

Non-life — Assignment 1B

Y^* and Z^\dagger

September 18, 2016

Makeham and Gompertz survival distributions

Q1

Let $Y = \log(1 + \frac{V \log(c)}{b}) / \log(c) = \log_c(1 + \frac{V \log(c)}{b})$ and $V \sim \text{exponential}(1)$. Then a straightforward calculation gives the result:

$$c^Y - 1 = c^{\log_c(1 + \frac{V \log(c)}{b})} - 1 = 1 + \frac{V \log(c)}{b} - 1 = \frac{V \log(c)}{b} \quad (1)$$

and because $V \sim \text{exponential}(1)$ we have that

$$P[c^Y - 1 \leq x] = P[V \frac{\log(c)}{b} \leq x] = P[V \leq x \frac{b}{\log c}] \quad (2)$$

so that $c^Y - 1 \sim \text{exponential}(\frac{b}{\log(c)})$. Then using the explicit form of the CDF of the exponential distribution we find using $Z = c^Y - 1$

$$\begin{aligned} P[Y > x] &= P[c^Y - 1 > c^x - 1] \\ &= P[Z > c^x - 1] = 1 - P[Z \leq c^x - 1] \\ &= 1 - F_Z(c^x - 1) = 1 - (1 - \exp(-\frac{b}{\log c}(c^x - 1))) \\ &= \exp(-\frac{b}{\log c}(c^x - 1)) \end{aligned} \quad (3)$$

which is equal to equation (2) from the exercises with $a = 0$ so $Y \sim \text{Gompertz}(b, c)$.

Q2

If the following code is entered into Rstudio

```
gen.Sample <- function(n, a, b, c)
{if (any(a<0,b<0,c<1)) stop("Invalid parameters")
  lifetimes <- log(1+rexp(n)*log(c)/b)/log(c)
  if (a>0) lifetimes <- pmin(lifetimes, rexp(n)/a)
  return(lifetimes)}
```

*Student number: 00000000

†Student number: 99999999

```

set.seed(2525); G <- gen.Sample(2000, 0, 8e-5, 1.08)
set.seed(2525); M <- gen.Sample(2000, 5e-4, 8e-5, 1.08)
all(G>=M)
mean(M==G)
rbind(Gompertz=summary(G), Makeham=summary(M))

```

we obtain the output:

```

> all(G>=M)
[1] TRUE
> mean(M==G)
[1] 0.959
> rbind(Gompertz=summary(G), Makeham=summary(M))
      Min. 1st Qu. Median  Mean 3rd Qu.  Max.
Gompertz 4.9450   73.72  85.12 81.98   93.73 116.6
Makeham   0.2809   71.85  84.15 80.08   93.35 116.6

```

Since the Makeham distribution is defined as $X = \min(Y, Z)$ with $Y \sim \text{Gompertz}(b, c)$ and $Z \sim \text{exponential}(a)$ the results of the Makeham distribution are always smaller or equal to the Gompertz results. This is verified by the result `all(G>M)` is TRUE.

The result of `mean(M==G)` is 0.959 which implies that in 95,9% of the cases the cause of death is equal to the Gompertz results. The probability of dying of old ages with these parameters is therefore 0.959.

Q3

We run the following code to assess the effect of changes on the parameters on the mean life-time.

```

S <- function(x, a, b, c) exp(-a*x - b/log(c)*(c^x-1))
a <- 5e-4; b <- 8e-5; c <- 1.09
mean.age <- integrate(S, 0, Inf, a, b, c)$value
mean(gen.Sample(1e6, a, b, c))
integrate(S, 0, Inf, 1.02*a, b, c)$value / mean.age
integrate(S, 0, Inf, a, 1.02*b, c)$value / mean.age
integrate(S, 0, Inf, a, b, 1.02*(c-1)+1)$value / mean.age

```

and obtain the following results

```

> integrate(S, 0, Inf, 1.02*a, b, c)$value / mean.age
[1] 0.9996164
> integrate(S, 0, Inf, a, 1.02*b, c)$value / mean.age
[1] 0.9969848
> integrate(S, 0, Inf, a, b, 1.02*(c-1)+1)$value / mean.age
[1] 0.9844023

```

The results imply that a 2% increase in the parameter a leads to a very small percentage decline of the mean life-time of approximately 0.04%. The impact of a 2% increase in the b parameter leads to a slightly larger decrease on the mean life-time of approximately 0.3%. The impact of the scaling of the c parameter leads to approximately a 1.6% decrease of the mean life-time. Increasing the parameters therefore leads to an increase in mortality.

Q4

By using the logarithm of the likelihood function we evade the problem of limited machine precision. If we let R calculate the following lines

```
exp(-8300)
exp(-10000)
exp(-5000)
exp(-100)
exp(-10)
```

we obtain the answers

```
> exp(-8300)
[1] 0
> exp(-10000)
[1] 0
> exp(-5000)
[1] 0
> exp(-100)
[1] 3.720076e-44
> exp(-10)
[1] 4.539993e-05
```

which implies that R does not have the capacity to differentiate between the numbers $\exp(-10000)$, $\exp(-8300)$ and $\exp(-5000)$ for example. Only if the exponent is sufficiently small R will have the ability to differentiate between the numbers and find an optimum. By passing to the logarithm of the likelihood function we circumvent this problem.

Q5

To obtain ML-estimates for the M sample the code given in the exercise is the following:

```
log.Lik <- function(p) {- sum(log.fx(M, p[1], p[2], p[3]))} ## Makeham on Makeham
a <- 5e-4; b <- 8e-5; c <- 1.08
oMM <- optim(c(a,b,c), log.Lik)
oMM$value ## 8428.489
oMM$par ## a=7.3e-04 b=6.0e-05 c=1.083
```

```
log.Lik <- function(p) {-sum(log.fx(M, 0, p[1], p[2]))} ## Gompertz on Makeham
a <- 0; b <- 8e-5; c <- 1.08
oGM <- optim(c(b,c), log.Lik) ## 13x NaN
oGM$value ## 8456.258
oGM$par ## a=0 b=1.3e-4 c=1.074
```

with the following output

```

> log.Lik <- function(p) {-sum(log.fx(M, p[1], p[2], p[3]))} ## Makeham on Makeham
> a <- 5e-4; b <- 8e-5; c <- 1.08
> oMM <- optim(c(a,b,c), log.Lik)
There were 26 warnings (use warnings() to see them)
> oMM$value ## 8428.489
[1] 8428.489
> oMM$par ## a=7.3e-04 b=6.0e-05 c=1.083
[1] 7.338301e-04 5.993626e-05 1.083515e+00
>
> log.Lik <- function(p) {-sum(log.fx(M, 0, p[1], p[2]))} ## Gompertz on Makeham
> a <- 0; b <- 8e-5; c <- 1.08
> oGM <- optim(c(b,c), log.Lik) ## 13x NaN
There were 13 warnings (use warnings() to see them)
> oGM$value ## 8456.258
[1] 8456.258
> oGM$par ## a=0 b=1.3e-4 c=1.074
[1] 0.0001292634 1.0741178742
>

```

To find the ML-estimates for the G sample we use the following code

```

log.Lik <- function(p) {-sum(log.fx(G, p[1], p[2], p[3]))} ## Makeham on Gompertz
a <- 5e-4; b <- 8e-5; c <- 1.08
oMG <- optim(c(a,b,c), log.Lik) ## 13x NaN
oMG$value
oMG$par

log.Lik <- function(p) {-sum(log.fx(G,0, p[1], p[2]))} ## Gompertz on Gompertz
a <- 0; b <- 8e-5; c <- 1.08
oGG <- optim(c(b,c), log.Lik) ## 13x NaN
oGG$value
oGG$par

```

With results

```

> log.Lik <- function(p) {-sum(log.fx(G, p[1], p[2], p[3]))} ## Makeham on Gompertz
> a <- 5e-4; b <- 8e-5; c <- 1.08
> oMG <- optim(c(a,b,c), log.Lik) ## 13x NaN
There were 33 warnings (use warnings() to see them)
> oMG$value
[1] 8245.877
> oMG$par
[1] 1.585592e-04 5.928508e-05 1.083783e+00
>
> log.Lik <- function(p) {-sum(log.fx(G,0, p[1], p[2]))} ## Gompertz on Gompertz
> a <- 0; b <- 8e-5; c <- 1.08
> oGG <- optim(c(b,c), log.Lik) ## 13x NaN
There were 13 warnings (use warnings() to see them)

```

```
> oGG$value
[1] 8248.117
> oGG$par
[1] 7.088547e-05 1.081598e+00
```

Summarizing the output we have the following results

	a	b	c	ML
Makeham on Makeham sample	7.338301e-04	5.993626e-05	1.083515e+00	8428.489
Gompertz on Makeham sample	0	0.0001292634	1.0741178742	8456.258
Makeham on Gompertz sample	1.585592e-04	5.928508e-05	1.083783e+00	8245.877
Gompertz on Gompertz sample	0	7.088547e-05	1.081598e+00	8248.117

Q6

We have the following formulas for the AIC

$$AIC = -2 \log L + 2k \quad (4)$$

and for the BIC criteria

$$BIC = -2 \log L + k \log n \approx -2 \log L + 7.6k \quad (5)$$

where the approximation is for the current situation. For the Makeham model there are 3 paramters ($k = 3$) and for the Gompertz model there are 2 parameters ($k = 2$).

Via R we compute the values for the AIC and BIC criteria in the optima for the different situations and test if the Makeham model has a smaller AIC or BIC

```
AIC.MM <- 2*oMM$value + 2 * 3;
AIC.GM <- 2*oGM$value + 2 * 2;
AIC.MG <- 2*oMG$value + 2 * 3;
AIC.GG <- 2*oGG$value + 2 * 2;

BIC.MM <- 2*oMM$value + 7.6 * 3;
BIC.GM <- 2*oGM$value + 7.6 * 2;
BIC.MG <- 2*oMG$value + 7.6 * 3;
BIC.GG <- 2*oGG$value + 7.6 * 2;

(AIC.MM<AIC.GM)
(AIC.MG<AIC.GG)
(BIC.MM<BIC.GM)
(BIC.MG<BIC.GG)
```

which gives the following output

```
> (AIC.MM<AIC.GM)
[1] TRUE
> (AIC.MG<AIC.GG)
[1] TRUE
> (BIC.MM<BIC.GM)
```

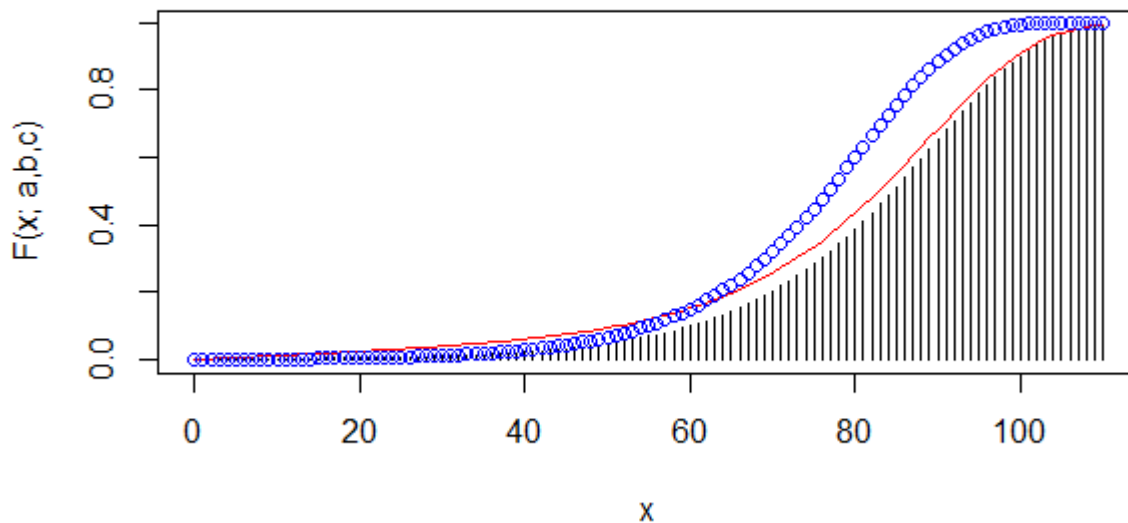


Figure 1: The bar chart describes a Gompertz distribution with $b = 8e-5$ and $c=1.08$. The red line is a Makeham distribution with $a=1e-3$ and b,c equal to the Gompertz distribution. The blue dots represent the Gompertz distribution with the parameter $c = 1.09$.

```
[1] TRUE
> (BIC.MG<BIC.GG)
[1] FALSE
```

As regards to AIC the Makeham model is better on both samples. As regards to the BIC the Makeham model is better on the Makeham sample but the Gompertz model is better on the Gompertz sample.

Q7

If the following code is submitted in R

```
S <- function(x, a, b, c) exp(-a*x - b/log(c)*(c^x-1))
x <- 0:110
plot(x, 1-S(x, 0, 8e-5, 1.08), type="h", ylab="F(x; a,b,c)")
lines(x, 1-S(x, 1e-3, 8e-5, 1.08), col="red")
points(x, 1-S(x, 0, 8e-5, 1.09), col="blue")
```

R generates the plot as seen in Figure 1.

The difference between the Gompertz (bars in Figure) and Makeham distribution (red line) can clearly be seen in the figure. Since Makeham also has a 'accident' part in the model the cdf shows higher probabilities at low ages. For old ages the two distribution converge and are approximately equal. The blue dots represent a Gompertz distribution with a higher parameter c . The effect of increasing the c parameter leads to an increase in the cdf especially for older ages.

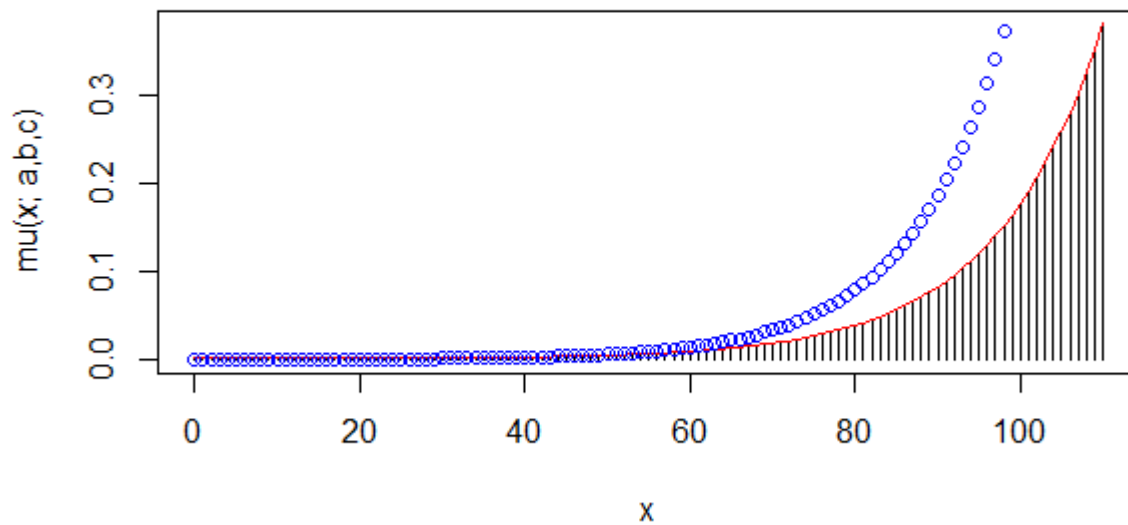


Figure 2: The bar chart describes the mortality rate for Gompertz' law with $b = 8e-5$ and $c=1.08$. The red line is the Makeham extension with $a=1e-3$ and b,c equal to the Gompertz case. The blue dots represent the Gompertz case with the parameter $c = 1.09$.

Q8

If the following code is submitted in R

```
mu <- function(x, a, b, c) (a + b*c^x)
x <- 0:110
plot(x, mu(x, 0, 8e-5, 1.08), type="h", ylab="F(x; a,b,c)")
lines(x, mu(x, 1e-3, 8e-5, 1.08), col="red")
points(x, mu(x, 0, 8e-5, 1.09), col="blue")
```

the output from R is given in figure 2.

The Makeham mortality rate with $a = 1e - 3$ is almost indistinguishable from the Gompertz mortality rate. Changing the parameter c to 1.09 has a large impact on the mortality rate. At ages higher than 70 years old the mortality rate increases dramatically.

Q9

Since $q_x = \frac{S(x-1)-S(x)}{S(x-1)}$ we can fill the dots in the code in the following way using the `diff(c)` and `head()` functions and optimize

```
S <- function(x, a, b, c) exp(-a*x - b/log(c)*(c^x-1))
agerange <- 1:nages ## denotes the range of ages accounted for when finding ML
```

```

LogLik <- function(p)
{ ddfs <- S(0:nages, p[1], p[2], p[3])
q.x <- -diff(ddfs)/head(ddfs,nages)
-sum(dbinom(D.x[agerange], e.x[agerange], q.x[agerange], log=TRUE))}
a <- 1e-4; b <- 8e-5; c <- 1.08
o <- optim(c(a,b,c), LogLik)
o$value
o$par

```

which gives the following results

There were 22 warnings (use warnings() to see them)

```

> o$value
[1] 172666.1
> o$par
[1] 1.076124e-03 1.433849e-06 1.166235e+00

```

so the ML-estimates for a,b and c are $a = 1.077e - 3$, $b = 1,433e - 6$ and $c = 1.166$.
Submitting the code

```

ddfs <- S(0:nages, o$par[1], o$par[2], o$par[3])
q.x <- -diff(ddfs)/head(ddfs,nages)

plot(1:nages,D.x[1:nages]/e.x[1:nages],type="p",
     main="Optimally estimated qx & fractions Dx/ex",xlab="age",ylab="q_x")
lines(1:nages, q.x[1:nages], col="blue")

```

gives the plot in figure 3.

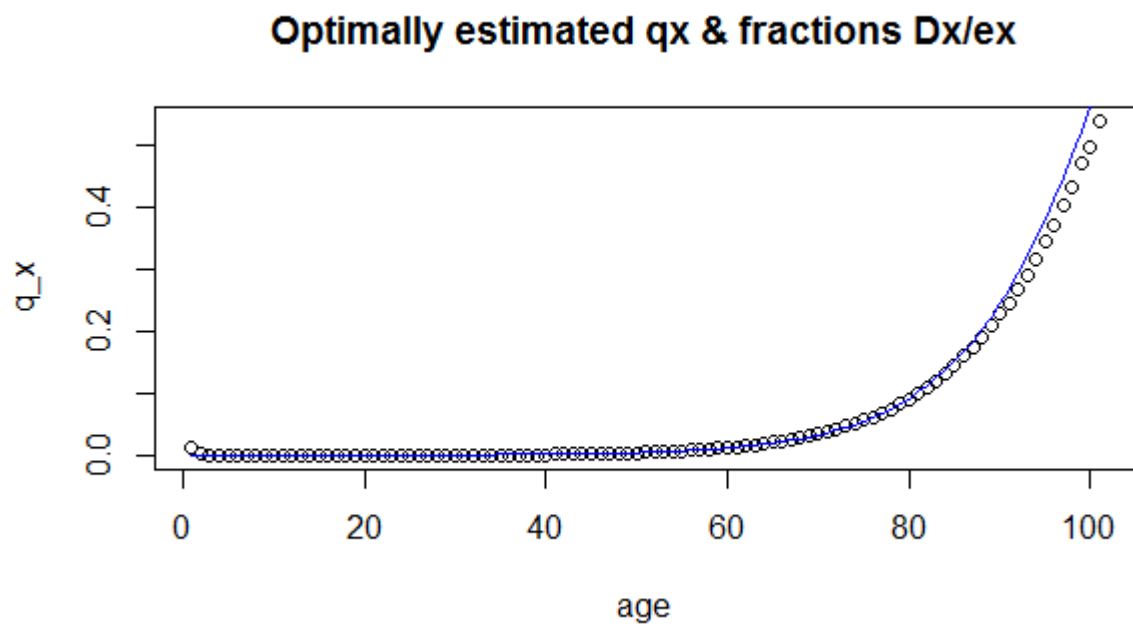


Figure 3: Plot of the ML estimate and the fractions D_x/e_x