Gompertz Maximum Likelihood Estimation of Truncated Death Distribuitons

Joshua R. Goldstein

November 8, 2019

Big Picture

Our challenge

We see only part of the picture (e.g., deaths aged 70 to 87).

No estimates of who died before or after

- How can we estimate death rates without denominators?
- How can we estimate $e(65)$ differences between groups?

Our idea

We can combine

- observed distribution of deaths (over limited range)
- our external knowledge of human mortality age-patterns

The hope is that this will produce good estimates of mortality rates, of $e(65)$, and of differences between groups

Today's agenda

- Intro to Maximum Likelihood Estimation
- Example with simulated data
- Attempt at validation with HMD
- Preliminary try at NUMIDENT
- Lessions and directions

Truncated Maximum Likelihood (in theory)

Philosophy

For given data X, we can a likelihood associated with a particular value of parameter θ.

We then choose the $\hat{\theta}$ to maximize this likelihood.

A simple example

Likelihood for observation i with value x_i:

$$L_i = L(\lambda|x_i) = f_\lambda(x_i)$$

Likelihood for all observations:

$$L = \prod_i L_i$$

Log-likelihood:

$$\mathcal{L} = \sum_i \log L_i$$

If we observe $x_1 = 3$ and $x_2 = 5$, then

$$L(\lambda|x_1) = \lambda e^{-3\lambda}$$
$$L(\lambda|x_2|) = \lambda e^{-5\lambda}$$
$$L(\lambda|x_1, x_2) = \lambda^2 e^{(} - 8\lambda)$$

$$\mathcal{L} = \sum_i \log L_i = 2 \log \lambda - 8\lambda$$

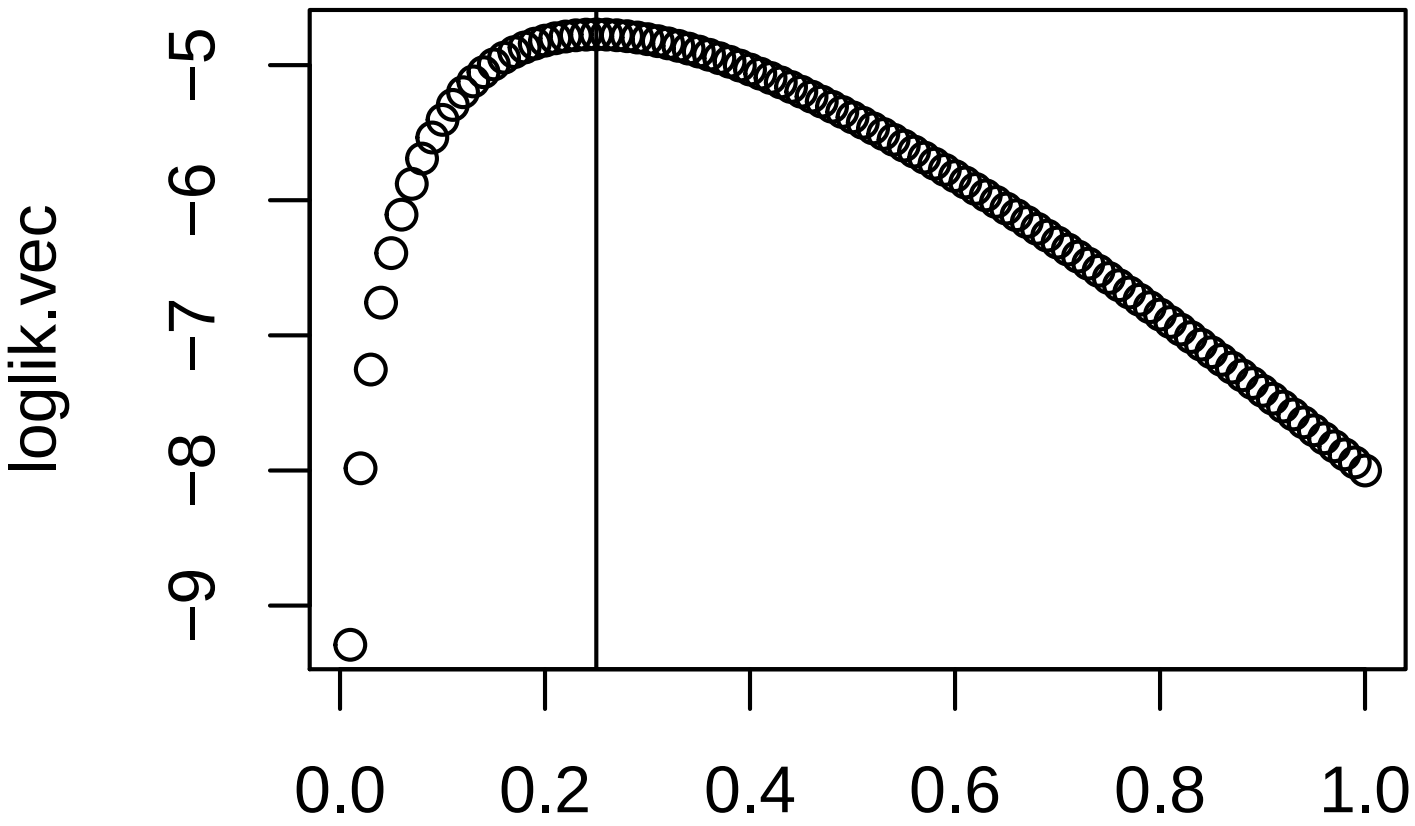
$$\frac{d\mathcal{L}}{d\lambda} = \frac{2}{\lambda} - 8 = 0$$

So,

$$\hat{\lambda}_{MLE} = 2/8 = 0.25$$

We did this by hand, but can also do with the computer

```
lambda.vec <- seq(.01, 1, .01)
loglik.vec = 2 * log(lambda.vec) - 8 * lambda.vec
plot(lambda.vec, loglik.vec)
abline(v = lambda.vec[which.max(loglik.vec)])
```



For truncated distribution we observe only from a to b

We can define the conditional distribution

$$f_{trunc} = \frac{f_\theta(x)}{\int_a^b f_\theta(x)\,dx} = \frac{f_\theta(x)}{F_\theta(b) - F_\theta(a)}$$

with likelihood

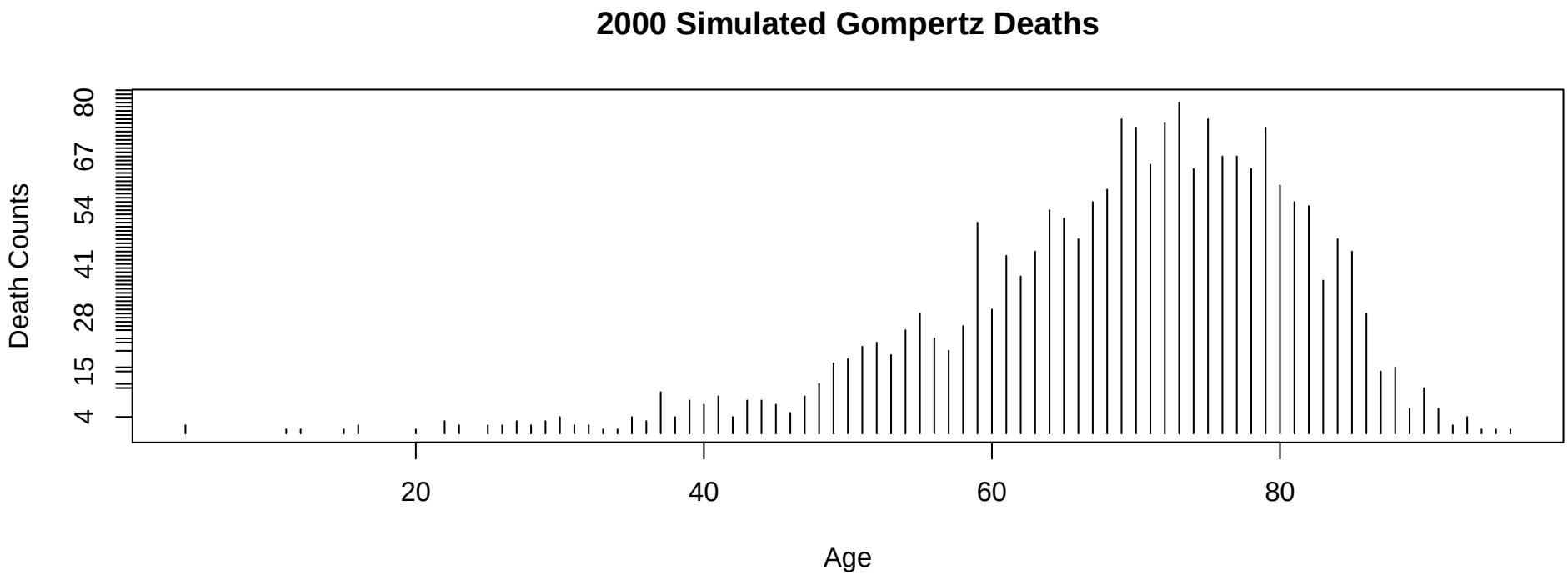
$$L(\theta|\mathbf{x}) = \prod \frac{f_\theta(x_i)}{F_\theta(b) - F_\theta(a)}$$

And then we maximize that.

Validation with Simulated Gompertz

Simulated Gompertz, without truncation

```
source("hmd_validation_functions.R")
N = 2000
set.seed(13)
x <- rgompertz.M(N, b = 1/10, M = 75)
Dx <- table(floor(x))
plot(names(Dx), Dx, type = "h",
     ylab = "Death Counts", xlab = "Age",
     main = "2000 Simulated Gompertz Deaths")
```



MLE estimation

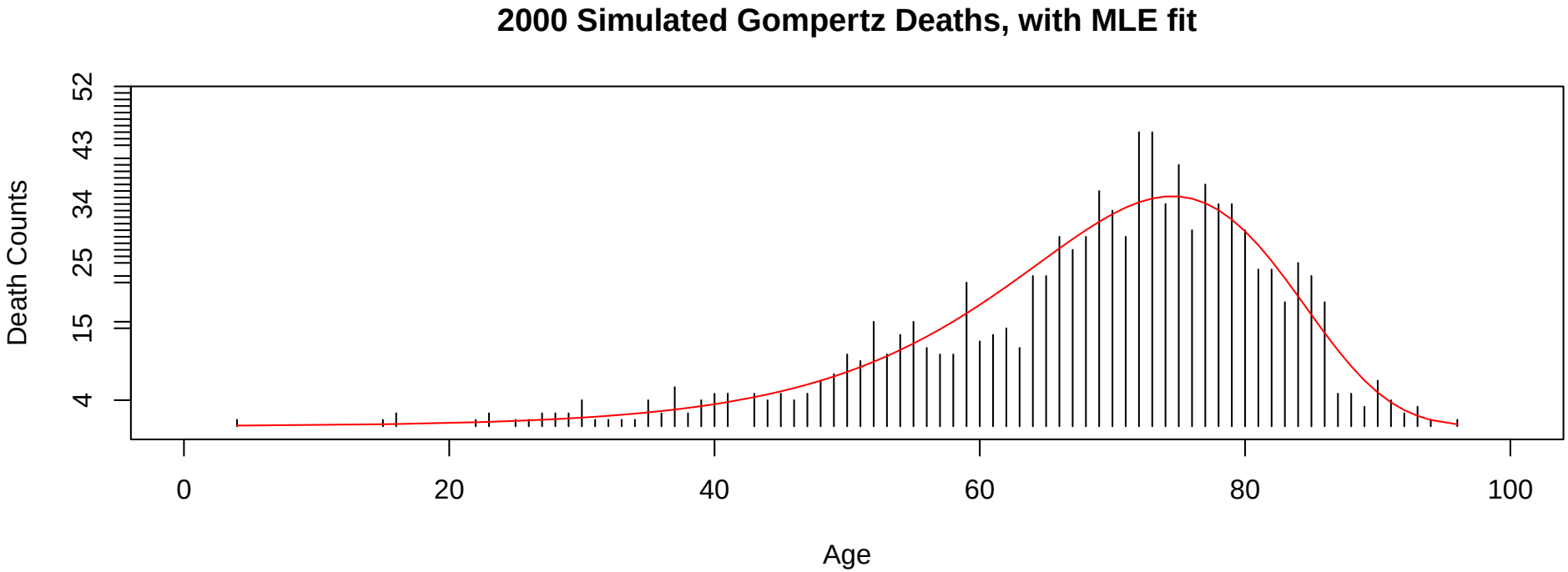
```
source("hmd_validation_functions.R")
N = 1000
set.seed(13)
x <- rgompertz.M(N, b = 1/10, M = 75)
Dx <- table(floor(x))
fit <- counts.trunc.gomp.est(Dx= Dx, x.left = 0, x.right = 200,
                             b.start = 1/9, M.start = 80)
(b.hat = exp(fit$par[1]))
```

```
## [1] 0.09572687
```

```
(M.hat = exp(fit$par[2]))
```

```
## [1] 75.01612
```

How did we do?


2000 Simulated Gompertz Deaths, with MLE fit
Death Counts
4
15
25
34
43
52
0
20
40
60
80
100
Age

Now artificially truncate to ages 65-90

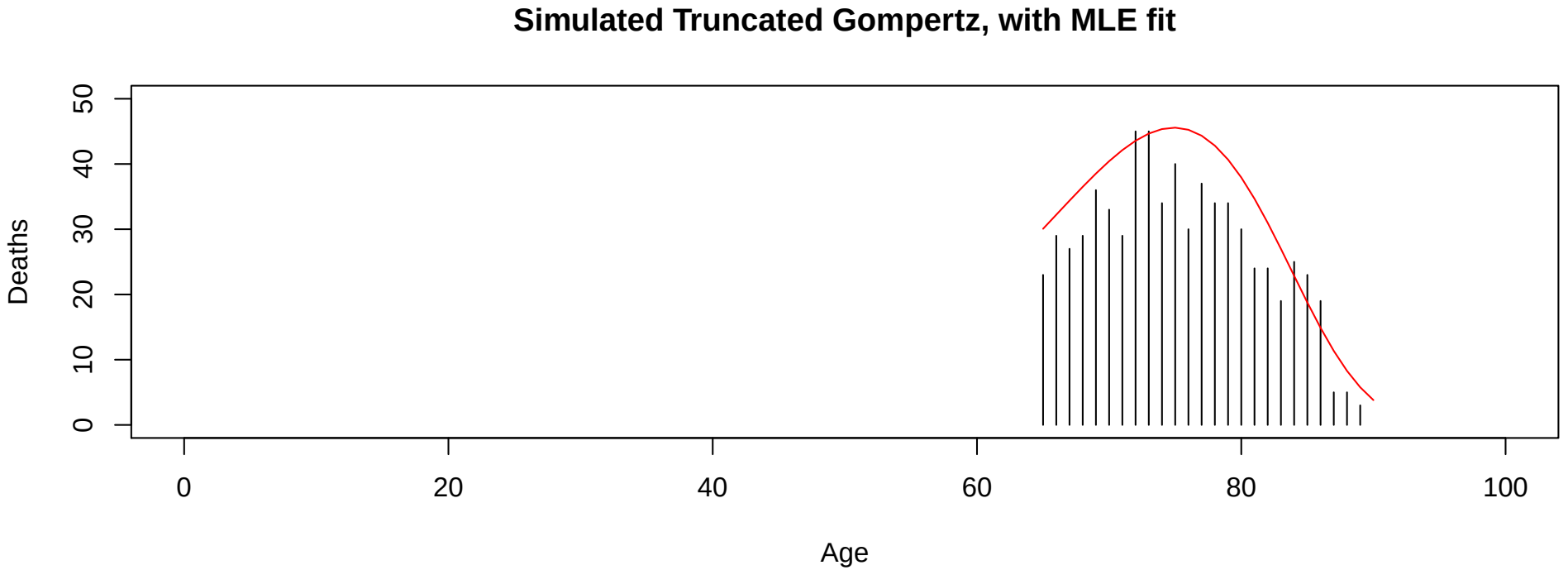
```
l <- 65
h <- 90
x.trunc <- x[x > l & x < h]
Dx <- table(floor(x.trunc))
fit <- counts.trunc.gomp.est(Dx= Dx, x.left = l, x.right = h,
                             b.start = 1/9, M.start = 80)
(b.hat = exp(fit$par[1]))
```

```
## [1] 0.1083535
```

```
(M.hat = exp(fit$par[2]))
```

```
## [1] 75.42279
```

Plot the fit


Simulated Truncated Gompertz, with MLE fit
Deaths
0
10
20
30
40
50
0
20
40
60
80
100
Age

Other ingredients

- A model
- Here we pick Gompertz, with parameters β and M.
- Some code to implement optimization routine
- Validatation
- Test on simulated Gompertz deaths, to see if we estimate right values
- Test on HMD to see if it works with real data
- Application
- NUMIDENT
- (Weighted Censoc)

Major Assumptions

- Gompertz model is appropriate
- Uniform coverage across ages to preserve the cohort distribution of deaths

Code

- Gompertz functions

```
source("hmd_validation_functions.R")

##
fit <- counts.trunc.gomp.est(Dx= Dx, x.left = 0, x.right = 200
                             b.start = 1/9, M.start = 80)
(b.hat = exp(fit$par[1]))
```

```
## [1] 0.1716812
```

```
(M.hat = exp(fit$par[2]))
```

```
## [1] 78.73153
```

- Optimization

```
## wrapper function that calls optim()
```

Without Truncation

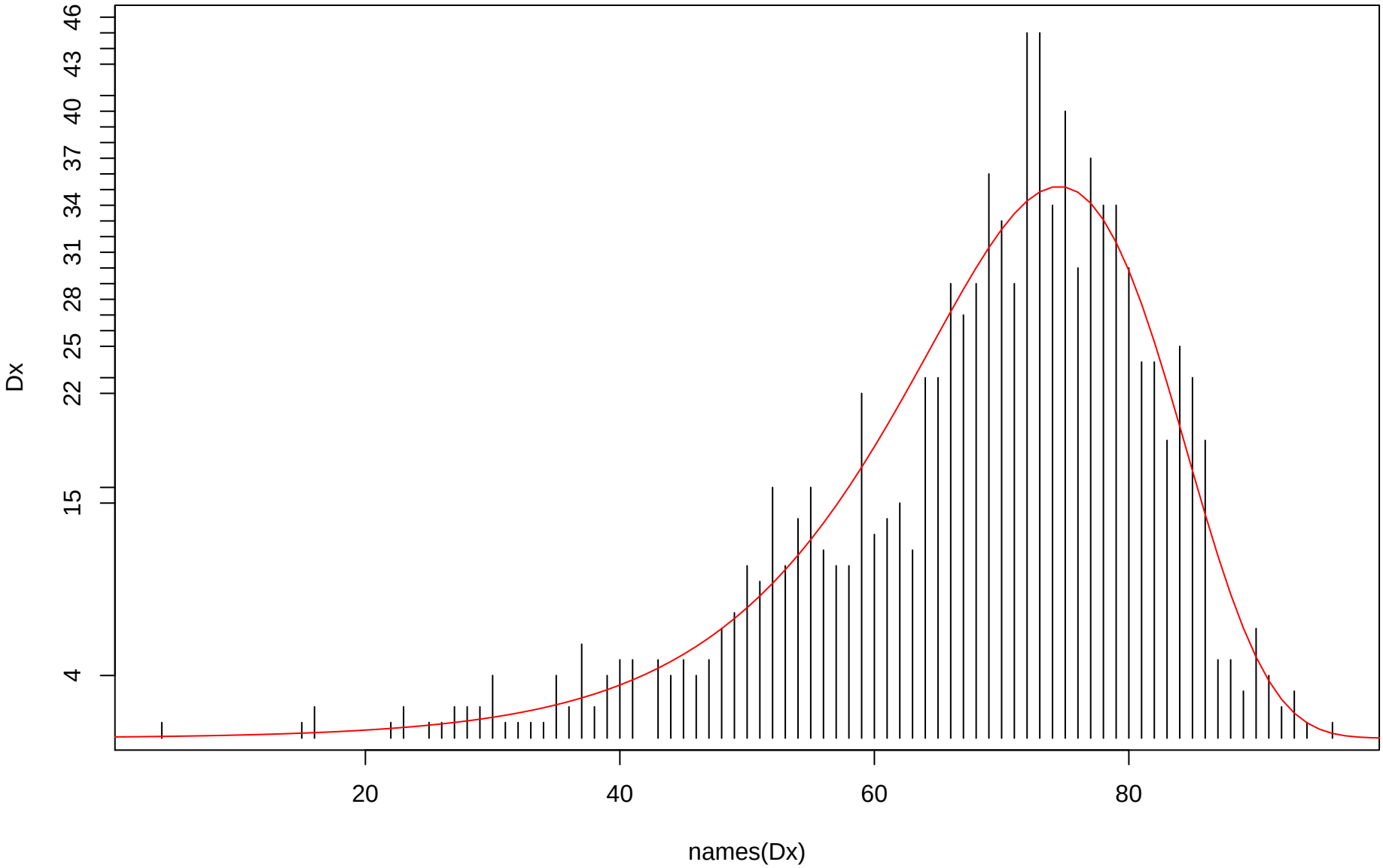
```
N = 1000
set.seed(13)
x <- rgompertz.M(N, b = 1/10, M = 75)
Dx <- table(floor(x))
fit <- counts.trunc.gomp.est(Dx= Dx, x.left = 0, x.right = 200,
                             b.start = 1/9, M.start = 80)
(b.hat = exp(fit$par[1]))
```

```
## [1] 0.09572687
```

```
(M.hat = exp(fit$par[2]))
```

```
## [1] 75.01612
```

Plot the fit



Validating Method with HMD

Our approach

- Use HMD death counts for the year and ages we have NUMIDENT deaths

(1988 to 2005, Ages 65 and over)

- Fit Gompertz and if we match
- Mortality rates
- e(65)

Import and prepare HMD data

```
library(data.table)
## read in and define age and cohort
dt.mx <- fread("~/Documents/hmd/hmd_statistics/death_rates/Mx_1x1/US
dt.mx[ , x := as.numeric(Age)]
```

```
## Warning in eval(jsub, SDenv, parent.frame()): NAs introduced by c
```

```
dt.mx[Age == "110+" , x := 110]
dt.mx[, cohort := Year - x]
dt <- fread("~/Documents/hmd/hmd_statistics/deaths/Deaths_1x1/USA.De
dt[ , x := as.numeric(Age)]
```

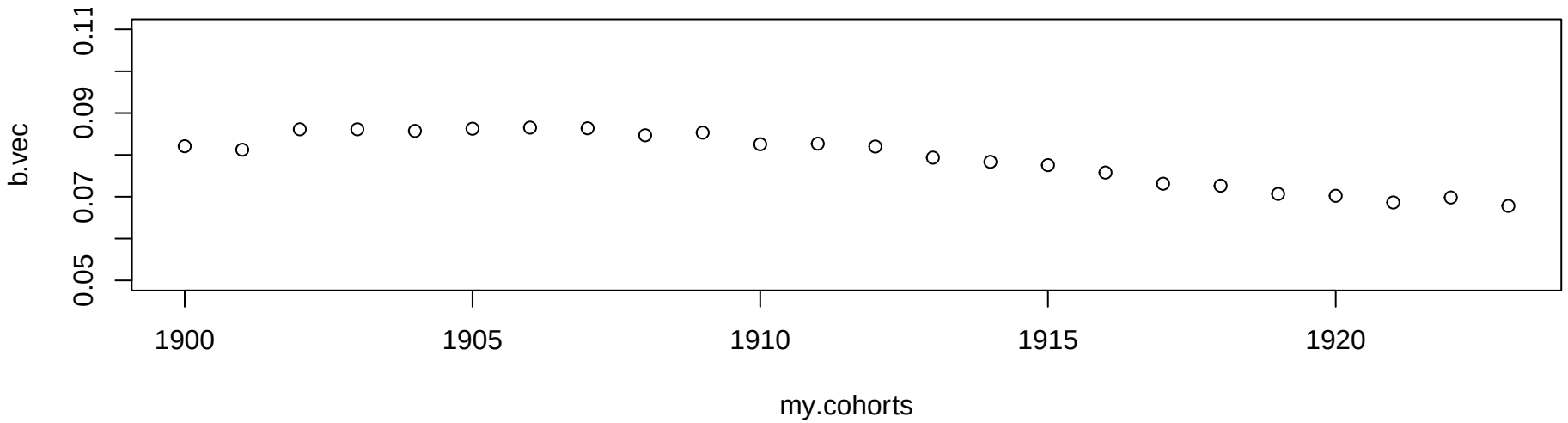
```
## Warning in eval(jsub, SDenv, parent.frame()): NAs introduced by c
```

```
dt[Age == "110+" , x := 110]
dt[, cohort := Year - x]
##
## make array, age x cohort x sex.
dt.long <- melt(dt, measure.vars = c("Male", "Female"), variable.na
```

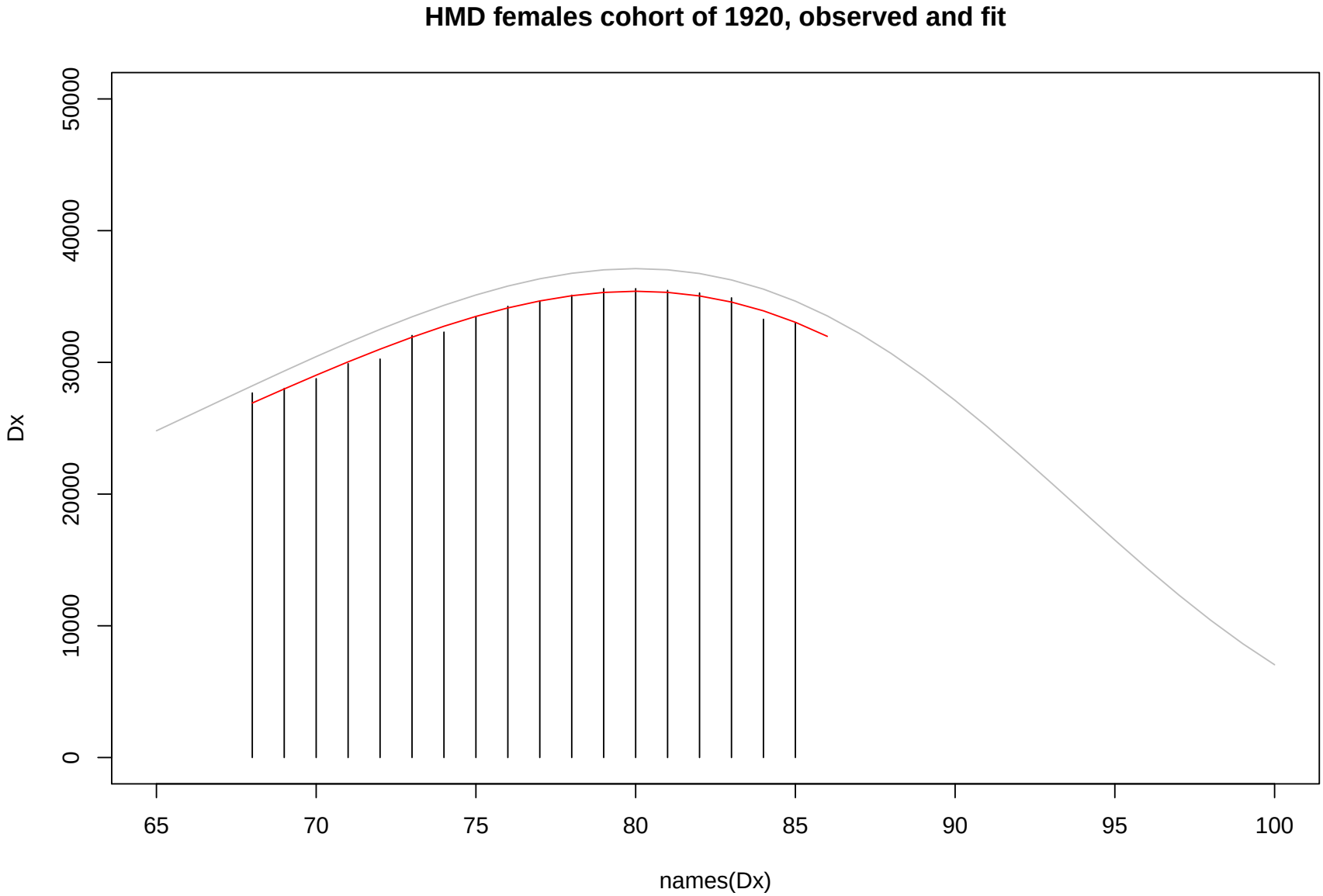
Fit HMD

```
## [1] 1900
## [1] 1901
## [1] 1902
## [1] 1903
## [1] 1904
## [1] 1905
## [1] 1906
## [1] 1907
## [1] 1908
## [1] 1909
## [1] 1910
## [1] 1911
## [1] 1912
## [1] 1913
## [1] 1914
## [1] 1915
## [1] 1916
```

Plot parameter values

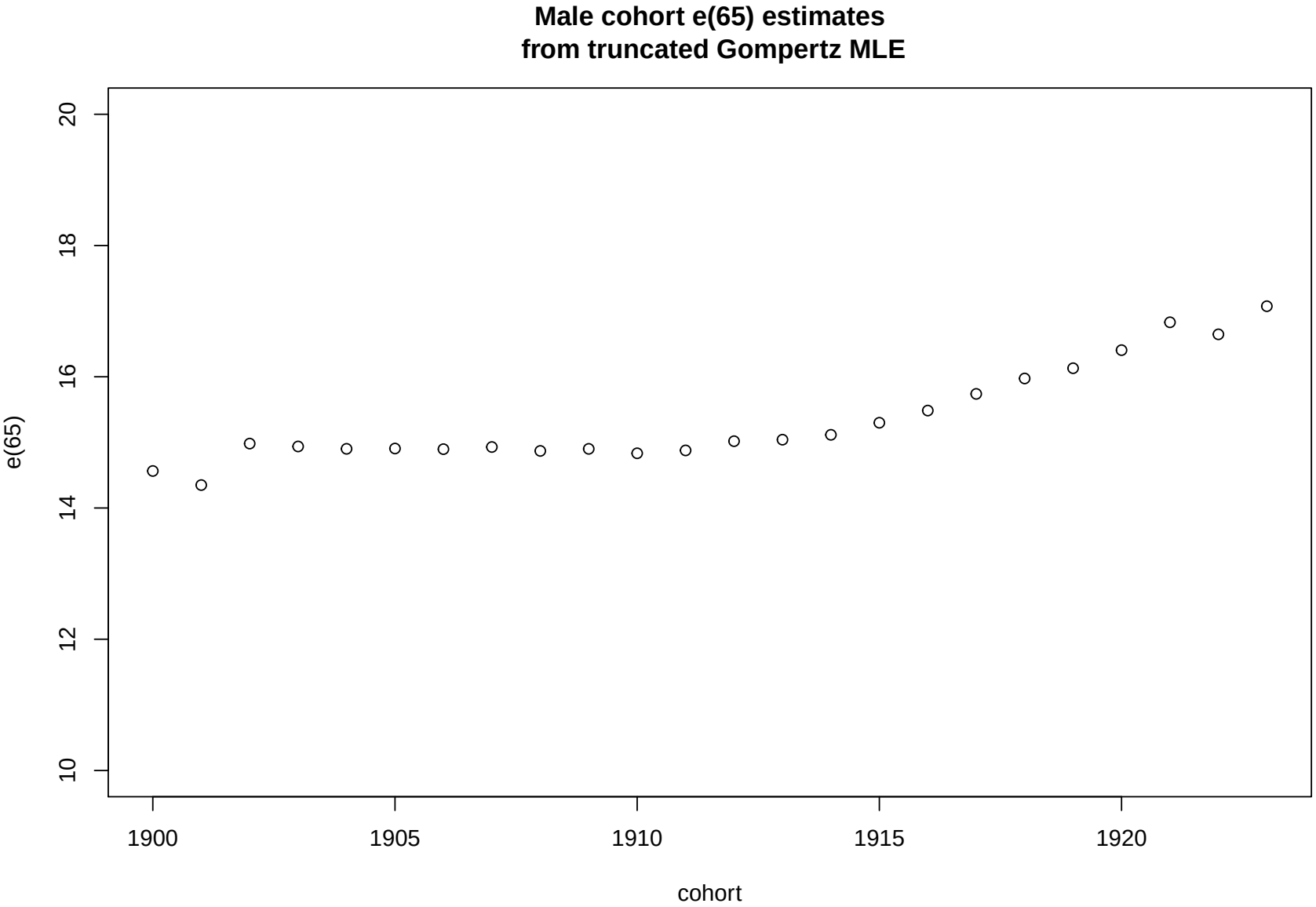


Look at cohort of 1920, pretending we observe only ages 68 to 85



Very high M is not crazy. Period life table for 2016 has mode at age 88

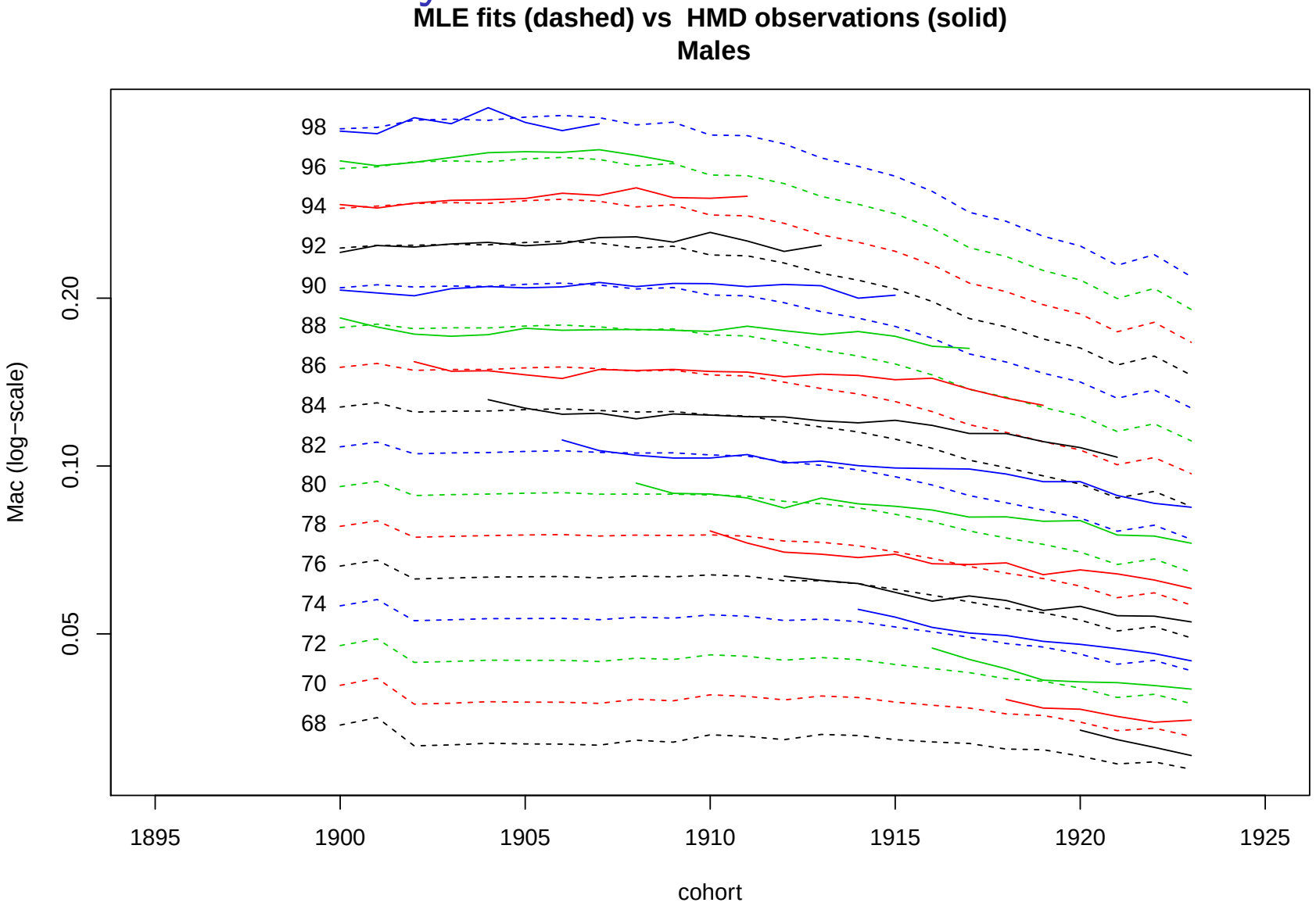
e(65)



For reference, period $e(65, 2016) = 18.36$.

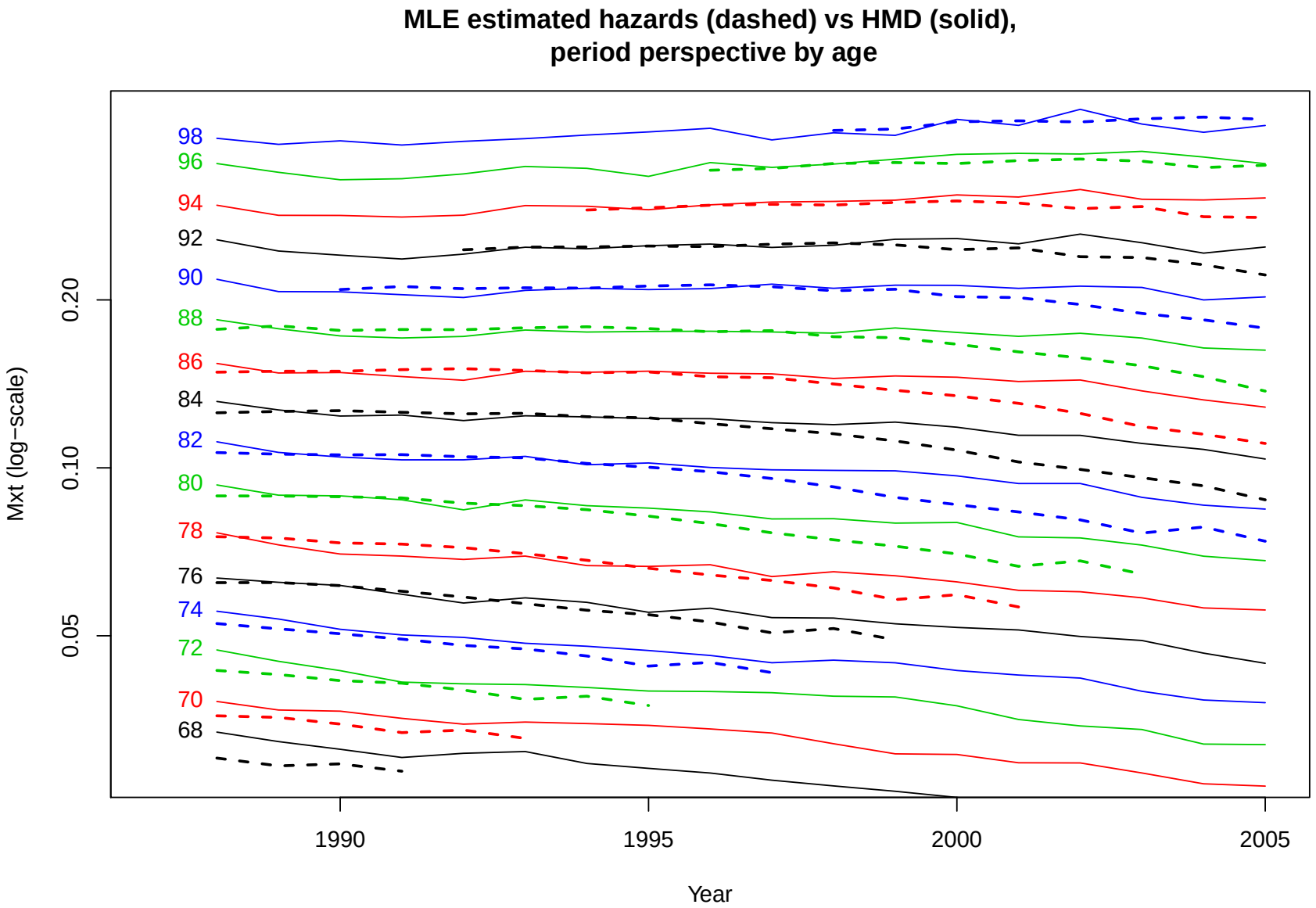
Mortality rates

Selected Mortality Rates



- Overall fit is remarkably good.
- But shouldn't rely on for unobserved ages (see old age decline in upper

Period perspective



Trying out the method with NUMIDENT