Force of Mortality

The force of mortality is defined as follows:

$$ {}_tp_x = exp\left\{ -\int_0^t \mu_{x+s} ds \right\} $$
$$ \mu_x = A + Bc^x $$

where

$$ A = 0.00022,\ B = 2.7x10^{-6},\ c = 1.124 $$

If we are just talking about Standard Ultimate Survival Model, the interest rate is $i = 0.05$. Note that

$$ \mu_{x+s} = A + Bc^{x+s} $$

So integrating this, we get

$$ -\int_0^t \mu_{x+s} ds = -\left[\frac{Bcx^{x+t} + Atln(c) - Bc^x}{ln(c)} \right] $$

So

$$ {}_tp_x = exp\left\{ -\left[\frac{Bc^{x+t} + Atln(c) - Bc^x}{ln(c)} \right] \right\} $$

Just define this as a function of t and x, which we can call $p(t, x)$.

In [34]:

```r
p <- function(t, x, type = 'Makeham')
{
    if(type == 'Makeham')
    {
        A <- 0.00022
        B <- 2.7*10^(-6)
        c <- 1.124
    }
    else if(type == 'Gompertz')
    {
        A <- 0
        B <- 0.0003
        c <- 1.07
    }
    numerator <- -1*(B*c^(x + t) + A*t*log(c) - B*c^x)
    denominator <- log(c)
    return(exp(numerator/denominator))
}
```

I defaulted the method to Makeham, but in rare instances, Gompertz may be needed. Let us put this to the test. On page 48, Example 3.6, we are asked to find

$$ \begin{aligned} {}_{0.4}q_{40.2} &= 1 - {}_{0.4}p_{40.2} \\ &= 1 - p(0.4, 40.2) \end{aligned} $$

In [36]:
```
1 - p(0.4, 40.2)
```

0.000209434992868185

Let us also calculate p_{40} from the same page.

In [37]:
```
p(1, 40)
```

0.999472779557205

Also calculate ${}_{0.7}q_{70.6}$

In [38]:
```
1 - p(0.7, 70.6)
```

0.00768313065738091

Proper Limits of Integration

Throughout MATH 471, you're just gonna be working with Makeham's Law with the interest rate i defaulted at 0.05. Plus, whenever you see an integration from 0 to ∞, you know it doesn't actually go to ∞. Just keep in mind the following

$$x + t \leq 120$$

120 is the magic number. When talking about insurance, your current age x and any additional years you will age t must be no more than 120 when added. So for example, on page 29,

$$\overset{o}{e}_x = \int_0^\infty {}_tp_x \, dt$$

you are not actually integrating it from 0 to ∞ but from 0 to $120 - x$(solve the above inequality for t), so the right way of integrating this is

$$\overset{o}{e}_x = \int_0^{120-x} {}_tp_x \, dt$$

. This is the same for every other functions you see that is being integrated all the way to ∞. Let us define this for the case of Standard Ultimate Survival Model since we are already at it. We will define $\overset{o}{e}_x$ as eo(x, type) where type = 'Makeham' or type = 'Gompertz'. Let us also define e_x as e(x, type). Recall

$$e_x = \sum_{k=1}^{\infty} {}_kp_x$$

For Gompertz' law, the upper limit of sum is $130 - x$ instead of $120 - x$.

In [70]:

```
eo <- function(x, ...)
{
    p1 <- function(t) p(t, x, ...)
    return(integrate(p1, 0, 120 - x)$value)
}

e <- function(x, ...)
{
    k <- 1:(130 - x)
    return(sum(p(k, x, ...)))
}
```

Let's replicate Table 2.2 on page 35.

In [79]:

```
x <- seq(from = 0, to = 100, by = 10)
index <- 1
E <- c()
Eo <- c()
for(i in x)
{
    E[index] <- e(i, 'Gompertz')
    Eo[index] <- eo(i, 'Gompertz')
    index <- index + 1
}
data.frame(x, E, Eo)
```

x	E	Eo
0	71.437538	71.937513
10	61.722842	62.222793
20	52.202974	52.702877
30	42.992149	43.491959
40	34.251927	34.751553
50	26.191880	26.691143
60	19.051899	19.550450
70	13.057704	13.554854
80	8.354054	8.848447
90	4.943593	5.432562
100	2.673255	3.151554

Generating a Table for Standard Ultimate Survival Model

Hold x fixed because that is your current age and let t, the additional number you age, vary.

In [28]:

```
SurvivalTable <- function(x)
{
    t1 <- 0
    t <- c()
    index <- 1
    while(x + t1 <= 120)
    {
        t[index] <- t1
        t1 <- t1 + 1
        index <- index + 1
    }
    P <- p(t, x)
    Q <- 1 - P
    return(data.frame(t, P, Q))
}
```

In [29]:

```
SurvivalTable(20)
```

t	P	Q
0	1.0000000	0.000000000
1	0.9997504	0.000249639
2	0.9994971	0.000502893
3	0.9992398	0.000760215
4	0.9989779	0.001022114
5	0.9987108	0.001289162
6	0.9984380	0.001562002
7	0.9981586	0.001841356
8	0.9978720	0.002128035
9	0.9975771	0.002422950
10	0.9972729	0.002727125
11	0.9969583	0.003041711
12	0.9966320	0.003367999
13	0.9962926	0.003707440
14	0.9959383	0.004061666
15	0.9955675	0.004432507
16	0.9951780	0.004822019
17	0.9947675	0.005232511
18	0.9943334	0.005666578
19	0.9938729	0.006127130
20	0.9933826	0.006617437
21	0.9928588	0.007141169
22	0.9922976	0.007702444
23	0.9916941	0.008305886
24	0.9910433	0.008956681
25	0.9903394	0.009660648
26	0.9895757	0.010424316
27	0.9887450	0.011255003
28	0.9878391	0.012160914
29	0.9868488	0.013151248
...	...	...
71	3.761856e-01	0.6238144
72	3.337988e-01	0.6662012
73	2.918378e-01	0.7081622

t	P	Q
74	2.509433e-01	0.7490567
75	2.117830e-01	0.7882170
76	1.750176e-01	0.8249824
77	1.412589e-01	0.8587411
78	1.110253e-01	0.8889747
79	8.469734e-02	0.9153027
80	6.248174e-02	0.9375183
81	4.438803e-02	0.9556120
82	3.022582e-02	0.9697742
83	1.962493e-02	0.9803751
84	1.207789e-02	0.9879221
85	6.999142e-03	0.9930009
86	3.790784e-03	0.9962092
87	1.902838e-03	0.9980972
88	8.769402e-04	0.9991231
89	3.671406e-04	0.9996329
90	1.379805e-04	0.9998620
91	4.593174e-05	0.9999541
92	1.334088e-05	0.9999867
93	3.324214e-06	0.9999967
94	6.972232e-07	0.9999993
95	1.204912e-07	0.9999999
96	1.674984e-08	1.0000000
97	1.823127e-09	1.0000000
98	1.507298e-10	1.0000000
99	9.148468e-12	1.0000000
100	3.923016e-13	1.0000000

Notice that rows $30 - 70$ are missing. R is pretty lazy sometimes and will omit certain rows if the data frame or matrix is longer than it wants. But this is easy. Just fill the missing rows manually.

In [31]:

```
t <- 30:70
P <- p(t, 20)
Q <- 1 - P
data.frame(t, P, Q)
```

t	P	Q
30	0.9857637	0.01423631
31	0.9845724	0.01542763
32	0.9832619	0.01673813
33	0.9818177	0.01818227
34	0.9802238	0.01977622
35	0.9784620	0.02153803
36	0.9765121	0.02348789
37	0.9743517	0.02564831
38	0.9719556	0.02804441
39	0.9692959	0.03070414
40	0.9663414	0.03365864
41	0.9630575	0.03694247
42	0.9594060	0.04059401
43	0.9553443	0.04465575
44	0.9508253	0.04917468
45	0.9457973	0.05420266
46	0.9402033	0.05979672
47	0.9339805	0.06601948
48	0.9270606	0.07293942
49	0.9193688	0.08063118
50	0.9108243	0.08917571
51	0.9013396	0.09866043
52	0.8908209	0.10917909
53	0.8791684	0.12083163
54	0.8662764	0.13372361
55	0.8520346	0.14796542
56	0.8363289	0.16367111
57	0.8190434	0.18095664
58	0.8000623	0.19993766
59	0.7792735	0.22072650
60	0.7565716	0.24342840
61	0.7318631	0.26813689

t	P	Q
62	0.7050719	0.29492806
63	0.6761460	0.32385398
64	0.6450650	0.35493499
65	0.6118488	0.38815118
66	0.5765668	0.42343316
67	0.5393473	0.46065268
68	0.5003865	0.49961351
69	0.4599564	0.54004359
70	0.4184105	0.58158948

In [ ]:

```
1
```