Nathan Esau
301197568

1. Mortality follows a select survival model with a 2-year select period. The ultimate part of the model follows Makeham's Law with $A = 0.00022$, $B = 2.5e - 05$ and $c = 1.1$. The select part of the model is such that $\mu_{[x]+s} = 0.9^{2-s}\mu_{x+s}$ for $0 \leq s \leq 2$. Starting with $l_{20} = 100,000$, calculate values of
   - l_x for $x = 20, 21, \ldots, 119$
   - $l_{[x]+1}$ for $x = 20, 21, \ldots, 119$
   - $l_{[x]}$ for $x = 20, 21, \ldots, 119$

   Be sure to code the parameters A, B and c as variable inputs.

Solution:

The code used in this assignment can be found at https://github.com/nathanesau/acma320/blob/master/R/assign4.R. The code used to produce the life table is shown below.

```
> d <- 2
> A <- 0.00022
> B <- 2.5e-05
> c <- 1.1
> omega <- 131
> radix <- 1e+05
> x <- 20
> i <- 0.05
> 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)
+ }
> createLifeTable <- function(x, radix, A, B, c, omega, d) {
+
+   if(d > 0) {
+
+     # creates the select life table
+     lt = data.frame(
```

```
+       c(rep(NA, d), x:(omega-d-1)),
+       lapply(0:(d-1),
+             function(y) {
+               c(rep(NA, d),
+                 tail(tpx(0:(omega-x-1), x, s = d)*radix, -d) /
+                 sapply(x:(omega-d-1),
+                        function(x) {
+                           tpx(d - y, x + y, y)
+                        }
+                        ))
+             }
+       ),
+       tpx(0:(omega-x-1), x, s = d)*radix, x:(omega-1)
+     )
+
+     # renames the select life table
+     names(lt) =  c("x", sapply(0:(d-1), function(x) {
+                                          paste0("l[x]+",x)
+                                        }),
+                    paste0("lx+",d), paste0("x+",d))
+   } else {
+     lt = data.frame(x:(omega-d-1), tpx(0:(omega-x-1), x - d, s = d)*radix,
+                    x:(omega-1))
+     names(lt) = c("x","lx","x")
+   }
+
+   lt
+ }
> lifeTable <- createLifeTable(x, radix, A, B, c, omega, d)
```

The life table is shown in Table 1.

x	$l_{[x]}$	$l_{[x]+1}$	l_{x+2}	$x+2$
			100,000.00	20
			99,960.36	21
20	99,992.14	99,958.27	99,918.98	22
21	99,952.15	99,916.79	99,875.67	23
22	99,910.38	99,873.38	99,830.25	24
23	99,866.65	99,827.84	99,782.50	25
24	99,820.77	99,779.97	99,732.21	26
25	99,772.51	99,729.53	99,679.10	27
26	99,721.65	99,676.27	99,622.91	28
27	99,667.93	99,619.91	99,563.33	29
28	99,611.06	99,560.14	99,500.02	30
29	99,550.73	99,496.62	99,432.61	31
30	99,486.60	99,428.99	99,360.70	32

x	$l_{[x]}$	$l_{[x]+1}$	l_{x+2}	$x+2$
31	99,418.29	99,356.83	99,283.84	33
32	99,345.39	99,279.70	99,201.55	34
33	99,267.44	99,197.11	99,113.29	35
34	99,183.95	99,108.52	99,018.47	36
35	99,094.38	99,013.34	98,916.45	37
36	98,998.11	98,910.92	98,806.52	38
37	98,894.51	98,800.56	98,687.91	39
38	98,782.84	98,681.47	98,559.78	40
39	98,662.33	98,552.82	98,421.19	41
40	98,532.10	98,413.65	98,271.12	42
41	98,391.21	98,262.95	98,108.48	43
42	98,238.62	98,099.61	97,932.03	44
43	98,073.21	97,922.41	97,740.45	45
44	97,893.73	97,730.00	97,532.30	46
45	97,698.83	97,520.94	97,305.98	47
46	97,487.03	97,293.62	97,059.78	48
47	97,256.73	97,046.32	96,791.80	49
48	97,006.15	96,777.15	96,500.00	50
49	96,733.40	96,484.04	96,182.16	51
50	96,436.37	96,164.76	95,835.84	52
51	96,112.81	95,816.89	95,458.44	53
52	95,760.25	95,437.78	95,047.10	54
53	95,376.03	95,024.58	94,598.76	55
54	94,957.26	94,574.21	94,110.09	56
55	94,500.81	94,083.33	93,577.53	57
56	94,003.32	93,548.37	92,997.25	58
57	93,461.16	92,965.47	92,365.12	59
58	92,870.44	92,330.50	91,676.75	60
59	92,226.99	91,639.06	90,927.48	61
60	91,526.35	90,886.46	90,112.33	62
61	90,763.79	90,067.72	89,226.06	63
62	89,934.29	89,177.58	88,263.18	64
63	89,032.56	88,210.52	87,217.92	65
64	88,053.02	87,160.78	86,084.33	66
65	86,989.89	86,022.38	84,856.25	67
66	85,837.16	84,789.18	83,527.42	68
67	84,588.64	83,454.90	82,091.54	69
68	83,238.07	82,013.24	80,542.34	70
69	81,779.14	80,457.92	78,873.70	71
70	80,205.59	78,782.86	77,079.79	72
71	78,511.37	76,982.23	75,155.24	73
72	76,690.71	75,050.69	73,095.29	74
73	74,738.36	72,983.52	70,896.06	75
74	72,649.73	70,776.90	68,554.76	76

x	$l_{[x]}$	$l_{[x]+1}$	l_{x+2}	$x+2$
75	70,421.14	68,428.10	66,069.99	77
76	68,050.07	65,935.81	63,442.03	78
77	65,535.43	63,300.39	60,673.18	79
78	62,877.91	60,524.26	57,768.08	80
79	60,080.25	57,612.19	54,734.08	81
80	57,147.65	54,571.69	51,581.54	82

Table 1: Life table for Question 1

2. (a) Use the life table you created in Question 1 together with the backward recursion formulas given in class (modified to suit a select-and-ultimate table) to construct a table of values of $A_{[x]}$ for integer ages ($x \geq 20$). Make the effective rate of interest, i an input parameter for your calculations.

   (b) Plot $A_{[x]}$ as a function of x when $i = 0.04$

   (c) Plot, on the same chart as in part (a), $A_{[x]}$ as a function of x when $i = 0.05$ and $i = 0.06$. How does $A_{[x]}$ change when the interest rate is increased?

Solution:

(a) The code used to produce the insurance table is shown below.

```
> v <- function(i, n) {
+   (1+i)^(-n)
+ }
> tEx <- function(t, x, s, i) {
+   v(i, t) * tpx(t, x, s)
+ }
> createInsuranceTable <- function(x, radix, A, B, c, omega, d, i, n,
+                                   moment) {
+
+   i <- exp(moment * log(1+i)) - 1
+
+   Ax = vector("list", d + 1) # whole life insurance
+   Ex = vector("list", d + 1) # pure endowment insurance
+   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]])
+     }
+
+     Ex[[t+1]] = sapply(x:(omega-d), function(s) {
+       tEx(n,s,t,i)
+     }
+     )
+   }
+
+   # combine lists into data frame (insurance table)
+   it = data.frame(x:(omega-d), Ax, Ex, 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, "", paste0(moment,":")),
+              paste0(n,"E[x]")),
+       sapply(1:(d-1), function(d) {
+         paste0(ifelse(moment == 1, "", paste0(moment, ":")),
+                paste0(n,"E[x]+"), d)
+       }),
+       paste0(ifelse(moment == 1, "", paste0(moment, ":")),
+              paste0(n, "Ex+"), d), paste0("x+",d))
+   } else {
+     names(it) = c("x", "Ax", paste0(n,"Ex"), "x")
+     if(moment > 1) {
+       names(it)[2:3] = paste0(moment, ":", names(it)[2:3])
+     }
+   }
+
+   # remove last row
+   head(it,-1)
+ }
> insuranceTable <- createInsuranceTable(x, radix, A, B, c,
+                                        omega, d, i, 5, 1)
```

The table of insurance factors is shown in Table 2.

x	$A_{[x]}$	$A_{[x]+1}$	A_{x+2}	${}_5E_{[x]}$	${}_5E_{[x]+1}$	${}_5E_{x+2}$	$x+2$
20	0.06891	0.07204	0.07528	0.78188	0.78184	0.78182	22
21	0.07199	0.07526	0.07865	0.78180	0.78175	0.78174	23
22	0.07521	0.07863	0.08216	0.78171	0.78166	0.78165	24
23	0.07857	0.08214	0.08583	0.78161	0.78156	0.78154	25
24	0.08208	0.08581	0.08966	0.78151	0.78145	0.78143	26
25	0.08575	0.08964	0.09366	0.78139	0.78133	0.78131	27
26	0.08957	0.09364	0.09784	0.78126	0.78119	0.78117	28
27	0.09357	0.09781	0.10219	0.78111	0.78105	0.78102	29
28	0.09774	0.10217	0.10674	0.78095	0.78088	0.78086	30
29	0.10209	0.10671	0.11147	0.78078	0.78070	0.78068	31
30	0.10662	0.11144	0.11641	0.78059	0.78051	0.78048	32
31	0.11135	0.11637	0.12155	0.78038	0.78029	0.78026	33
32	0.11628	0.12151	0.12690	0.78014	0.78005	0.78002	34
33	0.12141	0.12686	0.13247	0.77989	0.77979	0.77976	35
34	0.12676	0.13243	0.13827	0.77961	0.77950	0.77947	36
35	0.13232	0.13823	0.14430	0.77930	0.77919	0.77915	37
36	0.13811	0.14426	0.15058	0.77896	0.77884	0.77880	38
37	0.14413	0.15053	0.15709	0.77859	0.77846	0.77841	39
38	0.15039	0.15704	0.16386	0.77818	0.77804	0.77799	40
39	0.15689	0.16381	0.17089	0.77773	0.77758	0.77752	41

Nathan Esau
301197568

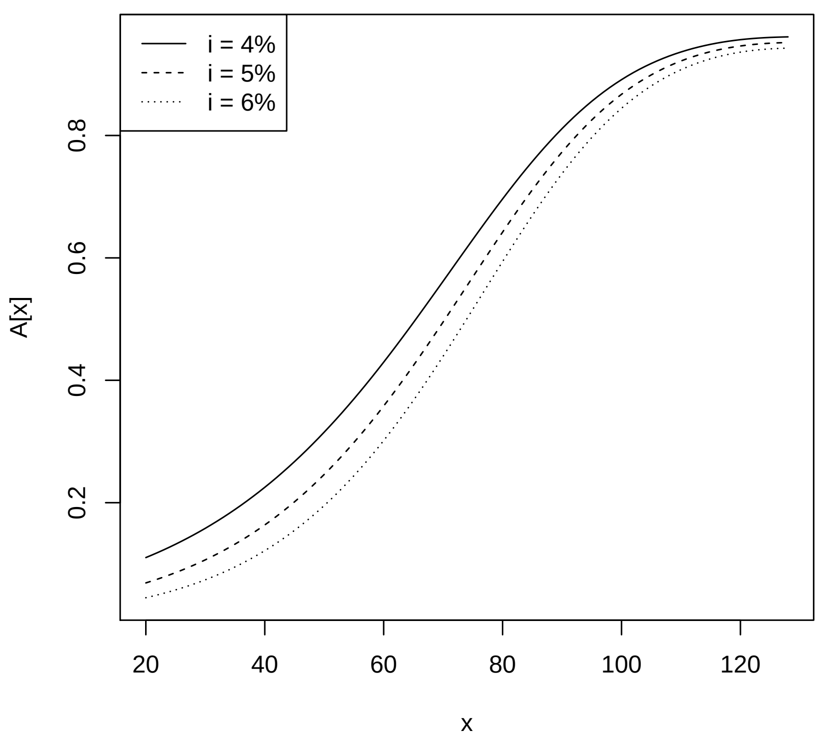
x	$A_{[x]}$	$A_{[x]+1}$	A_{x+2}	${}_5E_{[x]}$	${}_5E_{[x]+1}$	${}_5E_{x+2}$	$x+2$
40	0.16364	0.17083	0.17818	0.77723	0.77707	0.77701	42
41	0.17065	0.17812	0.18574	0.77669	0.77651	0.77645	43
42	0.17793	0.18567	0.19358	0.77609	0.77590	0.77583	44
43	0.18547	0.19350	0.20169	0.77543	0.77522	0.77515	45
44	0.19329	0.20161	0.21010	0.77471	0.77448	0.77440	46
45	0.20138	0.21001	0.21879	0.77391	0.77366	0.77358	47
46	0.20976	0.21870	0.22778	0.77304	0.77277	0.77268	48
47	0.21842	0.22767	0.23706	0.77208	0.77179	0.77169	49
48	0.22738	0.23695	0.24664	0.77103	0.77071	0.77060	50
49	0.23663	0.24652	0.25652	0.76987	0.76952	0.76940	51
50	0.24618	0.25640	0.26671	0.76860	0.76822	0.76809	52
51	0.25603	0.26657	0.27720	0.76720	0.76679	0.76665	53
52	0.26618	0.27705	0.28799	0.76567	0.76521	0.76506	54
53	0.27662	0.28783	0.29908	0.76398	0.76349	0.76332	55
54	0.28737	0.29891	0.31047	0.76214	0.76160	0.76142	56
55	0.29842	0.31029	0.32216	0.76011	0.75952	0.75932	57
56	0.30976	0.32197	0.33414	0.75789	0.75724	0.75703	58
57	0.32139	0.33393	0.34641	0.75545	0.75475	0.75451	59
58	0.33331	0.34618	0.35895	0.75278	0.75201	0.75175	60
59	0.34552	0.35870	0.37176	0.74985	0.74901	0.74873	61
60	0.35799	0.37150	0.38483	0.74664	0.74572	0.74542	62
61	0.37074	0.38455	0.39816	0.74313	0.74213	0.74179	63
62	0.38374	0.39786	0.41172	0.73929	0.73819	0.73782	64
63	0.39698	0.41139	0.42550	0.73508	0.73388	0.73348	65
64	0.41046	0.42515	0.43949	0.73048	0.72917	0.72874	66
65	0.42415	0.43912	0.45367	0.72545	0.72403	0.72356	67
66	0.43805	0.45327	0.46802	0.71996	0.71841	0.71790	68
67	0.45213	0.46760	0.48253	0.71397	0.71228	0.71172	69
68	0.46638	0.48208	0.49716	0.70744	0.70560	0.70499	70
69	0.48078	0.49669	0.51191	0.70033	0.69833	0.69766	71
70	0.49530	0.51140	0.52674	0.69258	0.69041	0.68969	72
71	0.50993	0.52620	0.54163	0.68416	0.68180	0.68102	73
72	0.52464	0.54106	0.55656	0.67502	0.67246	0.67161	74
73	0.53941	0.55596	0.57150	0.66510	0.66233	0.66141	75
74	0.55421	0.57086	0.58641	0.65436	0.65137	0.65037	76
75	0.56901	0.58574	0.60128	0.64274	0.63951	0.63844	77
76	0.58379	0.60057	0.61608	0.63021	0.62672	0.62557	78
77	0.59852	0.61532	0.63077	0.61670	0.61295	0.61171	79
78	0.61317	0.62997	0.64532	0.60217	0.59815	0.59682	80
79	0.62771	0.64449	0.65972	0.58659	0.58228	0.58085	81
80	0.64212	0.65885	0.67392	0.56991	0.56531	0.56379	82

Table 2: Table of A_x and ${}_5E_x$ values for Question 2(a)

(b) The plot of $A_{[x]}$ at $i = 0.04$, $i = 0.05$ and $i = 0.06$ is shown in Figure 1.

```
> Ax4 <- createInsuranceTable(x, radix, A, B, c, omega, d, 0.04,
+                             5, 1)$"A[x]"
> Ax5 <- createInsuranceTable(x, radix, A, B, c, omega, d, 0.05,
+                             5, 1)$"A[x]"
> Ax6 <- createInsuranceTable(x, radix, A, B, c, omega, d, 0.06,
+                             5, 1)$"A[x]"
> plot(x = x:(omega-d-1), Ax4, lty = 1, type = 'l',
+     ylab = "A[x]", xlab = "x", ylim = c(min(Ax6), max(Ax4)))
> lines(x = x:(omega-d-1), Ax5, lty = 2, type = 'l')
> lines(x = x:(omega-d-1), Ax6, lty = 3, type = 'l')
> legend('topleft', c("i = 4%", "i = 5%", "i = 6%"),
+        lty = c(1, 2, 3))
```

Figure 1: Plot of $A_{[x]}$ at $i = 0.04$, $i = 0.05$ and $i = 0.06$ for Question 2(b)(c)



(c) $A_{[x]}$ decreases as the interest rate increases.

3. In this question we explore the distribution of the present value random variable, Z, for a whole life insurance on a person aged 50 at selection with sum insured \$1 payable at the end of the year of death. Use the same survival model you constructed in Question 1.

   (a) Theoretical values

(i) Find the pmf of $K(50)$, i.e. $Pr[K(50) = k]$ for $k = 0, 1, 2, \ldots$. Assume the limiting age is 131.

(ii) Find the pmf of Z, i.e. $Pr[Z = z]$ for $z = 1, v, v^2, \ldots$

(iii) Compute the theoretical mean and standard deviation of Z

(iv) Compute the exact (not approximate) probability that the actual value of Z exceeds the theoretical mean $A_{[50]}$

(b) Simulation study with a sample size of $n = 50$

(i) Draw n observations of $K(50)$ from the distribution calculation in part (a). You can do this in R by calling the `rmultinom` function. In Excel, you need to use the inverse transform method: generate n uniformly distributed random numbers $(u_1, \ldots, u_n)$ using the function `RAND()` and then transform each to a value of $K(50)$ using the CDF of $K(50)$

(ii) For each observation, calculate the corresponding value of Z using $i = 0.05$

(iii) Plot a histogram of Z based on your sample. Also plot, if possible on the same chart, the theoretical distribution of Z you computed in part (a)(ii)

(iv) Determine the empirical mean and standard deviation of Z from your sample, and compare these against the theoretical mean and standard deviation computed in part (a)(iii)

(v) Use the simulated values of Z in your sample to estimate the probability that the actual value of Z exceeds the theoretical mean A_{50}. Compare this to the exact (theoretical) probability computed in part (a)(iv)

(c) Repeat part (b) with a sample size of $n = 500$

Solution:

(a) The code used to find the pmf of K and pmf of Z is below. The probabilities are in Table 3.

```
> udeferredtqx <- function(u, t, x, s) {
+   tpx(u, x, s) - tpx(u + t, x, s)
+ }
> createProbTable <- function(x, i) {
+
+   k <- 0:(omega-x-1)
+   pk <- udeferredtqx(k, 1, x, 0)
+   Fk <- cumsum(pk)
+   vk <- v(i,k)
+   pz <- c(0, head(pk, -1))
+   Fz <- cumsum(pz)
+
+   data.frame(k = k, pk = pk, Fk = Fk,
+              vk = vk, pz = pz, Fz = Fz)
+ }
> probTable <- createProbTable(50, i)
```

$K(x)$	$Pr[K(50) = k]$	$Pr[K(50) \leq k]$	v^k	$Pr(Z = v^k)$	$Pr[Z \geq v^k]$
0	0.00282	0.00282	1.00000	0.00000	0.00000
1	0.00341	0.00623	0.95238	0.00282	0.00282
2	0.00391	0.01014	0.90703	0.00341	0.00623
3	0.00427	0.01441	0.86384	0.00391	0.01014
4	0.00465	0.01906	0.82270	0.00427	0.01441
5	0.00507	0.02412	0.78353	0.00465	0.01906
6	0.00552	0.02964	0.74622	0.00507	0.02412
7	0.00602	0.03566	0.71068	0.00552	0.02964
8	0.00655	0.04222	0.67684	0.00602	0.03566
9	0.00714	0.04935	0.64461	0.00655	0.04222
10	0.00777	0.05712	0.61391	0.00714	0.04935
11	0.00845	0.06558	0.58468	0.00777	0.05712
12	0.00919	0.07477	0.55684	0.00845	0.06558
13	0.00998	0.08475	0.53032	0.00919	0.07477
14	0.01084	0.09559	0.50507	0.00998	0.08475
15	0.01175	0.10735	0.48102	0.01084	0.09559
16	0.01273	0.12008	0.45811	0.01175	0.10735
17	0.01378	0.13386	0.43630	0.01273	0.12008
18	0.01489	0.14875	0.41552	0.01378	0.13386
19	0.01606	0.16481	0.39573	0.01489	0.14875
20	0.01730	0.18212	0.37689	0.01606	0.16481
21	0.01860	0.20072	0.35894	0.01730	0.18212
22	0.01996	0.22068	0.34185	0.01860	0.20072
23	0.02136	0.24204	0.32557	0.01996	0.22068
24	0.02281	0.26484	0.31007	0.02136	0.24204
25	0.02428	0.28912	0.29530	0.02281	0.26484
26	0.02577	0.31489	0.28124	0.02428	0.28912
27	0.02725	0.34214	0.26785	0.02577	0.31489
28	0.02871	0.37085	0.25509	0.02725	0.34214
29	0.03012	0.40097	0.24295	0.02871	0.37085
30	0.03146	0.43243	0.23138	0.03012	0.40097
31	0.03269	0.46512	0.22036	0.03146	0.43243
32	0.03378	0.49890	0.20987	0.03269	0.46512
33	0.03469	0.53359	0.19987	0.03378	0.49890
34	0.03538	0.56897	0.19035	0.03469	0.53359
35	0.03582	0.60479	0.18129	0.03538	0.56897
36	0.03596	0.64075	0.17266	0.03582	0.60479
37	0.03578	0.67653	0.16444	0.03596	0.64075
38	0.03525	0.71179	0.15661	0.03578	0.67653
39	0.03435	0.74614	0.14915	0.03525	0.71179
40	0.03307	0.77921	0.14205	0.03435	0.74614
41	0.03142	0.81063	0.13528	0.03307	0.77921
42	0.02942	0.84005	0.12884	0.03142	0.81063
43	0.02711	0.86716	0.12270	0.02942	0.84005

$K(x)$	$Pr[K(50)=k]$	$Pr[K(50) \leq k]$	v^k	$Pr(Z=v^k)$	$Pr[Z \geq v^k]$
44	0.02454	0.89170	0.11686	0.02711	0.86716
45	0.02179	0.91349	0.11130	0.02454	0.89170
46	0.01894	0.93243	0.10600	0.02179	0.91349
47	0.01608	0.94851	0.10095	0.01894	0.93243
48	0.01330	0.96182	0.09614	0.01608	0.94851
49	0.01070	0.97252	0.09156	0.01330	0.96182
50	0.00834	0.98086	0.08720	0.01070	0.97252
51	0.00628	0.98714	0.08305	0.00834	0.98086
52	0.00456	0.99170	0.07910	0.00628	0.98714
53	0.00317	0.99487	0.07533	0.00456	0.99170
54	0.00211	0.99698	0.07174	0.00317	0.99487
55	0.00133	0.99831	0.06833	0.00211	0.99698
56	0.00080	0.99911	0.06507	0.00133	0.99831
57	0.00045	0.99956	0.06197	0.00080	0.99911
58	0.00024	0.99980	0.05902	0.00045	0.99956
59	0.00012	0.99991	0.05621	0.00024	0.99980
60	0.00005	0.99997	0.05354	0.00012	0.99991
61	0.00002	0.99999	0.05099	0.00005	0.99997
62	0.00001	1.00000	0.04856	0.00002	0.99999
63	0.00000	1.00000	0.04625	0.00001	1.00000
64	0.00000	1.00000	0.04404	0.00000	1.00000
65	0.00000	1.00000	0.04195	0.00000	1.00000
66	0.00000	1.00000	0.03995	0.00000	1.00000
67	0.00000	1.00000	0.03805	0.00000	1.00000
68	0.00000	1.00000	0.03623	0.00000	1.00000
69	0.00000	1.00000	0.03451	0.00000	1.00000
70	0.00000	1.00000	0.03287	0.00000	1.00000
71	0.00000	1.00000	0.03130	0.00000	1.00000
72	0.00000	1.00000	0.02981	0.00000	1.00000
73	0.00000	1.00000	0.02839	0.00000	1.00000
74	0.00000	1.00000	0.02704	0.00000	1.00000
75	0.00000	1.00000	0.02575	0.00000	1.00000
76	0.00000	1.00000	0.02453	0.00000	1.00000
77	0.00000	1.00000	0.02336	0.00000	1.00000
78	0.00000	1.00000	0.02225	0.00000	1.00000
79	0.00000	1.00000	0.02119	0.00000	1.00000
80	0.00000	1.00000	0.02018	0.00000	1.00000

Table 3: Table of $Pr[K(50)=k]$ and $Pr(Z=v^k)$ values for Question 3(a)(i)(ii)

(iii) The theoretical mean and standard deviation of Z are calculated below.

```
> Ax <- function(x, s, i, moment) {
+   i <- exp(moment * log(1+i)) - 1
+   k <- 0:(omega-x-1)
```

```
+   pk <- udeferredtqx(k, 1, x, s)
+   vk <- v(i, k+1)
+   sum(pk * vk)
+ }
> ExpectedValueZ <- Ax(50, 0, i, 1)
> StandardDeviationZ <- sqrt(Ax(50, 0, i, 2) - Ax(50, 0, i, 1)^2)
```

This gives $E(Z) = 0.24618$ and $SD(Z) = 0.16039$.

(iv) The probability that Z exceeds $A_{[50]}$ is calculated below.

```
> Fz <- function(z, x, s, i) {
+   k <- floor(uniroot(function(t) v(i,t) - z,
+                      interval = c(0, omega - x - 1))$root)
+   1 - sum(udeferredtqx(k:(omega-x-1), 1, x, s))
+ }
> FExpectedValueZ <- Fz(ExpectedValueZ, 50, 0, i)
```

The gives $Pr(Z > A_{[50]}) = 0.34214$.

(b) (i) The code used to simulate $K(50)$ is shown below.

```
> SimulateK <- function(n, x, omega) {
+   k <- 0:(omega-x-1)
+   pk <- udeferredtqx(k, 1, x, 0)
+   Fk <- cumsum(pk)
+
+   u <- runif(n)
+   sapply(u, function(y) head(k[Fk > y], 1)) + 1
+ }
> set.seed(2000)
> kvalues <- SimulateK(50, 50, omega)
```

(ii) The values of Z can be computed from the values of $K(50)$ as shown below.

```
> zvalues <- v(i, kvalues)
```

(iii) The histogram of Z is shown in Figure 2.

```
> hist(zvalues, main = "", xlab = "Simulated value of Z")
```

(iv) The empirical mean and standard deviation of Z are computed below.

```
> EstimatedExpectedValueZ <- mean(zvalues)
> EstimatedStandardDeviationZ <- sd(zvalues)
```

This gives $E(Z) \approx 0.23836$ and $SD(Z) \approx 0.15301$.

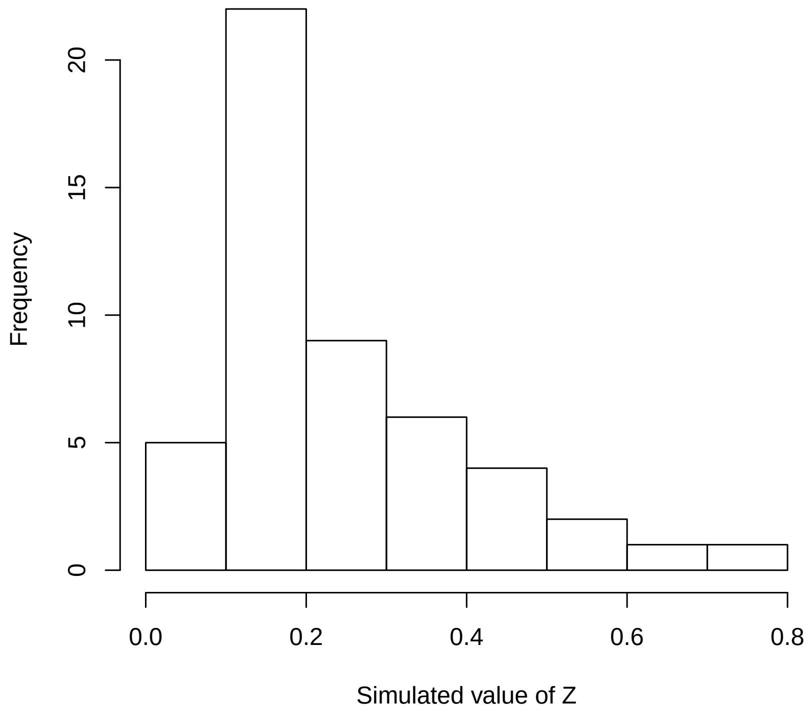
(v) The probability that Z exceeds the theoretical mean is estimated value.

```
> EstimatedFExpectedValueZ <- sum(zvalues > ExpectedValueZ) / 50
```

This gives $Pr(Z > A_{[50]}) \approx 0.32$.

(c) The code needed to produce the outputs in part (b) with $n = 500$ is shown below. The histogram is shown in Figure 3.

Figure 2: Histogram of simulated values of Z with $n = 50$ for Question 3(b)(iii)




```
> kvalues <- SimulateK(500, 50, omega)
> zvalues <- v(i, kvalues)
> hist(zvalues, main = "", xlab = "Simulated value of Z")
> EstimatedExpectedValueZ <- mean(zvalues)
> EstimatedStandardDeviationZ <- sd(zvalues)
> EstimatedFExpectedValueZ <- sum(zvalues > ExpectedValueZ) / 500
```

This gives $E(Z) \approx 0.2493$, $SD(Z) \approx 0.16111$ and $Pr(Z > A_{[50]}) \approx 0.348$.

Nathan Esau
301197568

Figure 3: Histogram of simulated values of Z with $n = 500$ for Question 3(c)(iii)