The first three values in each row are again σ, x, and ξ.

```
[1]  0.15 55.00  0.03  5.68
[1]  0.15 55.00  0.04  8.01
[1]  0.15 55.00  0.05 10.13
[1]  0.15 55.00  0.06 11.95
[1]  0.15 55.00   Inf 24.46
[1]  0.15 65.00  0.03  2.55
[1]  0.15 65.00  0.04  3.92
[1]  0.15 65.00  0.05  5.31
[1]  0.15 65.00  0.06  6.59
[1]  0.15 65.00   Inf 17.88
[1]  0.30 55.00  0.03 10.80
[1]  0.30 55.00  0.04 12.07
[1]  0.30 55.00  0.05 13.26
[1]  0.30 55.00  0.06 14.27
[1]  0.30 55.00   Inf 24.46
[1]  0.30 65.00  0.03  6.17
[1]  0.30 65.00  0.04  7.08
[1]  0.30 65.00  0.05  7.96
[1]  0.30 65.00  0.06  8.68
[1]  0.30 65.00   Inf 17.88
```

My results are summarized in the table below.

Value of Modified RCLA for $\sigma = 15\%$					
Purchase Age	$\xi = 3\%$	$\xi = 4\%$	$\xi = 5\%$	$\xi = 6\%$	$\xi = \infty$
Age x=55	5.68	8.01	10.13	11.95	24.46
Age x=65	2.55	3.92	5.31	6.59	17.88

Value of Modified RCLA) for $\sigma = 30\%$					
Purchase Age	$\xi = 3\%$	$\xi = 4\%$	$\xi = 5\%$	$\xi = 6\%$	$\xi = \infty$
Age $x = 55$	10.80	12.07	13.26	14.27	24.46
Age $x = 65$	6.17	7.08	7.96	8.68	17.88

Here is a table showing my values divided by those in table 14.1.

Modified RCLA divided by Ordinary RCLA for $\sigma = 15\%$					
Purchase Age	$\xi = 3\%$	$\xi = 4\%$	$\xi = 5\%$	$\xi = 6\%$	$\xi = \infty$
Age x=55	1.84	1.48	1.33	1.25	1
Age x=65	2.48	1.84	1.54	1.40	1

Modified RCLA divided by Ordinary RCLA for $\sigma = 30\%$					
Purchase Age	$\xi = 3\%$	$\xi = 4\%$	$\xi = 5\%$	$\xi = 6\%$	$\xi = \infty$
Age $x = 55$	1.52	1.35	1.27	1.20	0
Age $x = 65$	1.82	1.55	1.41	1.31	0

I see that the relative price of the modified RCLA becomes cheaper for higher ξ and for higher volatility, but it becomes more expensive for higher age. This makes sense intuitively because higher ξ means that if the portfolio loses 80% of its value its very likely to get ruined very soon after. Higher volatility also raises the likelihood of an 80% decline to be swiftly followed by ruin. Older age however implies that the gap in time between an 80% decline and time of ruin is a larger percentage of my remaining lifetime, thus the value of this product becomes relatively more expensive than the ordinary RCLA.

14.5 Original Question

Finally, please modify (or create a new) **R** script that prices the following variant of an RCLA. Namely, assume that you have $\tilde{F}_0 = \$100$, and are withdrawing at the rate of $\xi \tilde{F}_0$. But, if at the end of the year the portfolio value $\tilde{F}_t$ exceeds the original F_0, the withdrawal amount is **ratcheted** up and increased to $\xi \tilde{F}_t$. So, for example, if $\tilde{F}_1 = \$200$, by some lucky chance, then from that point onward the withdrawal amount is doubled, but never reduced. What is the value of an RCLA that pays-off \$1 if-and-when this enhanced index is ruined?

Answer

I modify the `RCLA` function by creating a variable called `Fmax` to track the maximum value of the portfolio. I initialize this value to 1. At each step in the path, I check if the portfolio value exceeds `Fmax`, if it does, then I replace `Fmax` with the portfolio value. When I am subtracting my consumption ξ from the portfolio value, I multiply ξ by `Fmax`.

```
RCLA.ratchet<-function(x,m,b,xi,sigma,r,N){
  mnths<-12*(110-x)
  path<-matrix(nrow=N,ncol=mnths)
  PLV<-c()
  for (i in 1:N){
    k2<-sigma/sqrt(12)
    k1<-r/12-(0.5)*k2^2
    return<-exp(rnorm(mnths,k1,k2))
    path[i,1]<-1
    #initialize highest portfolio value
    Fmax<-1
    for (j in 2:mnths){
      #multiply xi by highest portfolio value
      path[i,j]<-path[i,j-1]*return[j]-(Fmax*xi/12)
      if (path[i,j]<=0) {break}
      #if portfolio value exceeded is max update the max
      if (path[i,j]>Fmax){Fmax<-path[i,j]}}
    PLV[i]=GDLA(x,x+(j-2)/12,r,m,b)}
  mean(PLV)}
```

```
> x1<-round(RCLA.ratchet(60,88,10,0.03,0.25,0.01,50000),2)
> x2<-round(RCLA.ratchet(60,88,10,0.04,0.25,0.01,50000),2)
> x3<-round(RCLA.ratchet(60,88,10,0.05,0.25,0.01,50000),2)
> x4<-round(RCLA.ratchet(60,88,10,0.06,0.25,0.01,50000),2)
> x5<-round(RCLA.ratchet(60,88,10,Inf,0.25,0.01,50000),2)
>
> x1;x2;x3;x4;x5
[1] 5.46
[1] 7.33
[1] 8.84
[1] 10.11
[1] 21.14
```

The percentage difference between our values and those in question 14.2 are $36.5\%, 31\%, 25\%, 21\%, 0\%$ for our respective choices of ξ. This means the Ratcheted RCLA is relatively more expensive for lower values of ξ. This makes sense intuitively, since the lower the value of ξ the more likely it becomes for the portfolio to exceed its original value resulting in different spending paths from the ordinary RCLA.