Assignment 7 Nathan Esau
Acma 320 301197568

Instructions

In your code, the age at issue (x), the annual effective interest rate (i), the term of the contract (n), the payment frequency (m), and the parameters related to the life table should all be inputs. Everything else should be calculated internally. For purposes of your printouts, please use an age at issue of $x = 40$, an interest rate of $i = 0.05$, a term of $n = 10$ and a payment frequency of $m = 4$.

Task

1. (DHW Example 6.1 modified) Use the Standard Select Survival Model with interest at 5% per year, to produce a table showing values of $\ddot{a}_{[x]}$, $\ddot{a}_{[x]+1}$ and $\ddot{a}_{x+2}$ for $x = 20, 21, \ldots, 80$. Assume that $q_{131} = 1$. Use the parameters $A = 0.00022$, $B = 2.5e-05$, $c = 1.1$ and $\omega = 131$.

Solution:

The code used in this assignment can be found at https://github.com/nathanesau/acma320/blob/master/R/assign7.R. The code used to produce the annuity table is shown below.

```
> d <- 2
> A <- 0.00022
> B <- 2.5e-05
> c <- 1.1
> omega <- 131
> radix <- 1e+05
> x <- 40
> i <- 0.05
> n <- 10
> m <- 4
> tpx <- function(t,x,s) {
+
+   f <- function(t,x,s) {
+     u = pmax(0,d-s-t)
+     j = t <= (d-s)
+
+     exp(0.9^(u) * (
+       A*(1-j*0.9^(t))*t^(1-j)/(log(0.9)^j - 2*(1 - j)) +
+         B*c^(x)*(c^(t) - 0.9^(j*t))/(log(0.9)*j - log(c))
+     ))
+
+   }
+
+   m = pmax(0,d-s)
```

```
+   f(m,x,s)* f(t-m,x+m,d)
+ }
> v <- function(i, n) {
+   (1+i)^(-n)
+ }
> tEx <- function(t, x, s, i) {
+   v(i, t) * tpx(t, x, s)
+ }
> createAnnuityTable <- function(x, radix, A, B, c, omega, d, i,
+                                 moment) {
+
+   i <- exp(moment * log(1+i)) - 1
+
+   Ax = vector("list", d + 1) # whole life insurance
+   ax = vector("list", d + 1) # annuity table
+   p = vector("list", d + 1)
+
+   # recursive insurance
+   recins <- function(p, init = FALSE, Ax = NA) {
+
+     # Calculate A[x] values using recursion
+     prev = v(i,1)
+     A = prev
+
+     # Special case for init
+     for(t in (omega-d-1):x) {
+
+       prev = ifelse(init, (1-p[t-x+1])*v(i,1) + p[t-x+1]*v(i,1)*prev,
+                     (1-p[t-x+1])*v(i,1) + p[t-x+1]*v(i,1)*Ax[t-x+1])
+       A = c(A, prev)
+     }
+
+     # A is backwards (since we use a backwards recursion).
+     # Need to reverse A
+     rev(A)
+   }
+
+   # calculate probabilities of survival over one year intervals
+   p = lapply(0:d, function(t) {
+     sapply((omega-d-1):x, function(s) {
+       tpx(1, s + t, t)
+     }
+     )
+   }
+   )
```

```
+
+   # calculate select insurance formulas recursively
+   for(t in d:0) {
+     if(t == d) {
+       Ax[[t+1]] = recins(rev(p[[t+1]]), TRUE)
+     } else {
+       Ax[[t+1]] = recins(rev(p[[t+1]]), FALSE, Ax[[t+2]])
+     }
+   }
+
+   ax <- lapply(Ax, function(y) (1 - y)/(i/(1+i)))
+
+   # combine lists into data frame (insurance table)
+   it = data.frame(x:(omega-d), Ax, ax, x:(omega-d) + d)
+
+   # rename data frame
+   if(d>0) {
+     names(it) = c("x", paste0(ifelse(moment == 1, "", moment), "A[x]"),
+                   sapply(1:(d-1), function(d) {
+                     paste0(ifelse(moment == 1, "", moment), "A[x]+", d)
+                   }),
+                   paste0(ifelse(moment == 1, "", moment), "Ax+", d),
+                   paste0(ifelse(moment == 1, "", moment), "a[x]"),
+                   sapply(1:(d-1), function(d) {
+                     paste0(ifelse(moment == 1, "", moment), "a[x]+", d)
+                   }),
+                   paste0(ifelse(moment == 1, "", moment), "ax+", d),
+                   paste0("x+",d))
+   } else {
+     names(it) = c("x", "Ax", "ax", "x")
+     if(moment > 1) {
+       names(it)[2:3] = paste0(moment, ":", names(it)[2:3])
+     }
+   }
+
+   # remove last row
+   head(it,-1)
+ }
> annuityTable <- createAnnuityTable(x, radix, A, B, c,
+                                    omega, d, i, 1)
```

The values are shown in Table 1.

x	$A_{[x]}$	$A_{[x]+1}$	A_{x+2}	$\ddot{a}_{[x]}$	$\ddot{a}_{[x]+1}$	$\ddot{a}_{x+2}$	$x+2$
40	0.16364	0.17083	0.17818	17.56350	17.41260	17.25823	42
41	0.17065	0.17812	0.18574	17.41627	17.25958	17.09944	43

Nathan Esau
301197568

x	$A_{[x]}$	$A_{[x]+1}$	A_{x+2}	$\ddot{a}_{[x]}$	$\ddot{a}_{[x]+1}$	$\ddot{a}_{x+2}$	$x+2$
42	0.17793	0.18567	0.19358	17.26352	17.10089	16.93487	44
43	0.18547	0.19350	0.20169	17.10513	16.93643	16.76441	45
44	0.19329	0.20161	0.21010	16.94100	16.76609	16.58795	46
45	0.20138	0.21001	0.21879	16.77101	16.58977	16.40542	47
46	0.20976	0.21870	0.22778	16.59507	16.40737	16.21672	48
47	0.21842	0.22767	0.23706	16.41309	16.21883	16.02179	49
48	0.22738	0.23695	0.24664	16.22499	16.02407	15.82058	50
49	0.23663	0.24652	0.25652	16.03070	15.82303	15.61303	51
50	0.24618	0.25640	0.26671	15.83018	15.61567	15.39913	52
51	0.25603	0.26657	0.27720	15.62339	15.40198	15.17886	53
52	0.26618	0.27705	0.28799	15.41029	15.18193	14.95223	54
53	0.27662	0.28783	0.29908	15.19088	14.95554	14.71928	55
54	0.28737	0.29891	0.31047	14.96519	14.72284	14.48004	56
55	0.29842	0.31029	0.32216	14.73322	14.48387	14.23459	57
56	0.30976	0.32197	0.33414	14.49505	14.23872	13.98303	58
57	0.32139	0.33393	0.34641	14.25075	13.98747	13.72548	59
58	0.33331	0.34618	0.35895	14.00040	13.73025	13.46208	60
59	0.34552	0.35870	0.37176	13.74415	13.46721	13.19301	61
60	0.35799	0.37150	0.38483	13.48213	13.19851	12.91847	62
61	0.37074	0.38455	0.39816	13.21453	12.92438	12.63870	63
62	0.38374	0.39786	0.41172	12.94156	12.64503	12.35395	64
63	0.39698	0.41139	0.42550	12.66343	12.36073	12.06453	65
64	0.41046	0.42515	0.43949	12.38043	12.07178	11.77074	66
65	0.42415	0.43912	0.45367	12.09285	11.77850	11.47295	67
66	0.43805	0.45327	0.46802	11.80101	11.48123	11.17154	68
67	0.45213	0.46760	0.48253	11.50527	11.18038	10.86693	69
68	0.46638	0.48208	0.49716	11.20600	10.87635	10.55955	70
69	0.48078	0.49669	0.51191	10.90364	10.56958	10.24988	71
70	0.49530	0.51140	0.52674	10.59861	10.26054	9.93841	72
71	0.50993	0.52620	0.54163	10.29138	9.94974	9.62567	73
72	0.52464	0.54106	0.55656	9.98246	9.63769	9.31219	74
73	0.53941	0.55596	0.57150	9.67236	9.32492	8.99854	75
74	0.55421	0.57086	0.58641	9.36161	9.01201	8.68530	76
75	0.56901	0.58574	0.60128	9.05077	8.69952	8.37304	77
76	0.58379	0.60057	0.61608	8.74042	8.38805	8.06238	78
77	0.59852	0.61532	0.63077	8.43113	8.07818	7.75391	79
78	0.61317	0.62997	0.64532	8.12349	7.77053	7.44823	80
79	0.62771	0.64449	0.65972	7.81809	7.46568	7.14595	81
80	0.64212	0.65885	0.67392	7.51554	7.16424	6.84766	82
81	0.65636	0.67301	0.68791	7.21640	6.86680	6.55393	83
82	0.67042	0.68696	0.70165	6.92127	6.57392	6.26533	84
83	0.68425	0.70066	0.71512	6.63070	6.28617	5.98238	85
84	0.69785	0.71409	0.72830	6.34525	6.00408	5.70562	86
85	0.71117	0.72723	0.74117	6.06542	5.72815	5.43550	87

x	$A_{[x]}$	$A_{[x]+1}$	A_{x+2}	$\ddot{a}_{[x]}$	$\ddot{a}_{[x]+1}$	$\ddot{a}_{x+2}$	$x+2$
86	0.72420	0.74005	0.75369	5.79171	5.45888	5.17250	88
87	0.73692	0.75254	0.76586	5.52460	5.19668	4.91701	89
88	0.74931	0.76467	0.77765	5.26449	4.94198	4.66940	90
89	0.76134	0.77642	0.78905	5.01179	4.69514	4.43000	91
90	0.77301	0.78779	0.80004	4.76684	4.45647	4.19909	92
91	0.78429	0.79875	0.81062	4.52994	4.22626	3.97690	93
92	0.79517	0.80930	0.82078	4.30135	4.00472	3.76363	94
93	0.80565	0.81943	0.83050	4.08129	3.79204	3.55940	95
94	0.81572	0.82913	0.83979	3.86991	3.58836	3.36432	96
95	0.82536	0.83839	0.84865	3.66734	3.39377	3.17844	97
96	0.83459	0.84722	0.85706	3.47364	3.20829	3.00174	98
97	0.84339	0.85562	0.86504	3.28885	3.03194	2.83421	99
98	0.85176	0.86359	0.87258	3.11294	2.86466	2.67575	100
99	0.85972	0.87112	0.87970	2.94585	2.70638	2.52625	101
100	0.86726	0.87824	0.88640	2.78750	2.55697	2.38558	102

Table 1: Annuity values for Question 1

2. Consider the following contracts sold to $[x]$
   (i) n-year temporary life annuity contract payable $m$thly
   (ii) n-year deferred whole life annuity contract payable $m$thly

   For each contract, use your results from question 1 to calculate:

   (a) The expected present value of the cashflow(s) (i.e. insurance benefit or annuity payments)
   (b) The standard deviation of the present value of the cashflows
   (c) The 70th and 95th percentiles of the distribution of the present value of cashflow(s) for a portfolio of c contracts, $c = 100$ or 10,000 (use the Normal approximation)

Assume UDD for fractional ages (i.e. do not use the actual underlying Makeham distribution).

Solution:

The code used to compute $A^{(m)}_{[x]:\overline{n}|}$, $A^{1\ (m)}_{[x]:\overline{n}|}$ (with $A^{1\ (m)}_{[x]:\overline{\omega-x-1}|} = A^{(m)}_{[x]}$) and $\ddot{a}^{(m)}_{[x]:\overline{n}|}$ (with $\ddot{a}^{(m)}_{[x]:\overline{\omega-x-1}|} = \ddot{a}^{(m)}_{[x]}$) is shown below.

```
> udeferredtqx <- function(u, t, x, s) {
+   tpx(u, x, s) - tpx(u + t, x, s)
+ }
> nEx <- function(n, x, s, i, moment) {
+   i <- exp(moment * log(1+i)) - 1
```

```
+
+   tpx(n, x, s) * v(i, n)
+ }
> Ax <- function(x, s, i, moment, n, m, type = 'term') {
+   ioriginal <- i
+   i <- exp(moment * log(1+i)) - 1
+   iupperm <- m*((1+i)^(1/m) - 1) # UDD
+
+   k <- seq(0, pmin(n, omega-x) - 1, 1)
+   pk <- udeferredtqx(k, 1, x, s)
+   vk <- v(i, k+1)
+   EPVterm <- sum(pk * vk) * i/iupperm
+
+   if(type == 'endowment') {
+     return(EPVterm + nEx(n, x, s, ioriginal, moment))
+   } else { # term of or whole life (for whole life use large n)
+     return(EPVterm)
+   }
+ }
> dupperm <- function(i, m) {
+   dannual <- i/(1+i)
+   m*(1 - (1-dannual)^(1/m))
+ }
> annx <- function(x, s, i, moment, n, m) {
+   (1 - Ax(x, s, i, moment, n, m, type='endowment'))/dupperm(i,m)
+ }
```

The EPV and variance for contract 1 are

$$
\begin{aligned}
EPV(\text{Contract 1}) &= \ddot{a}^{(m)}_{[x]:\overline{n}|} \\
Var(\text{Contract 2}) &= \frac{{}^2A^{(m)}_{[x]:\overline{n}|} - \left(A^{(m)}_{[x]:\overline{n}|}\right)^2}{(d^{(m)})^2}
\end{aligned}
$$

The EPV and variance for contract 2 are

$$
\begin{aligned}
EPV(\text{Contract 2}) &= {}_nE_{[x]}\left(\ddot{a}^{m}_{[x]+n}\right) \\
Var(\text{Contract 2}) &= {}^2{}_nE_{[x]}\left(\ddot{a}^{(m)}_{[x]+n}\right)^2 - \left({}_nE_{[x]}\ddot{a}^{(m)}_{[x]+n}\right)^2 + \\
&\quad \frac{{}^2{}_nE_{[x]}\left({}^2A^{(m)}_{[x]+n} - \left(A^{(m)}_{[x]+n}\right)^2\right)}{(d^{(m)})^2}
\end{aligned}
$$

These quantities are computed below.

```
> EPVInsurance1 <- annx(x, 0, i, 1, n, m)
> VarInsurance1 <- (Ax(x, 0, i, 2, n, m, type = 'endowment') -
+          Ax(x, 0, i, 1, n, m, type = 'endowment')^2) * dupperm(i, m)^(-2)
> EPVInsurance2 <- nEx(n, x, 0, i, 1) * (annx(x + n, n, i, 1, omega - x - 1, m))
> VarInsurance2 <- nEx(n, x, 0, i, 2) * annx(x + n, n, i, 1, omega - x - 1, m)^2 -
+          (nEx(n, x, 0, i, 1) * annx(x + n, n, i, 1, omega - x - 1, m))^2 +
+          nEx(n, x, 0, i, 2) * (Ax(x + n, n, i, 2, omega - x - 1, m) -
+          Ax(x + n, n, i, 1, omega - x - 1, m)^2) * dupperm(i, m)^(-2)
```

(a) The expected present of the cashflow is 7.90023 for Contract 1 and 9.28381 for Contract 2.

(b) The standard deviation of the present value of the cashflow is 0.5257 for Contract 1 and 2.46079 for Contract 2.

(c) The percentiles are computed below.

```
> Percentile <- function(EPV, Var, p, c) {
+   qnorm(p) * sqrt(Var) * sqrt(c) + c * EPV
+ }
> p70Insurance1_100 <- Percentile(EPVInsurance1, VarInsurance1, 0.70, 100)
> p95Insurance1_100 <- Percentile(EPVInsurance1, VarInsurance1, 0.95, 100)
> p70Insurance2_100 <- Percentile(EPVInsurance2, VarInsurance2, 0.70, 100)
> p95Insurance2_100 <- Percentile(EPVInsurance2, VarInsurance2, 0.95, 100)
> p70Insurance1_10000 <- Percentile(EPVInsurance1, VarInsurance1, 0.70, 10000)
> p95Insurance1_10000 <- Percentile(EPVInsurance1, VarInsurance1, 0.95, 10000)
> p70Insurance2_10000 <- Percentile(EPVInsurance2, VarInsurance2, 0.70, 10000)
> p95Insurance2_10000 <- Percentile(EPVInsurance2, VarInsurance2, 0.95, 10000)
```

For $c = 100$, the 70th percentile is 792.77982 for Contract 1 and 941.28576 for Contract 2. The 95th percentile is 798.67003 for Contract 1 and 968.85772 for Contract 2.

For $c = 10000$, the 70th percentile is 79029.87253 for Contract 1 and 92967.18228 for Contract 2. The 95th percentile is 79088.77468 for Contract 1 and 93242.90187 for Contract 2.