Uncertainty in TOPALS fits

Carl Schmertmann

22 Aug 2019

Main Idea

When we fit a mortality model to (exposure,death) data in a small population, there is considerable uncertainy about the estimated rates. That's true of any regression model, of course. And in all models it's useful to know how much to trust fitted results.

Here I show how to use the detailed information from the **TOPALS_fit** function to approximate confidence intervals for various output(s).

Example Case: Italian Female Mortality 1980

We'll assume that the true mortality schedule for a population is a the HMD single-year schedule for Italian females in 1980 over ages 0..99.

For a TOPALS standard schedule we'll pick something arbitrary: a smoothed version of the HMD schedule for Canadian females in 1959.

The code below fits a TOPALS model for exposure and death data aggregated into age groups $[0,1), [1,5), [5,10), \ldots [95,100)$.

```
library(tidyverse)
library(splines)

# include the fitting function
source('TOPALS_fit.R', echo=TRUE)
```

```
##
## > TOPALS_fit = function(N, D, std, age_group_bounds = 0:100,
## +     knot_positions = c(0, 1, 10, 20, 40, 70), penalty_precision = 2,
## +     max_iter = .... [TRUNCATED]
```

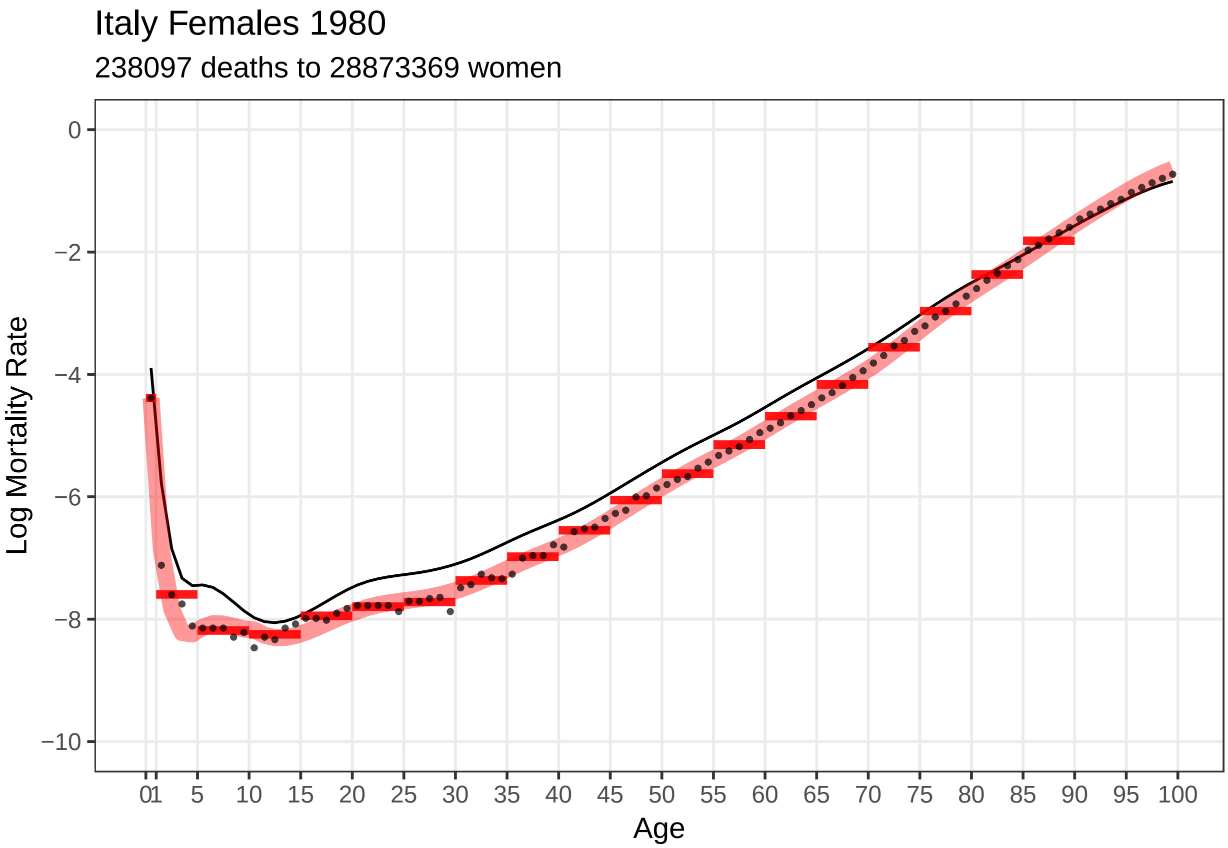
Get Italy data

```
# read the single-year Italian female 1980 HMD data ages 0,1,...110+
ITA = read.csv('ITA-Female-1980.csv')

# read the std schedule (log rates for CAN females 1959, ages 0...99)
```

```r
std = read.csv('female-std.csv')$std

## age-grouping function
agg = function(x,bounds) {
  age = seq(x)-1  # 0,1,2,...
  L = head(bounds,-1)
  U = tail(bounds,-1)
  as.vector( tapply( x, cut(age, breaks=bounds, right=FALSE), sum))
}


bb = c(0,1,seq(5,90,5))  # last group is [85,90)

N = agg(ITA$N, bounds=bb)
D = agg(ITA$D, bounds=bb)

names(N) = names(D) = head(bb,-1)


show_topals = function(fit, emphasize_sd=FALSE, hue='red') {

 df_grouped = data.frame(
                L = head( fit$age_group_bounds, -1),
                U = tail( fit$age_group_bounds, -1),
                N = fit$N,
                D = fit$D
               ) %>%
          mutate(logmx_obs = log(D/N))


 df_single  = data.frame(
                  age=  seq(fit$std) - .50,  # 0.5, 1.5, ...
                  std = fit$std,
                  logmx_true = ITA$logmx[1:100],
                  logmx_fit  = fit$logm
               )

  this_plot =
  ggplot(data = df_single, aes(x=age,y=logmx_true)) +
      labs(x='Age',y='Log Mortality Rate',
           title='Italy Females 1980',
           subtitle = paste(sum(fit$D),'deaths to',round(sum(fit$N)),'women')) +
      scale_x_continuous(breaks=c(0,1,seq(5,100,5)),minor_breaks = NULL) +
      scale_y_continuous(limits=c(-10,0),breaks=seq(-10,0,2),minor_breaks = NULL) +
      theme_bw()


 if (!emphasize_sd) {


 this_plot = this_plot +
      geom_line(aes(x=age,y=std), color='black', lwd=0.5) +
      geom_line(aes(x=age,y=logmx_fit), color=hue, lwd=3, alpha=.40) +
      geom_segment(data=df_grouped,aes(x=L,xend=U,
                                       y=logmx_obs,
                                       yend=logmx_obs),
                   color=hue,lwd=1.5, alpha=.90) +
      geom_point(size=0.60, alpha=.70)


 } else {

    sd = sqrt( diag( fit$B %*% fit$covar %*% t(fit$B)))
    factor = qnorm(.90)
    df_errors = data.frame(age = seq(fit$std) - 0.50,
                           L = fit$logm - factor * sd,
                           H = fit$logm + factor * sd)

    this_plot = this_plot +
                  geom_segment(data=df_errors,
                               aes(x=age, xend=age, y=L, yend=H), lwd=0.5, color=hue,
                               inherit.aes = FALSE)

  }

  print(this_plot)
} # show_topals
```

Fit the Italy 1980 data and show the result.

```
fit = TOPALS_fit(N,D,std,
                 age_group_bounds = bb,
                 details=TRUE)

show_topals(fit)
```


Italy Females 1980

238097 deaths to 28873369 women

Evaluating uncertainty

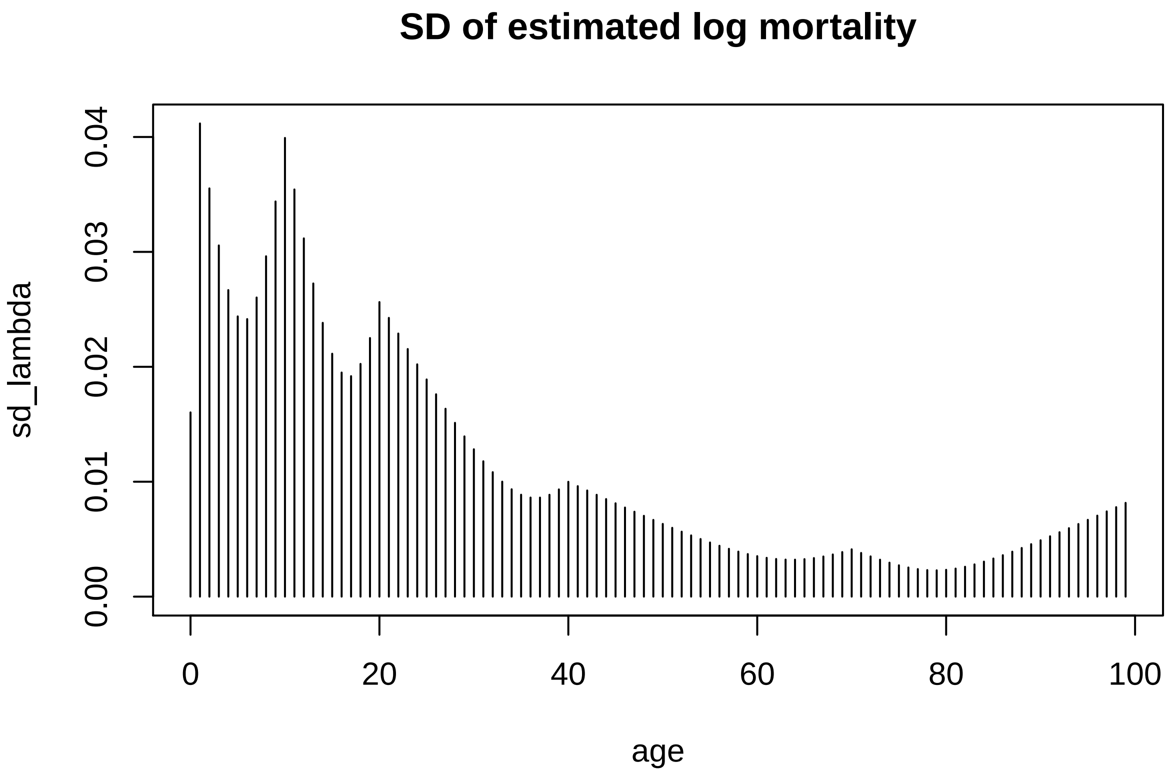
The **TOPALS_fit** function returns an estimate of the 7×7 covariance matrix $V(\boldsymbol{\alpha})$, which is the negative of the Hessian matrix of second derivatives $-\left(\frac{\partial Q}{\partial \boldsymbol{\alpha}\, \partial \boldsymbol{\alpha}'}\right)$. With this estimate we can approximate the 100×100 covariance matrix of log mortality rates as

$$V(\boldsymbol{\lambda}) = \boldsymbol{B}\, V(\boldsymbol{\alpha})\, \boldsymbol{B}'$$

We can use this to build a multinormal approximation to the posterior distribution of the vector of log mortality rates, and thus to simulate the distribution of derived quantities like e_0, life expectancy at birth.

As a start, let's look at the posterior standard deviations of the estimated log mortality rates – these are the square roots of the diagonal elements of $V(\boldsymbol{\lambda})$.

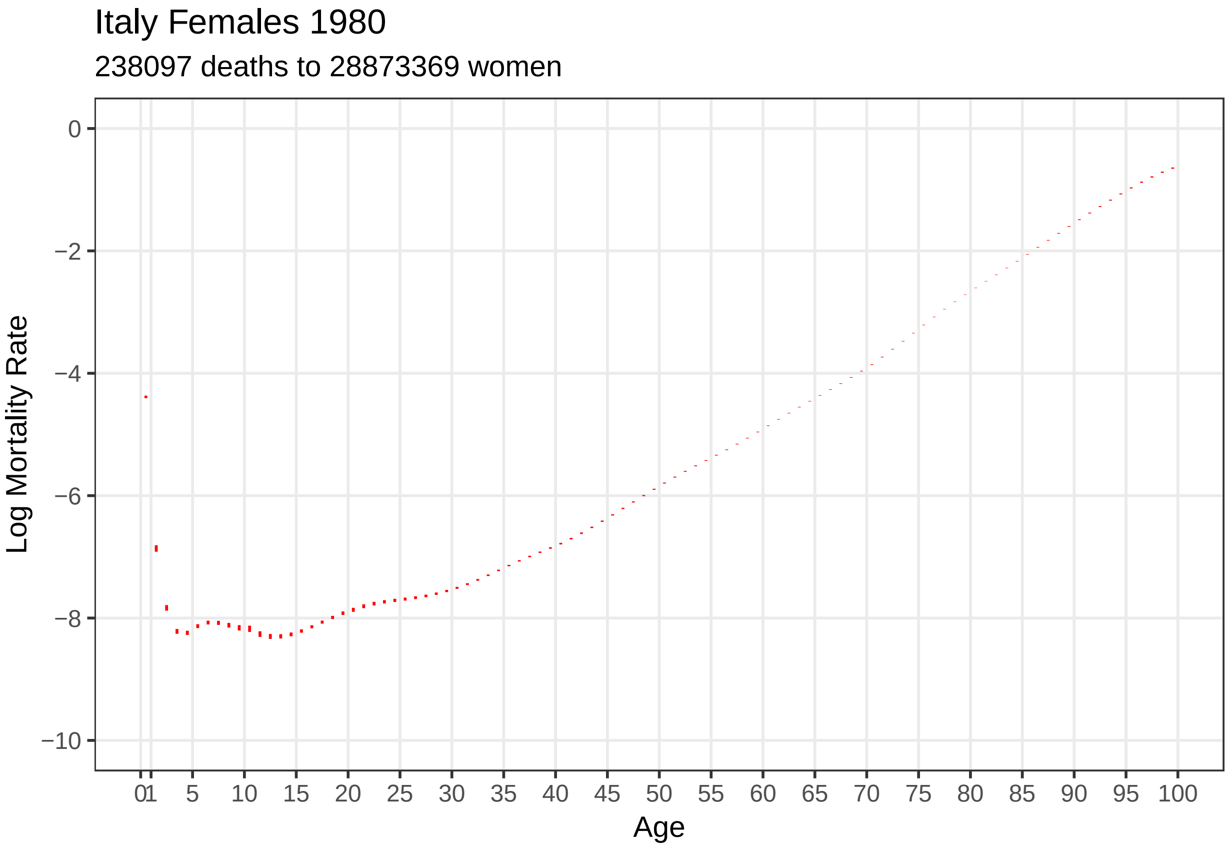
```
sd_lambda = sqrt( diag ( fit$B %*% fit$covar %*% t(fit$B) ) )

plot(0:99, sd_lambda, type='h', xlab='age', ylim=range(0,sd_lambda),
     main='SD of estimated log mortality')
```



So, even with national data in a large country there is some uncertainty about age-specific rates. Notice in this case how the uncertainty increases at the highest ages (85+), for which there was no age-group input data. Estimates at those ages are pure extrapolations based on typical age patterns.

These errors are very small relative to the rates, however, as we can see by plotting the TOPALS estimates with error bars:

```
  show_topals(fit, emphasize_sd = TRUE, hue='red')
```

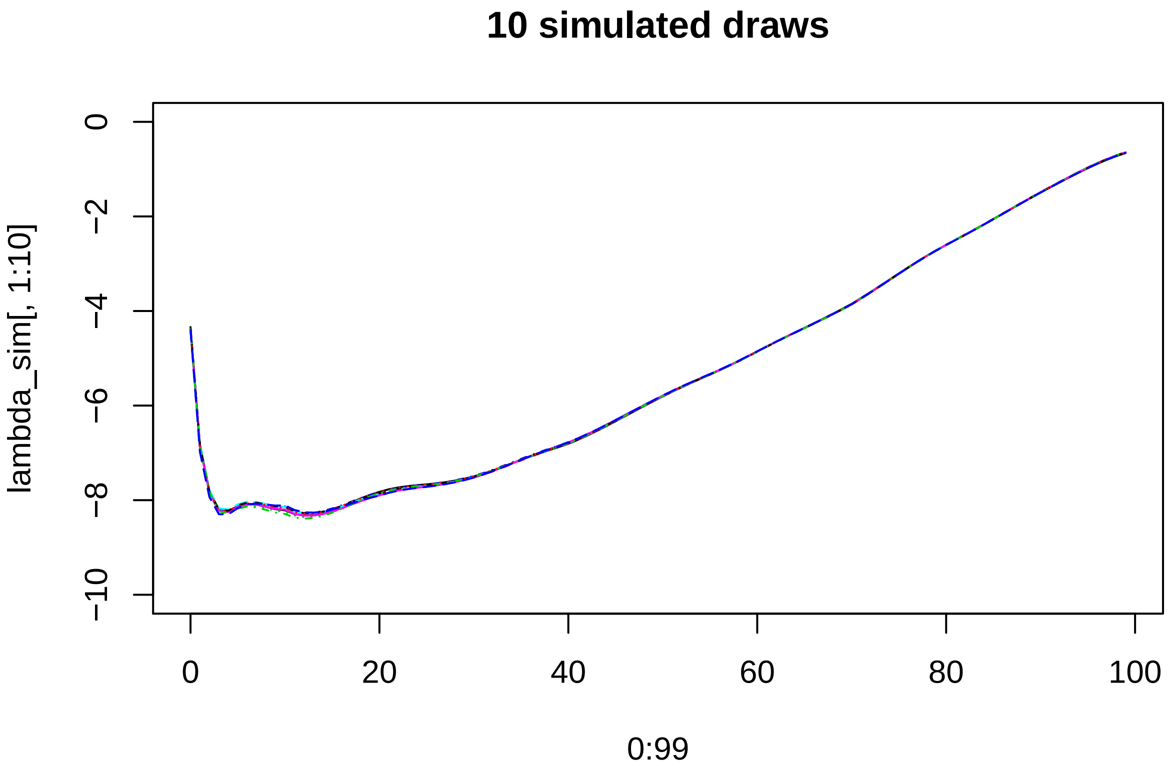


We can also *simulate* draws from the distribution by using a Cholesky decomposition:

$$\tilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda}^* + \boldsymbol{B} \left(\hat{\boldsymbol{\alpha}} + [V(\boldsymbol{\alpha})]^{\frac{1}{2}} \, \boldsymbol{z} \right) \qquad \boldsymbol{z} \sim N(\mathbf{0}, \boldsymbol{I}_7)$$

as in the code below

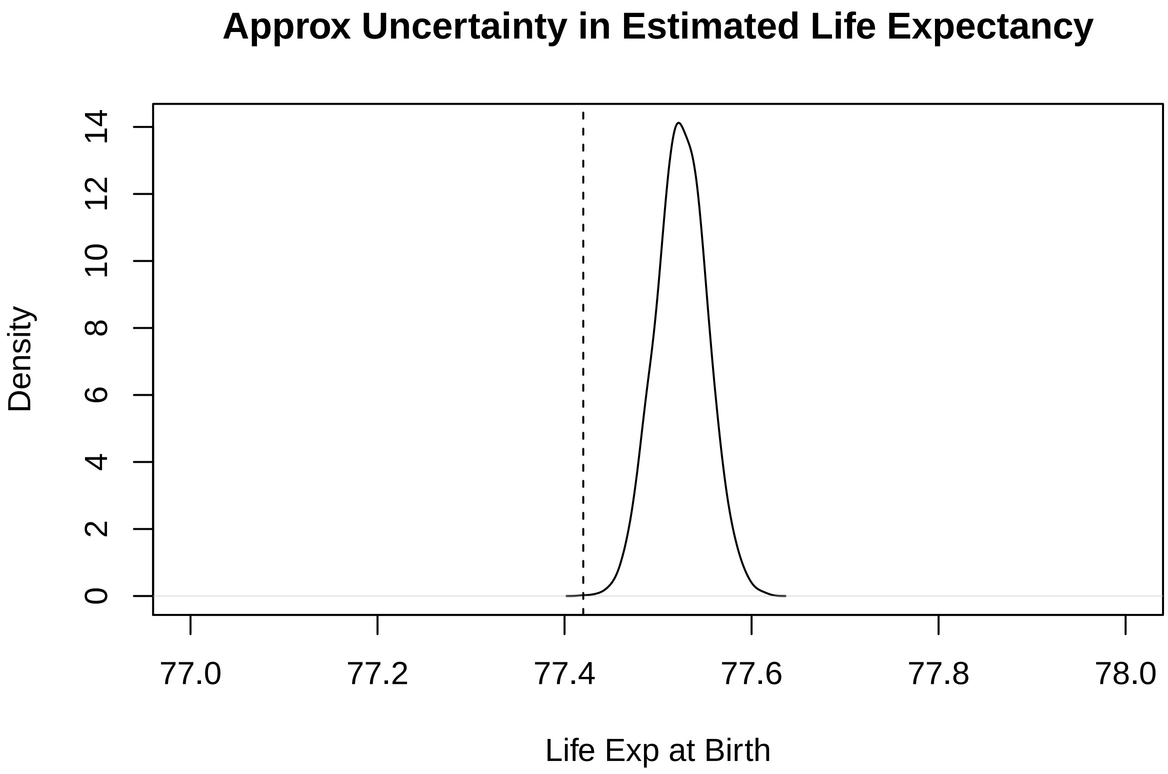
```
      C          = t(chol( fit$covar) )

# generate 10000 draws from the log mortality schedule
      lambda_sim = fit$std + fit$B %*% (fit$alpha + C %*% matrix(rnorm(10000*7), nrow=7))

# plot the first 10 draws
      matplot(0:99, lambda_sim[,1:10], type='l', main='10 simulated draws', ylim=c(-10,0))
```



and we can estimate our uncertainty about Italian female life expectancy estimates from age-group data

```
e0 = function(logmx) {

  x = c(seq(logmx)-1, length(logmx))   # 0,1,...,start open interv

  mx = exp(c(logmx, tail(logmx,1)))  # add open interval
  nx = c( rep(1, length(logmx)), Inf)
  px = exp(-mx * nx)
  lx = c(1,cumprod( head(px,-1)))  # for 0,1,...,start open interv
  ax = 1/mx - 1*(px/(1-px))
  ax[1] = 0.1
  dx = -diff(c(lx,0))
  Lx = lx*px + dx*ax
  return( sum(Lx))
}

esim = apply(lambda_sim,2, e0)
```

```
plot(density(esim , adj=1.5), xlim=c(77,78),
     xlab='Life Exp at Birth',
     main='Approx Uncertainty in Estimated Life Expectancy')

abline(v=77.42, lty=2)  # HMD life exp value
```



TOPALS estimates from grouped data have very little variance for this big population, but there is a small bias : the true e_0 from the HMD is 77.42, while the mean TOPALS estimate is 77.52.

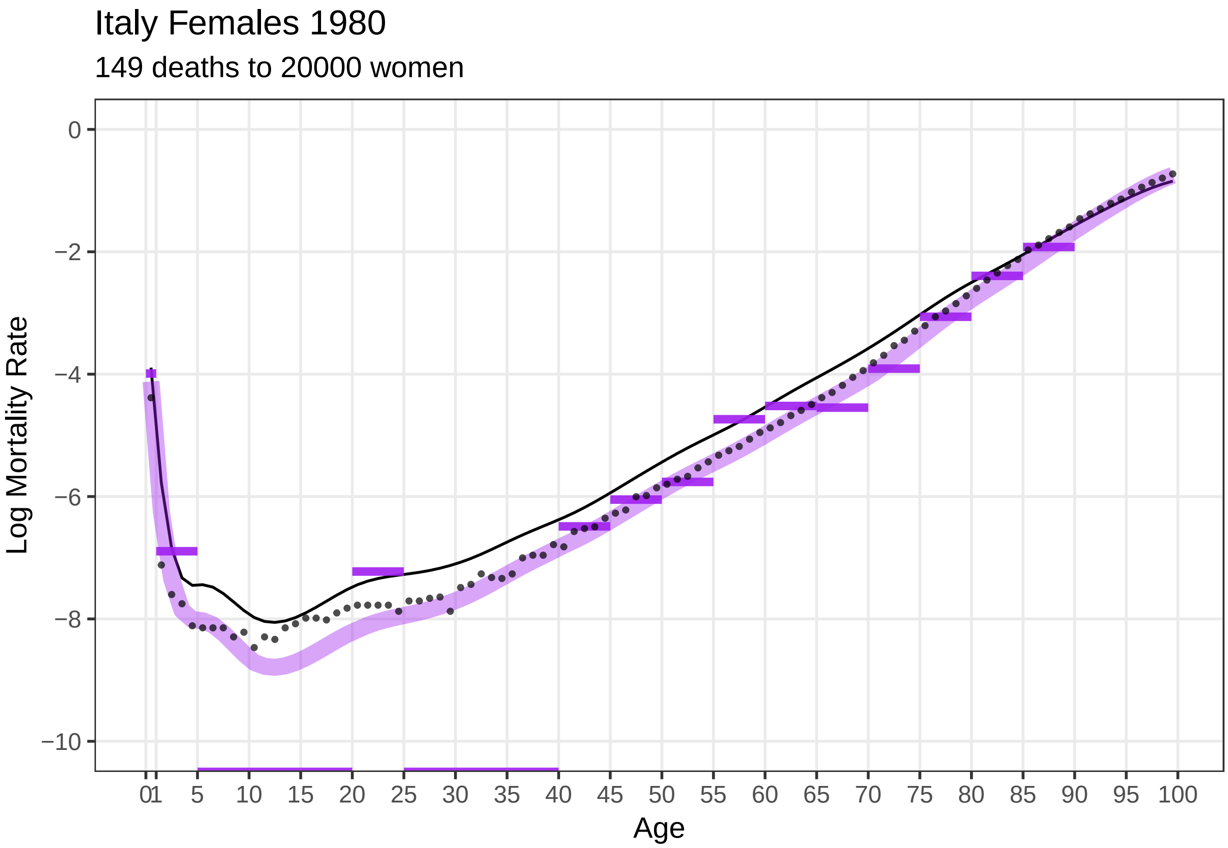
Estimates in small populations

With smaller risk populations there will be fewer deaths, noisier data, and less precise estimates. The procedure above helps us evaluate.

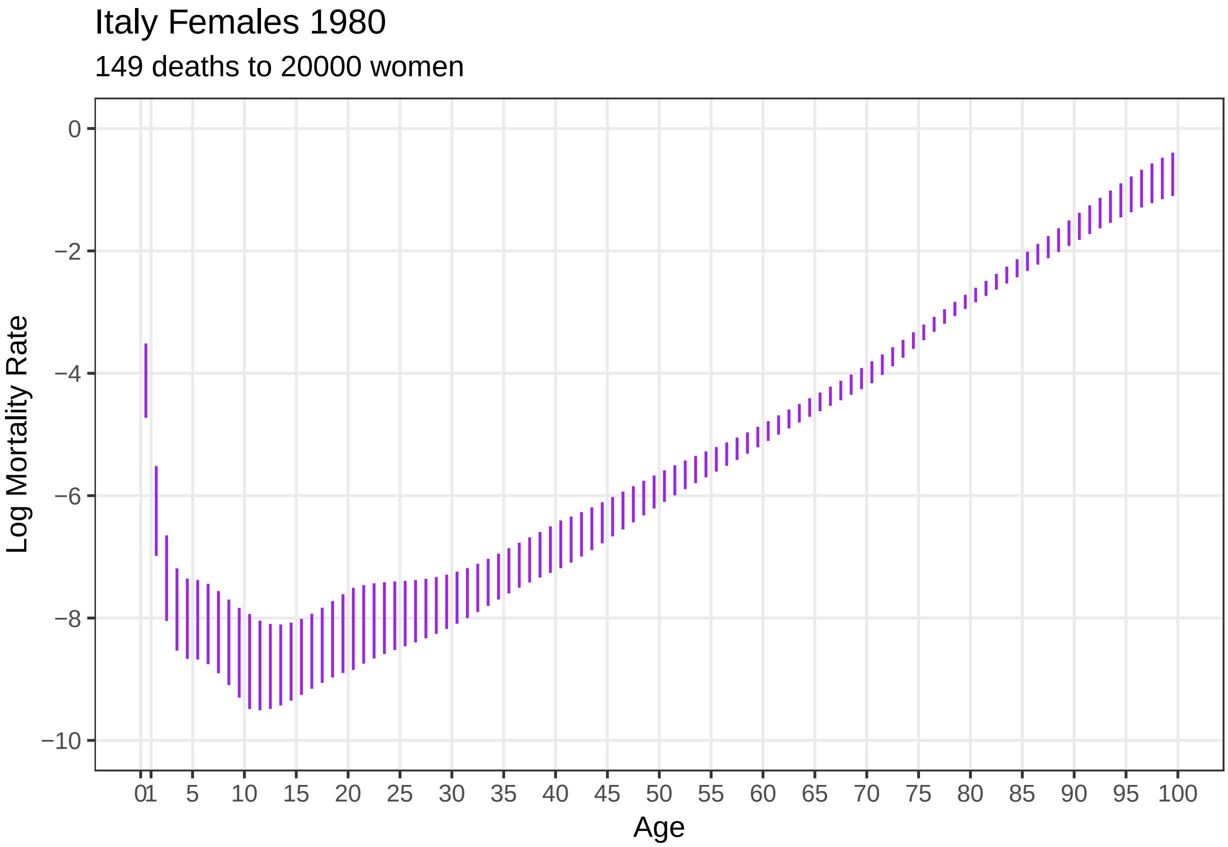
Consider a small female population of 20000 people with Italy's 1980 age structure. We'll simulate single-year deaths, estimate TOPALS from grouped data, and plot uncertainty as before.

```
target_pop = 20000
Nsmall = N * target_pop/sum(N)
Dsmall = rpois( length(Nsmall), Nsmall * D/N)
```

```
fit = TOPALS_fit(Nsmall, Dsmall, std,
                 age_group_bounds = bb,
                 details=TRUE)

show_topals(fit, hue='purple')
```

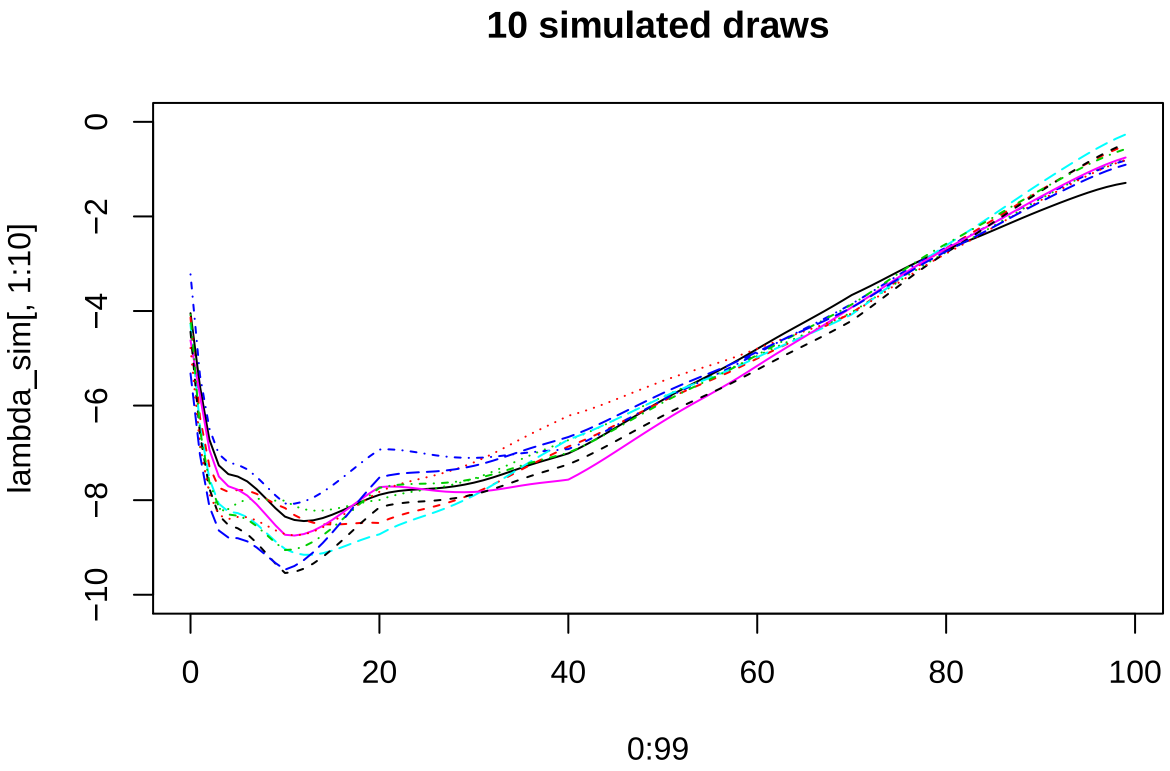


```
show_topals(fit, emphasize_sd = TRUE, hue='purple')
```

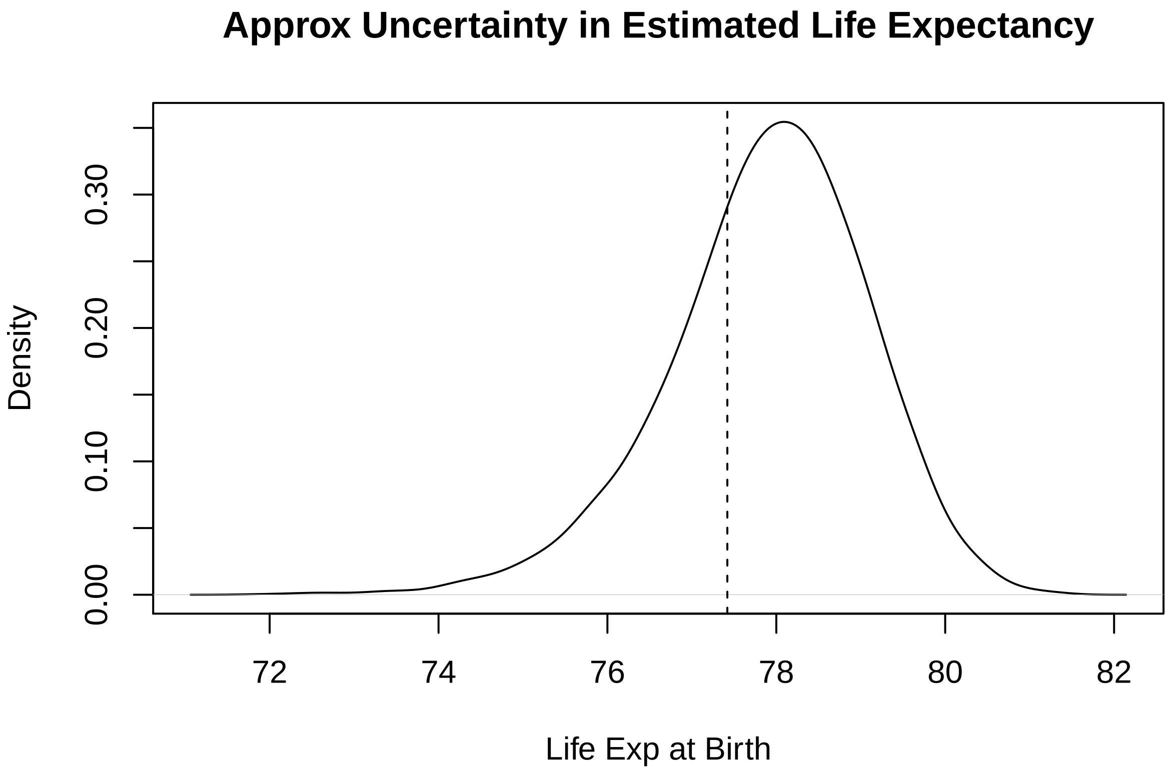

Italy Females 1980

149 deaths to 20000 women

```r
# generate 10000 draws from the log mortality schedule
      C = t(chol( fit$covar) )

      lambda_sim = fit$std +
         fit$B %*% (fit$alpha + C %*% matrix(rnorm(10000*7), nrow=7))

# plot the first 10 draws
      matplot(0:99, lambda_sim[,1:10], type='l', main='10 simulated draws', ylim=c(-10,0))
```



```
esim = apply(lambda_sim,2, e0)

plot(density(esim , adj=1.5),
     xlab='Life Exp at Birth',
     main='Approx Uncertainty in Estimated Life Expectancy')

abline(v=77.42, lty=2)  # HMD life exp value
```


Approx Uncertainty in Estimated Life Expectancy
Density
0.00
0.10
0.20
0.30
72
74
76
78
80
82
Life Exp at Birth