D-spline estimation for partial or age-grouped data

Carl Schmertmann

5 Aug 2021

Purpose

In this document I illustrate how to adapt the D-spline estimation method (from Schmertmann 2021,*D-splines: Estimating rate schedules using high-dimensional splines with empirical demographic penalties*, http://www.demographic-research.org/Volumes/Vol44/45/ DOI: 10.4054/DemRes.2021.44.45) for cases with partial and/or age-grouped death and exposure data.

I illustrate with data from Florida counties 2018-2019, downloaded from CDC Wonder (https://wonder.cdc.gov) in June 2021.

Organization

- The first part of this exposition is quite technical and detailed. The published paper assumes that age groups are one year wide, and that we have death and exposure data for all ages. Here I generalize. I derive analytical, closed-form expressions for the gradient vector and Hessian matrix of the D-spline penalized likelihood *when data is for age groups rather than single-year ages, and may not be available for all ages.* These are essential building blocks for finding the set of spline coefficients that best fit the data.
- The second part of this document contains R code that defines a generalized fitting function that can be used or partial or age-grouped data.
- The third part presents examples of fits for small-area data from Florida counties.

1. Analytical Expressions for D-spline gradient and Hessian with Grouped Data

Notation As in Schmertmann (2021)

- $\mathbf{B}$ is the $A \times K$ matrix of B-spline constants
- $\theta \in \mathbb{R}^K$ is the vector of spline coefficients
- $s = \mathbf{B}\theta$ is the spline function that represents age-specific log mortality rates for single year ages $0 \ldots A-1$
- D-spline residual vectors $\varepsilon(\theta) = \mathbf{AB}\theta - c \in \mathbb{R}^R$ are near zero for "good" spline schedules that conform to HMD patterns

Unlike Schmertmann (2021), suppose that

- deaths and exposure are available for G age groups, which may or may not include all ages $0 \ldots A-1$
- The relationship between the vector of single-year mortality rates $(\mu_0 \ldots \mu_{A-1})'$ and age-group rates $(M_1 \ldots M_G)'$ is

  $$M = \mathbf{W}\mu,$$

  where $\mathbf{W}$ is a $G \times A$ matrix of weights
- Given exposures by age group $N = (N_1 \ldots N_G)'$, observed deaths $D = (D_1 \ldots D_G)'$ have independent Poisson distributions

  $$D_g \sim Pois(N_g \, M_g)$$

- The penalized log-likelihood combines the Poisson likelihood summed over *age groups* with a D-spline penalty that rewards “good” *single-year* schedules $s = \mathbf{B}\theta$

$$Q(\theta) = \sum_{g=1}^{G} \left[D_g \ln M_g(\theta) - N_g M_g(\theta) \right] - \frac{1}{2} \varepsilon'(\theta) \mathbf{V}^{-1} \varepsilon(\theta) \tag{1}$$

Newton-Raphson The D-spline estimator selects θ^* to maximize the penalized log likelihood. In practice we use Newton-Raphson iteration. For any current value $\theta_t \in \mathbb{R}^K$, we calculate the current $K \times 1$ gradient vector

$$g_t = g(\theta_t) = \left(\frac{\partial Q}{\partial \theta_1} \cdots \frac{\partial Q}{\partial \theta_K} \right)'$$

and $K \times K$ Hessian matrix

$$H_t = H(\theta_t) \;=\; \begin{bmatrix} \frac{\partial^2 Q}{\partial \theta_1 \partial \theta_1} & \cdots & \frac{\partial^2 Q}{\partial \theta_1 \partial \theta_K} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 Q}{\partial \theta_K \partial \theta_1} & \cdots & \frac{\partial^2 Q}{\partial \theta_K \partial \theta_K} \end{bmatrix}$$

and find the next approximation to the optimum θ^* by solving for θ_{t+1} in the linear system

$$H_t \; \theta_{t+1} = H_t \; \theta_t - g_t$$

We then recalculate g and H at the new θ_{t+1}, and repeat until convergence. See the original paper for more details.

The key point here is that the gradient $g(\theta)$ and Hessian $H(\theta)$ are much more complicated when the input data comes from age groups rather than single-year ages.

D-spline gradient with Age Groups Begin with the vector of rates (*not* log rates):

$$\mu' = \left[\begin{array}{ccc} \mu_1 & \cdots & \mu_A \end{array} \right] = \left[\begin{array}{ccc} e^{b_1'\theta} & \cdots & e^{b_A'\theta} \end{array} \right]$$

and note that the derivatives of these rates with respect to spline coefficients θ are

$$\frac{\partial \mu'}{\partial \theta} = \left[\begin{array}{ccc} \boldsymbol{b}_1 \mu_1 & \cdots & \boldsymbol{b}_A \mu_A \end{array} \right] = \boldsymbol{B'} \text{diag}(\mu) \qquad (K \times A)$$

Age *group* rates and their derivatives are:

$$M = \left[\begin{array}{ccc} M_1 & \cdots & M_G \end{array} \right] = \boldsymbol{W} \mu$$

$$M' = \mu' \boldsymbol{W'}$$

$$\frac{\partial M'}{\partial \theta} = \boldsymbol{B'} \, \text{diag}(\mu) \, \boldsymbol{W'} \qquad (K \times G)$$

Expected deaths by age group and their derivatives are

$$\hat{D}' = \left[\begin{array}{ccc} \hat{D}_1 & \cdots & \hat{D}_G \end{array} \right] = \left[\begin{array}{ccc} N_1 M_1 & \cdots & N_G M_G \end{array} \right] \; = \; M' \text{diag}(N)$$

and

$$\frac{\partial \hat{D}'}{\partial \theta} = \boldsymbol{B'} \, \text{diag}(\mu) \, \boldsymbol{W'} \text{diag}(N) \qquad (K \times G)$$

If we denote the logs of age *group* rates as

$$\lambda' \;=\; \left[\begin{array}{ccc} \ln M_1 & \cdots & \ln M_G \end{array}\right]$$

then its derivatives are

$$\begin{aligned}
\frac{\partial \lambda'}{\partial \theta} &= \left[\begin{array}{ccc} \frac{1}{M_1}\frac{\partial M_1}{\partial \theta} & \cdots & \frac{1}{M_1}\frac{\partial M_1}{\partial \theta} \end{array}\right] \\
&= \frac{\partial M'}{\partial \theta}\,\mathrm{diag}\left(\frac{1}{M}\right) \\
&= \boldsymbol{B'}\,\mathrm{diag}(\mu)\,\boldsymbol{W'}\,\mathrm{diag}\left(\frac{1}{M}\right)
\end{aligned}$$

Penalty residuals and their derivatives are

$$\begin{aligned}
\varepsilon &= \boldsymbol{AB}\theta - c \\
\frac{\partial \varepsilon'}{\partial \theta} &= \boldsymbol{B'A'} \qquad\qquad (K \times R)
\end{aligned}$$

Using these abbreviations, the penalized Poisson log likelihood is

$$Q \;=\; \lambda' D - \hat{D}'1 - \tfrac{1}{2}\varepsilon'\boldsymbol{V}^{-1}\varepsilon$$

and the gradient is therefore the $K \times 1$ vector

$$g(\theta) \;=\; \frac{\partial Q}{\partial \theta} \;=\; \boldsymbol{B'}\,\mathrm{diag}(\mu)\,\boldsymbol{W'}\,\mathrm{diag}\left(\frac{1}{M}\right)D \;-\; \boldsymbol{B'}\,\mathrm{diag}(\mu)\,\boldsymbol{W'}N \;-\; \boldsymbol{B'A'V}^{-1}\varepsilon \tag{2}$$

where the parts that depend on θ are μ, M, and ε.

D-spline Hessian with Age Groups The Hessian is even more complicated. To derive its form, we'll start with one arbitrary scalar element of $g(\theta)$ and differentiate it by an arbitrary scalar element of θ. Then we will re-assemble the pieces into a general form for the $K \times K$ Hessian.

For example, the third element of the gradient vector is the partial derivative of the penalized likelihood function with respect to θ_3:

$$g_3 = \boldsymbol{b_3}'\,\mathrm{diag}(\mu)\,\boldsymbol{W'}\,\mathrm{diag}\left(\frac{1}{M}\right)D \;-\; \boldsymbol{b_3}'\,\mathrm{diag}(\mu)\,\boldsymbol{W'}N \;-\; \boldsymbol{b_3}'\boldsymbol{A'V}^{-1}\varepsilon$$

where $\mathbf{b}_3 \in \mathbb{R}^A$ is the third column of the B-spline basis matrix $\mathbf{B}$.

The partial derivative of g_3 with respect to, say, θ_6 is

$$\begin{aligned}
\frac{\partial g_3}{\partial \theta_6} = \boldsymbol{b_3}'\frac{\partial}{\partial \theta_6}\left[\mathrm{diag}(\mu)\right]\,\boldsymbol{W'}\,\mathrm{diag}\left(\frac{1}{M}\right)D & \qquad (3)\\
+\; \boldsymbol{b_3}'\,\mathrm{diag}(\mu)\,\boldsymbol{W'}\,\frac{\partial}{\partial \theta_6}\left[\mathrm{diag}\left(\frac{1}{M}\right)\right]D & \qquad (4)\\
-\; \boldsymbol{b_3}'\,\frac{\partial}{\partial \theta_6}\left[\mathrm{diag}(\mu)\right]\,\boldsymbol{W'}N & \qquad (5)\\
-\; \boldsymbol{b_3}'\boldsymbol{A'V}^{-1}\boldsymbol{A}\boldsymbol{b}_6 & \qquad (6)
\end{aligned}$$

There are **two** different diagonal matrices in the equation above that vary with θ. Everything else is a known constant. Consider the two diagonal matrices one at time. The **first** is

$$\begin{align}
\frac{\partial}{\partial\theta_6}\left[\operatorname{diag}(\mu)\right] &= \operatorname{diag}\left(\frac{\partial\mu_1}{\partial\theta_6}\cdots\frac{\partial\mu_A}{\partial\theta_6}\right) \tag{7}\\
&= \operatorname{diag}\left(\boldsymbol{b}_6'\frac{\partial\mu'}{\partial\theta}\right) \tag{8}\\
&= \operatorname{diag}\left(\boldsymbol{b}_6'\operatorname{diag}(\mu)\right) \tag{9}\\
&= \operatorname{diag}\left(\begin{array}{ccc} b_{16}\mu_1 & \cdots & b_{A6}\mu_A \end{array}\right) \tag{10}\\
&= \operatorname{diag}(\boldsymbol{b}_6)\operatorname{diag}(\mu) \qquad\qquad (A\times A) \tag{11}
\end{align}$$

The **second** diagonal matrix that varies with θ is

$$\begin{align}
\frac{\partial}{\partial\theta_6}\left[\operatorname{diag}\left(\frac{1}{M}\right)\right] &= \operatorname{diag}\left(\frac{\partial}{\partial\theta_6}\left[\tfrac{1}{M_1}\right]\cdots\frac{\partial}{\partial\theta_6}\left[\tfrac{1}{M_A}\right]\right) \tag{12}\\
&= -\operatorname{diag}\left(M_1^{-2}\frac{\partial M_1}{\partial\theta_6}\cdots M_G^{-2}\frac{\partial M_G}{\partial\theta_6}\right) \tag{13}\\
&= -\operatorname{diag}\left(\boldsymbol{e}_6'\frac{\partial M'}{\partial\theta}\right)\operatorname{diag}(M^{-2}) \tag{14}\\
&= -\operatorname{diag}\left(\boldsymbol{b_6}'\operatorname{diag}(\mu)\,\boldsymbol{W}'\right)\operatorname{diag}(M^{-2}) \tag{15}\\
&= -\operatorname{diag}\left(\boldsymbol{W}\operatorname{diag}(\mu)\,\boldsymbol{b_6}\right)\operatorname{diag}(M^{-2}) \qquad\qquad (G\times G) \tag{16}
\end{align}$$

Replacing the two matrices in the $\frac{\partial g_3}{\partial\theta_6}$ formula with these new expressions that depend on $\boldsymbol{b}_6$, we get

$$\begin{align}
\frac{\partial g_3}{\partial\theta_6} = \ & \boldsymbol{b_3}'\operatorname{diag}(\boldsymbol{b}_6)\operatorname{diag}(\mu)\,\boldsymbol{W}'\operatorname{diag}\left(\frac{1}{M}\right)D \tag{17}\\
& -\ \boldsymbol{b_3}'\operatorname{diag}(\mu)\,\boldsymbol{W}'diag\left[\boldsymbol{W}\operatorname{diag}(\mu)\,\boldsymbol{b_6}\right]\operatorname{diag}(M^{-2})\,D \tag{18}\\
& -\ \boldsymbol{b_3}'\operatorname{diag}(\boldsymbol{b_6})\operatorname{diag}(\mu)\,\boldsymbol{W}'N \tag{19}\\
& -\ \boldsymbol{b_3}'\boldsymbol{A}'\boldsymbol{V}^{-1}\boldsymbol{A}\boldsymbol{b}_6 \tag{20}
\end{align}$$

or more compactly

$$\frac{\partial g_3}{\partial\theta_6} = \boldsymbol{b}_3'\,\boldsymbol{z}_6$$

where z_6 is a complicated $A\times 1$ vector that depends on the 6th column of $\boldsymbol{B}$ as:

$$\begin{aligned}
\boldsymbol{z}_6 = \ & \operatorname{diag}(\boldsymbol{b}_6)\operatorname{diag}(\mu)\,\boldsymbol{W}'\operatorname{diag}\left(M^{-1}\right)\left(D-\hat{D}\right)\\
& -\operatorname{diag}(\mu)\,\boldsymbol{W}'diag\ \left(\boldsymbol{W}\operatorname{diag}(\mu)\,\boldsymbol{b_6}\right)\operatorname{diag}(M^{-2})\,D\\
& -\boldsymbol{A}'\boldsymbol{V}^{-1}\boldsymbol{A}\boldsymbol{b}_6
\end{aligned}$$

Generalizing from this (3,6) element, we get the full (and, admittedly, quite complicated) analytical form of the Hessian matrix

$$H(\theta) = \left[\begin{array}{cccc}
\boldsymbol{b}_1'\boldsymbol{z}_1 & \boldsymbol{b}_1'\boldsymbol{z}_2 & \cdots & \boldsymbol{b}_1'\boldsymbol{z}_K\\
\boldsymbol{b}_2'\boldsymbol{z}_1 & \boldsymbol{b}_2'\boldsymbol{z}_2 & \cdots & \boldsymbol{b}_2'\boldsymbol{z}_K\\
\vdots & \vdots & \ddots & \vdots\\
\boldsymbol{b}_K'\boldsymbol{z}_1 & \boldsymbol{b}_K'\boldsymbol{z}_2 & \cdots & \boldsymbol{b}_K'\boldsymbol{z}_K
\end{array}\right] = \boldsymbol{B}'\left[\begin{array}{ccc} z_1 & \cdots & z_k \end{array}\right] = \boldsymbol{B}'\boldsymbol{Z} \tag{21}$$

which is symmetric.

In the special case of single-year age groups, as in the Appendix of Schmertmann (2021), $\boldsymbol{W} = \boldsymbol{I}$ and $\boldsymbol{M} = \boldsymbol{\mu}$, so that z_k simplifies to

$$z_k \;=\; -\text{diag}(b_k)\,\hat{D} \;-\; \boldsymbol{A}'\boldsymbol{V}^{-1}\boldsymbol{A}b_k$$

and the (i,j) element of the Hessian matrix is

$$H_{ij}(\theta) \;=\; -\left[b_i' \diamond b_j'\right]\;\hat{D} - b_i'\,\boldsymbol{A}'\boldsymbol{V}^{-1}\boldsymbol{A}b_j \qquad\qquad i,j \in \{1...K\}$$

where $\diamond$ indicates element-by-element multiplication. Over all (i,j) this is

$$H(\theta) \;=\; -\boldsymbol{B}'\text{diag}(\hat{D})\boldsymbol{B} \;-\; \boldsymbol{B}'\boldsymbol{A}'\boldsymbol{V}^{-1}\boldsymbol{A}\boldsymbol{B}$$

2. An R function to maximize the penalized likelihood, given age-group deaths and exposure

The following function implements Newton-Raphson search using the gradient and Hessian derived above. The key difference between this function and earlier vintages are the 3rd and 4th arguments, which define the lower and upper bounds for age *groups*, and the new generalized formulas for the gradient and the Hessian.

Notice that the default settings are for 100 single-year age groups, with lower bounds of 0,1,. . . ,99 and upper bounds of 1,2,. . . ,100 – i.e, [0,1), [1,2), . . . [99,100).

```r
library('splines')

Dspline_fit = function(N, D,
                       age_group_lower_bounds = 0:99,
                       age_group_upper_bounds = 1:100,
                       Amatrix, cvector, SIGMA.INV,
                       knots          = seq(from=3,to=96,by=3),
                       max_iter       = 20,
                       theta_tol      = .00005,
                       details        = FALSE) {

  require(splines)

  # cubic spline basis
  B    = bs(0:99, knots=knots, degree=3, intercept=TRUE)

  # number of spline parameters
  K = ncol(B)

  ## number and width of age groups
  age_group_labels = paste0('[',age_group_lower_bounds,',',age_group_upper_bounds,')')

  G     = length(age_group_lower_bounds)
  nages = age_group_upper_bounds - age_group_lower_bounds

  ## weighting matrix for mortality rates (assumes uniform
  ## distribution of single-year ages within groups)
  W = outer(seq(G), 0:99, function(g,x){ 1*(x >= age_group_lower_bounds[g])*
      (x <  age_group_upper_bounds[g])}) %>%
    prop.table(margin=1)

  dimnames(W) = list(age_group_labels , 0:99)
  ## penalized log lik function
  pen_log_lik = function(theta) {
```

```
  lambda.hat = as.numeric( B %*% theta)
  eps        = Amatrix %*% lambda.hat - cvector
  penalty    = 1/2 * t(eps) %*% SIGMA.INV %*% eps

  M    = W %*% exp(B %*% theta)   # mortality rates by group
  logL = sum(D * log(M) - N * M)
  return(logL  - penalty)
}

## expected deaths function
Dhat = function(theta) {
  M    = W %*% exp(B %*% theta)   # mortality rates by group
  return(  as.numeric( N * M ))
}

## gradient function (1st deriv of pen_log_lik wrt theta)
gradient = function(theta) {
  lambda.hat = as.numeric( B %*% theta)
  eps        = Amatrix %*% lambda.hat - cvector

  mx    = exp(lambda.hat)
  Mg    = as.numeric(W %*% mx)
  X     = W %*% diag(mx) %*% B
  return( t(X) %*% diag(1/Mg) %*% (D-Dhat(theta)) -
            t(B) %*% t(Amatrix) %*% SIGMA.INV %*% eps )
}


hessian = function(theta) {

  mu     = as.vector( exp(B %*% theta))
  M      = as.vector( W %*% mu)

  Dhat = N * M

  construct_zvec = function(k) {
    part1 = diag(B[,k]) %*% diag(mu) %*% t(W) %*% diag(1/M) %*% (D - Dhat)
    part2 = diag(mu) %*% t(W) %*% diag(as.vector(W %*% diag(mu) %*% B[,k])) %*% diag(1/(M^2)) %*% D
    part3 = t(Amatrix) %*% SIGMA.INV %*% Amatrix %*% B[,k]

    return(part1 - part2 - part3)
  }

  Z = sapply(1:K, construct_zvec)

  H = t(B) %*% Z

  # slight clean-up to guarantee total symmetry
  return( (H + t(H))/2 )
} # hessian

#-------------------------------------------------
# iteration function:
 # next theta vector as a function of current theta
 #-----------------------------------------------

next_theta = function(theta) {
  H = hessian(theta)
  return( as.vector( solve( H, H %*% theta - gradient(theta) )))
}

 ## main iteration:
 th = rep( log(sum(D)/sum(N)), K)  #initialize at overall avg log rate
 niter = 0

 repeat {

   niter      = niter + 1
   last_param = th
   th         = next_theta( th )  # update
   change     = th - last_param

   converge = all( abs(change) < theta_tol)
   overrun  = (niter == max_iter)

   if (converge | overrun) { break }

 } # repeat

 if (details | !converge | overrun) {
   if (!converge) print('did not converge')
   if (overrun) print('exceeded maximum number of iterations')

   dhat = Dhat(th)
   H    = hessian(th)
   g    = gradient(th)

   BWB   = t(B) %*% t(W) %*% diag(dhat) %*% W %*% B
   BAVAB = t(B) %*% t(Amatrix) %*% SIGMA.INV %*% Amatrix %*% B
   df    = sum( diag( solve(BWB+BAVAB) %*% BWB)) # trace of d[Dhat]/d[D'] matrix

   lambda.hat = B %*% th

   dev = 2 * sum( (D>0) * D * log(D/dhat), na.rm=TRUE)

   return( list( N                 = N,
                 D                 = D,

                 age_group_lower_bounds = age_group_lower_bounds,
                 age_group_upper_bounds = age_group_upper_bounds,

                 B                 = B,
                 theta             = as.vector(th),
                 lambda.hat        = as.vector(lambda.hat),
                 gradient          = as.vector(g),
                 dev               = dev,
                  df               = df,
                  bic              = dev + df * log(length(D)),
                  aic              = dev + 2*df,
                  fitted.values    = as.vector(dhat),
                  obs.values       = D,
                  obs.expos        = N,
                  hessian          = H,
                  covar            = solve(-H),
                  pen_log_lik      = pen_log_lik(th),
                  niter            = niter,
                  converge         = converge,
                  maxiter          = overrun))
  } else return( th )

} # Dspline_fit
```

3. Examples

Here I use three examples with CDC age-group data from Florida counties for 2018-2019. Many counties are small enough that CDC intentionally supresses death counts for some age groups.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# load the Dspline constants, structured as
# List of 3
#  $ Female:List of 3
#   ..$ D-1 :List of 3
#   ..$ D-2 :List of 3
#   ..$ D-LC:List of 5
#  $ Male  :List of 3
#   ..$ D-1 :List of 3
#   ..$ D-2 :List of 3
#   ..$ D-LC:List of 5
#  $ Total :List of 3
#   ..$ D-1 :List of 3
#   ..$ D-2 :List of 3
#   ..$ D-LC:List of 5
#
# each of the three main components has subcomponents
# A, c, SIGMA.INV for the  D-1 and D-2 models
# and A, c, SIGMA.INV, LCa, LCb for the D-LC model

load(url('http://bonecave.schmert.net/general_Dspline_constants.RData'))

# read FL county data and display a small chunk
```

```
FL = read.delim(file=url('http://bonecave.schmert.net/Underlying Cause of Death, 2018-2019, Single Race
               header=TRUE, sep="\t",
               na.strings = c('Not Applicable','Unreliable','Suppressed')) %>%
  mutate(County = str_replace(County,' County, FL', ''))


# there are 21 age groups in the CDC data
age_group_info = tibble(
  Five.Year.Age.Groups =
    c("< 1 year", "1-4 years", "5-9 years", "10-14 years", "15-19 years",
    "20-24 years", "25-29 years", "30-34 years", "35-39 years", "40-44 years",
    "45-49 years", "50-54 years", "55-59 years", "60-64 years ",
    "65-69 years", "70-74 years", "75-79 years", "80-84 years", "85-89 years",
    "90-94 years", "95-99 years"),
  L = c(0,1,seq(from=5,to=95,by=5)),
  H = c(1, seq(from=5, to=100, by=5))
)

FL = inner_join(FL, age_group_info, by='Five.Year.Age.Groups') %>%
      select(County, Gender.Code, Five.Year.Age.Groups, Deaths, Population, L, H)
```

A quick look at the first several observations shows how small death counts are surpressed: the number of deaths is not reported for anyone age 1-4 or 5-9 in Alachua County because those numbers were too small.

```
head(FL, 16)
```

```
##     County Gender.Code Five.Year.Age.Groups Deaths Population  L   H
## 1  Alachua           F             < 1 year     26       2717  0   1
## 2  Alachua           M             < 1 year     31       2863  0   1
## 3  Alachua           F            1-4 years     NA      11135  1   5
## 4  Alachua           M            1-4 years     NA      11216  1   5
## 5  Alachua           F            5-9 years     NA      13672  5  10
## 6  Alachua           M            5-9 years     NA      14109  5  10
## 7  Alachua           F          10-14 years     NA      12902 10  15
## 8  Alachua           M          10-14 years     NA      13356 10  15
## 9  Alachua           F          15-19 years     NA      22716 15  20
## 10 Alachua           M          15-19 years     12      20229 15  20
## 11 Alachua           F          20-24 years     NA      42443 20  25
## 12 Alachua           M          20-24 years     20      40724 20  25
## 13 Alachua           F          25-29 years     NA      23598 25  30
## 14 Alachua           M          25-29 years     22      23667 25  30
## 15 Alachua           F          30-34 years     13      18417 30  35
## 16 Alachua           M          30-34 years     31      18311 30  35
```

Fitting for grouped data: Alachua County FL

Now we'll fit D-spline models for Alachua County males (Although I no longer live there, I was born at Alachua General Hospital in 1959!).

```
# get the Alachua male data for the subset of age groups that have both death and population counts

df = FL %>%
      filter(County=='Alachua', Gender.Code=='M',
             is.finite(Deaths), is.finite(Population)) %>%
      mutate(logM = log(Deaths/Population))
```

```
print(df)
```

```
##     County Gender.Code Five.Year.Age.Groups Deaths Population  L  H      logM
## 1  Alachua           M             < 1 year     31       2863  0  1 -4.525638
## 2  Alachua           M          15-19 years     12      20229 15 20 -7.429966
## 3  Alachua           M          20-24 years     20      40724 20 25 -7.618841
## 4  Alachua           M          25-29 years     22      23667 25 30 -6.980794
## 5  Alachua           M          30-34 years     31      18311 30 35 -6.381270
## 6  Alachua           M          35-39 years     38      16281 35 40 -6.060168
## 7  Alachua           M          40-44 years     34      13273 40 45 -5.967127
## 8  Alachua           M          45-49 years     51      13002 45 50 -5.541033
## 9  Alachua           M          50-54 years     78      12403 50 55 -5.068985
## 10 Alachua           M          55-59 years    132      13522 55 60 -4.629271
## 11 Alachua           M         60-64 years     215      13208 60 65 -4.117940
## 12 Alachua           M          65-69 years    235      11912 65 70 -3.925716
## 13 Alachua           M          70-74 years    262       9232 70 75 -3.562086
## 14 Alachua           M          75-79 years    233       6095 75 80 -3.264186
## 15 Alachua           M          80-84 years    225       3377 80 85 -2.708643
```

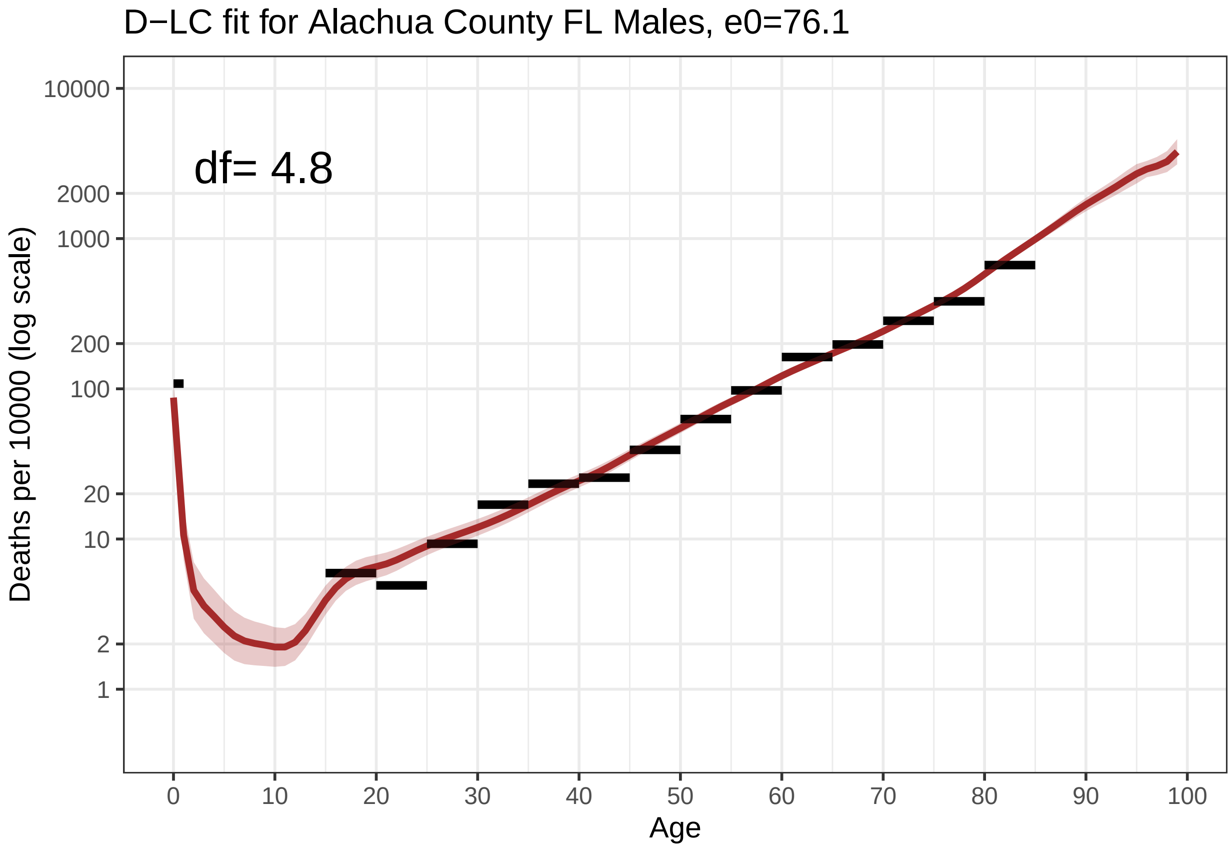
Notice that only 15 of 21 age groups have published data. Also notice that each group has a lower age bound L and and upper bound H.

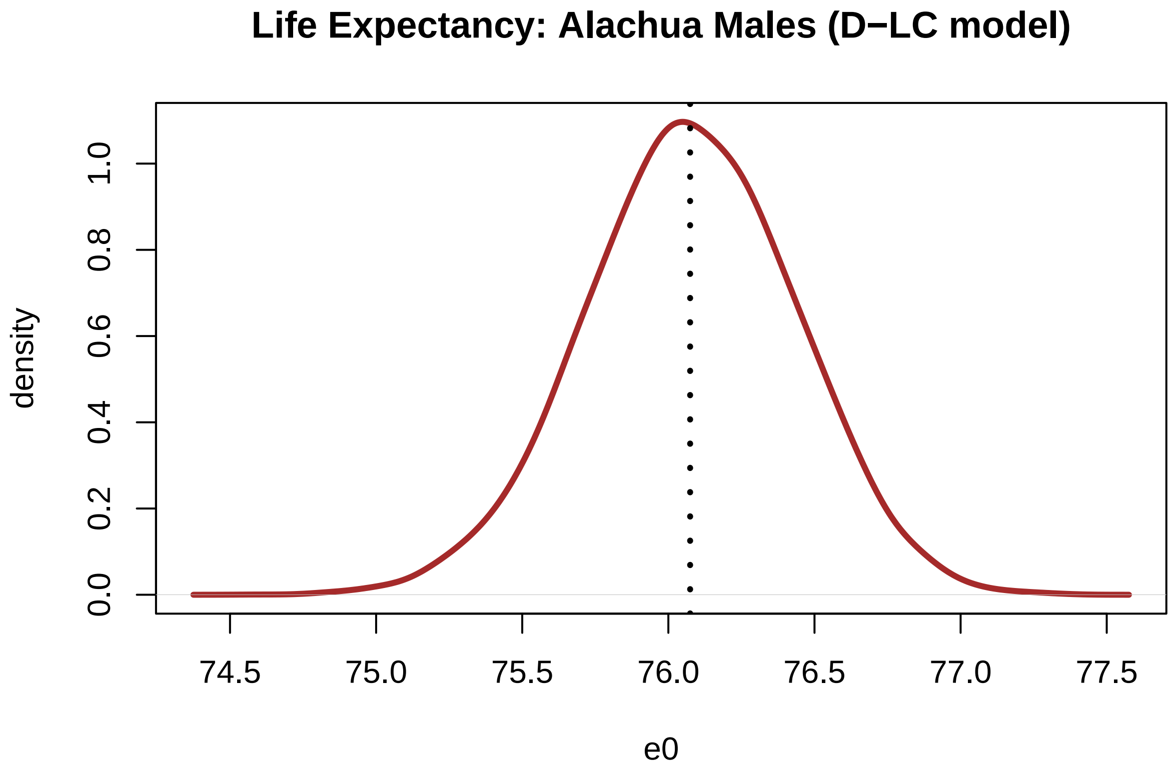
Let's fit a D-LC (Lee-Carter Dspline) model to the available Alachua age group data, and then plot some of the results. We'll do this through a reusable function for which we can change several parameters.

```
make_example = function(this_county, this_gender_code, this_method) {

  this_sex = c('F'='Female', 'M'='Male')[this_gender_code]
  this_hue =   this_hue = c('D-1'='darkgreen',
                          'D-2'='blue',
                          'D-LC'='brown')[this_method]

  df = FL %>%
      filter(County==this_county, Gender.Code==this_gender_code,
             is.finite(Deaths), is.finite(Population)) %>%
      mutate(logM = log(Deaths/Population))

  print(df)

  this_N      = df$Population
  this_D      = df$Deaths

  fit = Dspline_fit(N=df$Population, D=df$Deaths,
            age_group_lower_bounds = df$L,
            age_group_upper_bounds = df$H,
            Amatrix    = Dspline_constants[[this_sex]][[this_method]]$A,
            cvector    = Dspline_constants[[this_sex]][[this_method]]$c,
            SIGMA.INV = Dspline_constants[[this_sex]][[this_method]]$SIGMA.INV,
            max_iter   = 50,
            details=TRUE)

  # first illustrate the fitted log mortality rates (and uncertainty)
    G = ggplot() +
       geom_line(aes(x=0:99, y=fit$lambda.hat),
                 lwd=1.2, color=this_hue) +
       theme_bw() +
       scale_x_continuous(breaks=seq(0,100,10), minor_breaks = seq(0,100,5)) +
       scale_y_continuous(limits=c(-10,0),
                          breaks=log(c(.0001,.0002, .0010,.0020, .0100,.0200,.1000,.2000,1)),
                          minor_breaks = NULL,
                          labels = c('1','2', '10','20', '100','200','1000','2000','10000')) +
       labs(x='Age',y='Deaths per 10000 (log scale)') +
       geom_text(aes(x=2, y=log(.30)),
                 label=paste0('df= ',round(fit$df,1)),
                 hjust=0, size=6)


  G = G +
      geom_segment(data=df, aes(x=L, y=logM, xend=H, yend=logM),
                   size=1.5)


  mx = exp(fit$lambda.hat)
  px = exp(-mx)
  lx = c(1, cumprod(px))
  e0 = sum( tail(lx,-1) + head(lx,-1)) /2 +
        tail(lx,1) * 1/(tail(mx,1))


  se = (fit$B %*% fit$covar %*% t(fit$B)) %>% diag() %>% sqrt()

  G = G + geom_ribbon(aes(x=0:99, ymin=fit$lambda.hat-1.28*se, ymax=fit$lambda.hat+1.28*se),
                  fill=this_hue, alpha=.25) +
       labs(title=paste0(paste0(this_method,' fit for ',this_county,' County FL ',this_sex,'s, e0=', rou

  print(G)


  # next illustrate estimated life expectancy (and uncertainty)
  # simulate 10000 draws of the spline coefficient vector,
  # using a multivariate normal approx
  B         = fit$B
  CH        = t(chol(fit$covar))
  theta.sim = fit$theta + CH %*% matrix(rnorm(10000*ncol(CH)),nrow=ncol(CH))

  lambda.sim = B %*% theta.sim

  # function to convert log mortality rates into e0
  e0 = function(lambda) {
    mx = exp(lambda)
    px = exp(-mx)
    lx = cumprod( c(1,px))

    life.exp = sum((head(lx,-1) + tail(lx,-1))/2) +
      tail(lx,1)/tail(mx,1)
  }

  e = apply(lambda.sim, 2, e0)
```

```
  plot(density(e, adjust=1.5),lwd=3,
       xlab='e0', ylab='density',
       main=paste0('Life Expectancy: ',this_county,' ',this_sex,'s (', this_method, ' model)'),
       col=this_hue)
  abline(v=median(e),lty='dotted',lwd=3)


}

make_example('Alachua', 'M', 'D-LC')
```

```
##     County Gender.Code Five.Year.Age.Groups Deaths Population  L  H      logM
## 1  Alachua           M             < 1 year     31       2863  0  1 -4.525638
## 2  Alachua           M          15-19 years     12      20229 15 20 -7.429966
## 3  Alachua           M          20-24 years     20      40724 20 25 -7.618841
## 4  Alachua           M          25-29 years     22      23667 25 30 -6.980794
## 5  Alachua           M          30-34 years     31      18311 30 35 -6.381270
## 6  Alachua           M          35-39 years     38      16281 35 40 -6.060168
## 7  Alachua           M          40-44 years     34      13273 40 45 -5.967127
## 8  Alachua           M          45-49 years     51      13002 45 50 -5.541033
## 9  Alachua           M          50-54 years     78      12403 50 55 -5.068985
## 10 Alachua           M          55-59 years    132      13522 55 60 -4.629271
## 11 Alachua           M         60-64 years     215      13208 60 65 -4.117940
## 12 Alachua           M          65-69 years    235      11912 65 70 -3.925716
## 13 Alachua           M          70-74 years    262       9232 70 75 -3.562086
## 14 Alachua           M          75-79 years    233       6095 75 80 -3.264186
## 15 Alachua           M          80-84 years    225       3377 80 85 -2.708643
```


D−LC fit for Alachua County FL Males, e0=76.1
df= 4.8
Deaths per 10000 (log scale)
10000
2000
1000
200
100
20
10
2
1
0
10
20
30
40
50
60
70
80
90
100
Age

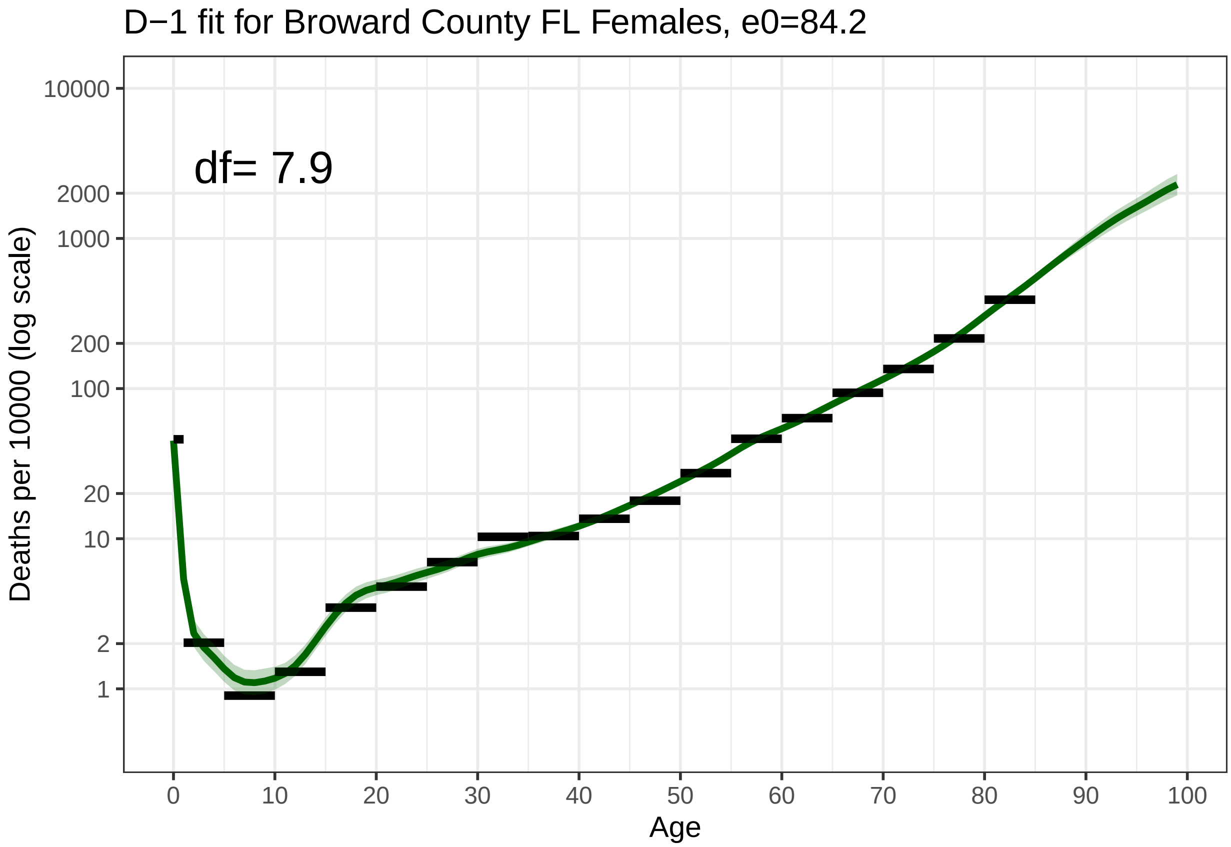


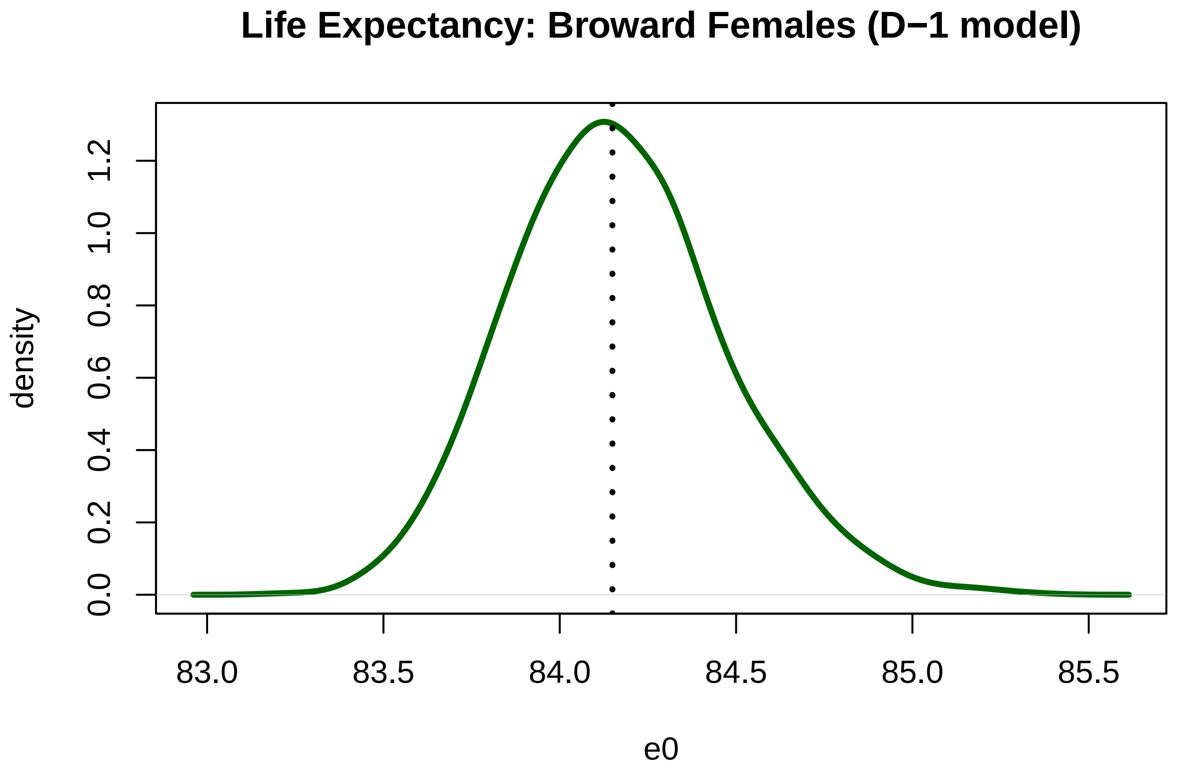
Fitting for grouped data: Broward County FL

Now we'll fit a D-1 model for females in Broward County, which is one of Florida's **most** populous.

```
make_example('Broward','F','D-1')
```

```
##     County Gender.Code Five.Year.Age.Groups Deaths Population  L  H      logM
## 1  Broward           F             < 1 year     99      21552  0  1 -5.383104
## 2  Broward           F            1-4 years     18      88722  1  5 -8.502891
## 3  Broward           F            5-9 years     10     110961  5 10 -9.314349
## 4  Broward           F          10-14 years     15     115489 10 15 -8.948880
## 5  Broward           F          15-19 years     38     109367 15 20 -7.964878
## 6  Broward           F          20-24 years     52     108432 20 25 -7.642635
## 7  Broward           F          25-29 years     91     130345 25 30 -7.267081
## 8  Broward           F          30-34 years    137     133037 30 35 -6.878402
## 9  Broward           F          35-39 years    142     136306 35 40 -6.866831
## 10 Broward           F          40-44 years    178     131086 40 45 -6.601825
## 11 Broward           F          45-49 years    246     137349 45 50 -6.324949
## 12 Broward           F          50-54 years    386     141227 50 55 -5.902286
## 13 Broward           F          55-59 years    661     142724 55 60 -5.374914
## 14 Broward           F         60-64 years     810     127489 60 65 -5.058751
## 15 Broward           F          65-69 years   1011     107875 65 70 -4.670033
## 16 Broward           F          70-74 years   1215      89953 70 75 -4.304543
## 17 Broward           F          75-79 years   1412      65464 75 80 -3.836493
## 18 Broward           F          80-84 years   1830      46820 80 85 -3.241994
```


D−1 fit for Broward County FL Females, e0=84.2
df= 7.9
Deaths per 10000 (log scale)
10000
2000
1000
200
100
20
10
2
1
0
10
20
30
40
50
60
70
80
90
100
Age

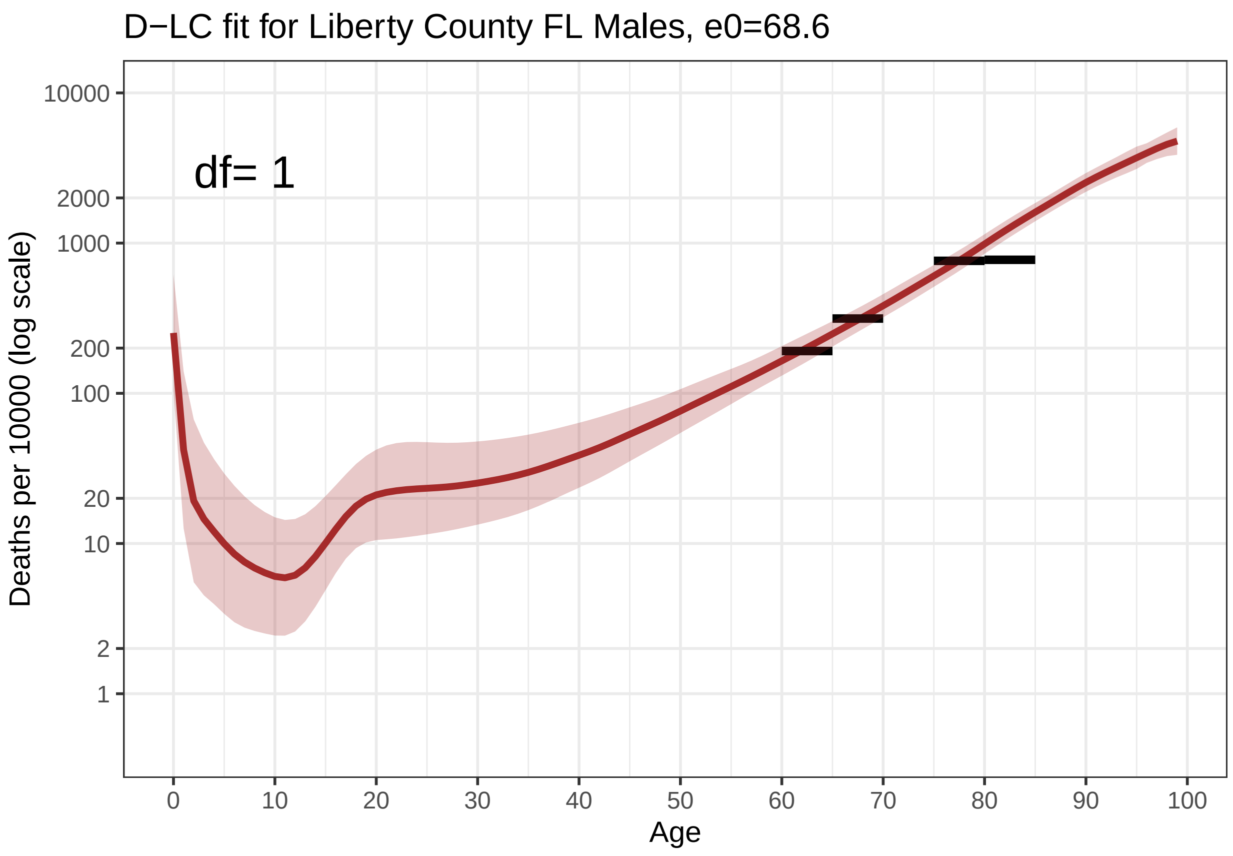


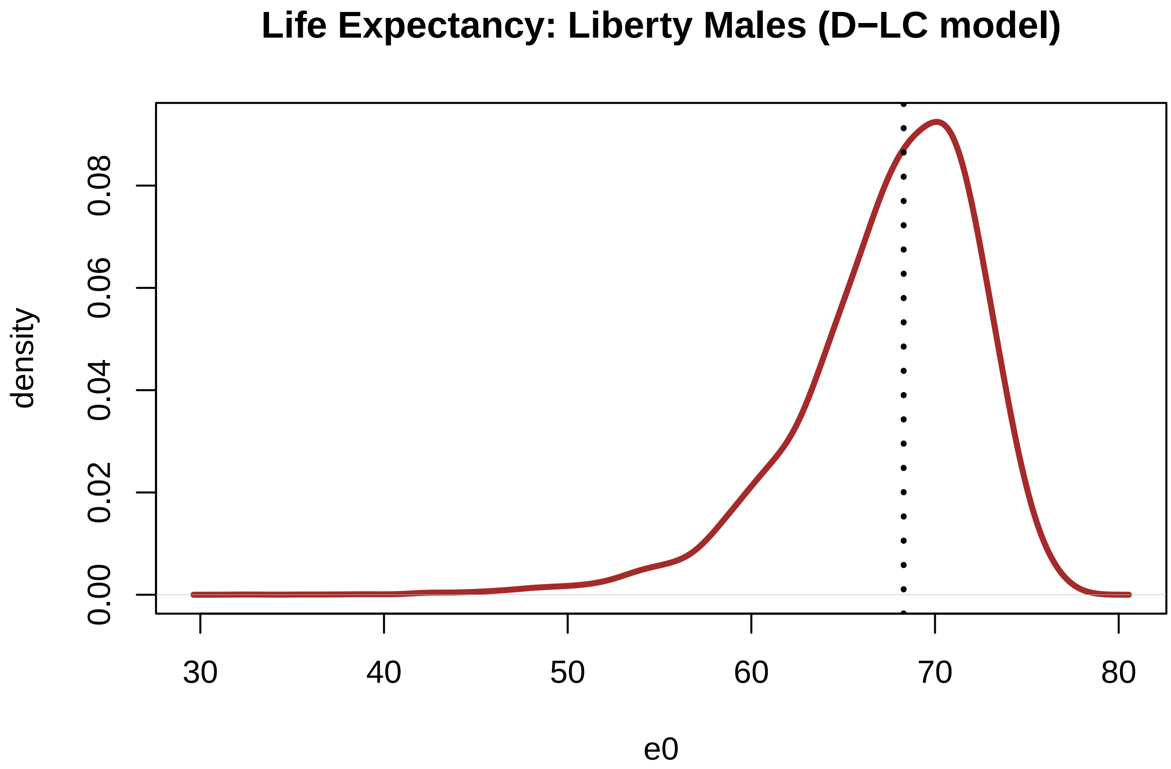
Fitting for grouped data: Liberty FL

Last we'll fit a D-LC model for males in Liberty County, which is one of Florida's **least** populous.

```
make_example('Liberty','M','D-LC')
```

```
##    County Gender.Code Five.Year.Age.Groups Deaths Population  L  H      logM
## 1 Liberty           M          60-64 years     10        523 60 65 -3.956996
## 2 Liberty           M           65-69 years     14        445 65 70 -3.459017
## 3 Liberty           M           75-79 years     16        210 75 80 -2.574519
## 4 Liberty           M           80-84 years     12        155 80 85 -2.558518
```


D−LC fit for Liberty County FL Males, e0=68.6
df= 1
Deaths per 10000 (log scale)
10000
2000
1000
200
100
20
10
2
1
0
10
20
30
40
50
60
70
80
90
100
Age


Life Expectancy: Liberty Males (D−LC model)
density
0.00
0.02
0.04
0.06
0.08
30
40
50
60
70
80
e0