

Fitting a TOPALS mortality model with age-grouped data, by Penalized Iteratively Weighted Least Squares (PIRLS)

Carl Schmertmann

21 Aug 2019, updated 14 Oct 2021 (cleaned up two minor notational errors), re-rendered
verbatim 08 Aug 2022

Main Idea

The objective is to fit a TOPALS model mortality schedule for A single years of age $x = 0, 1, \dots, (A - 1)$ from exposure and death data for $G \leq A$ age groups: $N_1 \dots N_G$ and deaths $D_1 \dots D_G$.

Sample Data

Observed data consists of deaths D_g and exposure N_g for closed, non-overlapping age groups $g = 1 \dots G$. Age groups are defined by a vector of $G + 1$ boundary ages – e.g. if the boundary ages are 0, 1, 5, 10, \dots , 85, 90 then the age groups are $[0, 1), [1, 5), \dots, [85, 90)$. Denote X_g as the set of integer ages that belong to group g , and n_g as the number of ages in X_g .

Typical age groups have boundaries like 0, 1, 5, 10, \dots , 85, 90. However, if single-year death and exposure data is available then the “groups” could be simply integer ages, in which case boundaries for A single-year “groups” would be 0, 1, 2, \dots , A

Poisson Log Likelihood for Grouped Data

Regardless of the age grouping in the data, we assume that there is a latent schedule for A single-year ages. Deaths at each single-year age have a Poisson distribution with expected value equal to the (possibly unobserved) exposure N_x times the mortality rate:

$$D_x \sim \text{Poisson}(N_x \mu_x) \quad x = 0, 1, 2, \dots, (A - 1)$$

Under the standard assumption that age-specific deaths are statistically independent, this implies that deaths are also Poisson distributed within each age group:

$$D_g \sim \text{Poisson}\left(\sum_{x \in X_g} N_x \mu_x\right) \quad g = 1, 2, \dots, G$$

Expressing this in terms of the total *observed* exposure in each age group,

$$D_g \sim \text{Poisson} \left(N_g \sum_{x \in X_g} \left[\frac{N_x}{N_g} \right] \mu_x \right)$$

or

$$D_g \sim \text{Poisson} (N_g M_g)$$

where M_g represents the exposure-weighted average mortality rate across ages in group g .

In the absence of single-year exposure data, we approximate with $M_g = [\frac{1}{n_g}] \sum_{x \in X_g} \mu_x$.¹ The entire vector of G averaged mortality rates by age group is therefore

$$\mathbf{M} = \begin{pmatrix} n_1^{-1} \dots n_1^{-1} & \dots & 0 \dots 0 \\ \vdots & \ddots & \vdots \\ 0 \dots 0 & \dots & n_G^{-1} \dots n_G^{-1} \end{pmatrix} \begin{pmatrix} \mu_0 \\ \vdots \\ \mu_{A-1} \end{pmatrix} = \mathbf{W} \boldsymbol{\mu}$$

where \mathbf{W} is a $G \times A$ matrix of weights with row sums all equal to one.²

The sample log likelihood for a single-year mortality schedule $\boldsymbol{\mu}$ is thus

$$\begin{aligned} \ln L(\boldsymbol{\mu}) &= c + \sum_{g=1}^G (D_g \ln M_g - N_g M_g) \\ &= c + \sum_{g=1}^G (D_g \ln M_g - \hat{D}_g) \end{aligned}$$

where $\hat{D}_g = N_g M_g$ represents the expected number of deaths in group g . It will also be useful to express the likelihood in matrix terms:

$$\ln L = c + [\ln M_1 \dots \ln M_G] \mathbf{D}_{obs} - \hat{\mathbf{D}}' \mathbf{1} \quad (1)$$

where \mathbf{D}_{obs} and $\hat{\mathbf{D}}$ are $G \times 1$ vectors of observed and expected deaths, ordered by age group, and $\mathbf{1}$ is a $G \times 1$ vector of ones.

TOPALS Model Mortality Schedule

The TOPALS model for single-year mortality rates is

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}^* + \mathbf{B} \boldsymbol{\alpha}$$

¹An alternative approach might interpolate single-year exposure $(N_0, N_1, \dots, N_{A-1})$ from the available N_g to construct more nuanced weights for intra-group averaging of mortality rates. In most populations this would have only minor effects on estimates.

²When "groups" correspond to single years, $\mathbf{W} = \mathbf{I}_A$ and $\mathbf{M} = \boldsymbol{\mu}$.

where $\boldsymbol{\lambda}$ is an $A \times 1$ vector of age-specific log mortality rates for ages $0 \dots (A-1)$, $\boldsymbol{\lambda}^*$ is a vector with fixed constants representing a standard schedule for those ages, \mathbf{B} is a $A \times K$ matrix of linear B-spline constants³, and $\boldsymbol{\alpha}$ is a K -dimensional vector representing deviations from the standard log mortality schedule at specified ages.

In this model the log mortality rate at age $x \in \{0 \dots (A-1)\}$ is

$$\ln \mu_x = \lambda_x = \lambda_x^* + \mathbf{b}_x' \boldsymbol{\alpha}$$

and the mortality rate is

$$\mu_x = \exp(\lambda_x^* + \mathbf{b}_x' \boldsymbol{\alpha})$$

where \mathbf{b}_x' is the $1 \times K$ row of \mathbf{B} that corresponds to age x .

TOPALS derivatives

Derivatives of age-specific rates with respect to TOPALS parameters $\boldsymbol{\alpha}$ are $K \times 1$ vectors:

$$\frac{\partial \ln \mu_x}{\partial \boldsymbol{\alpha}} = \mathbf{b}_x \quad , \quad \frac{\partial \mu_x}{\partial \boldsymbol{\alpha}} = \mu_x \mathbf{b}_x$$

For the entire $G \times 1$ vector of group mortality rates, this implies

$$\begin{aligned} \frac{\partial \mathbf{M}'}{\partial \boldsymbol{\alpha}} &= \frac{\partial}{\partial \boldsymbol{\alpha}} (\boldsymbol{\mu}' \mathbf{W}') \\ &= \left(\frac{\partial \mu_0}{\partial \boldsymbol{\alpha}} \quad \dots \quad \frac{\partial \mu_{A-1}}{\partial \boldsymbol{\alpha}} \right) \mathbf{W}' \\ &= \left(\mu_0 \mathbf{b}_0 \quad \dots \quad \mu_{A-1} \mathbf{b}_{A-1} \right) \mathbf{W}' \\ &= \mathbf{B}' \text{diag}(\boldsymbol{\mu}) \mathbf{W}' \end{aligned}$$

Abbreviate this as

$$\frac{\partial \mathbf{M}'}{\partial \boldsymbol{\alpha}} = \mathbf{X}'$$

remembering that the $G \times K$ matrix $\mathbf{X} = \mathbf{W} \text{diag}(\boldsymbol{\mu}) \mathbf{B}$ varies with parameters $\boldsymbol{\alpha}$ via the μ terms in the central diagonal matrix.

The $G \times 1$ vector of expected deaths in groups $1 \dots G$ is

$$\hat{\mathbf{D}} = \begin{pmatrix} N_1 M_1 \\ \vdots \\ N_G M_G \end{pmatrix} = \text{diag}(\mathbf{N}) \mathbf{M}$$

³Typically $K = 7$, with spline knots fixed at ages $t = 0, 1, 10, 20, 40, 70, 99$. Values in the k th column of \mathbf{B} are

$$B_{xk} = \begin{cases} \frac{x - t_{k-1}}{t_k - t_{k-1}} & \text{for } x \in [t_{k-1}, t_k] \\ \frac{t_{k+1} - x}{t_{k+1} - t_k} & \text{for } x \in [t_k, t_{k+1}] \\ 0 & \text{otherwise} \end{cases}$$

and the derivative of its transpose with respect to TOPALS parameters α is a $K \times G$ matrix:

$$\frac{\partial \hat{\mathbf{D}}'}{\partial \alpha} = \frac{\partial \mathbf{M}'}{\partial \alpha} \text{diag}(\mathbf{N}) = \mathbf{X}' \text{diag}(\mathbf{N}) \quad (2)$$

Penalized Log Likelihood for TOPALS parameters

With the TOPALS parameterization the log likelihood for a sample $\{D_g, N_g\}$ is

$$\begin{aligned} \ln L(\alpha) &= c + \sum_{g=1}^G \left(D_g \ln M_g(\alpha) - \hat{D}_g(\alpha) \right) \\ &= c + [\ln M_1 \cdots \ln M_G] \mathbf{D}_{obs} - \hat{\mathbf{D}}' \mathbf{1} \end{aligned}$$

To stabilize estimates in small populations with few deaths, we add a small penalty term to the log likelihood:

$$\begin{aligned} Q(\alpha) &= \ln L(\alpha) - \text{penalty}(\alpha) \\ &= c + [\ln M_1 \cdots \ln M_G] \mathbf{D} - \hat{\mathbf{D}}' \mathbf{1} - \frac{1}{2} \alpha' \mathbf{P} \alpha \end{aligned} \quad (3)$$

where \mathbf{P} is a $K \times K$ matrix of constants selected so that the penalty term equals the sum of squared differences between consecutive α parameters – i.e., $\frac{1}{2} \alpha' \mathbf{P} \alpha = \sum_{k=2}^K (\alpha_k - \alpha_{k-1})^2$. Adding the penalty gives priority to sets of TOPALS parameters $(\alpha_1 \dots \alpha_K)$ that are similar to one another, and thus to log mortality schedules that look more like up-and-down vertical shifts of the standard schedule. For all but the smallest populations the penalty term has virtually no effect on parameter estimates. For very small populations with zero deaths in some age groups, the addition of the penalty stabilizes estimated schedules by borrowing strength across groups.

Maximizing via Iteratively Reweighted Regression

We want to select α to maximize Q . This requires setting a vector of derivatives equal to zero: $\frac{\partial Q}{\partial \alpha} = 0 \in \mathbb{R}^K$.

Differentiating Eq. (3) with respect to the TOPALS parameters and substituting some of the results above produces

$$\begin{aligned} \frac{\partial Q}{\partial \alpha} &= \left[\frac{1}{M_1} \frac{\partial M_1}{\partial \alpha} \cdots \frac{1}{M_G} \frac{\partial M_G}{\partial \alpha} \right] \mathbf{D}_{obs} - \frac{\partial \hat{\mathbf{D}}'}{\partial \alpha} \mathbf{1} - \mathbf{P} \alpha \\ &= \frac{\partial \mathbf{M}'}{\partial \alpha} \text{diag}\left(\frac{1}{\mathbf{M}}\right) \mathbf{D}_{obs} - \frac{\partial \mathbf{M}'}{\partial \alpha} \text{diag}(\mathbf{N}) \mathbf{1} - \mathbf{P} \alpha \\ &= \mathbf{X}' \text{diag}\left(\frac{1}{\mathbf{M}}\right) \mathbf{D}_{obs} - \mathbf{X}' \mathbf{N} - \mathbf{P} \alpha \\ &= \mathbf{X}' \text{diag}\left(\frac{1}{\mathbf{M}}\right) \mathbf{D}_{obs} - \mathbf{X}' \text{diag}\left(\frac{1}{\mathbf{M}}\right) \hat{\mathbf{D}} - \mathbf{P} \alpha \\ &= \mathbf{X}' \text{diag}\left(\frac{1}{\mathbf{M}}\right) (\mathbf{D}_{obs} - \hat{\mathbf{D}}) - \mathbf{P} \alpha \end{aligned}$$

For reasons that will become clear in a moment, it is convenient to define the abbreviation $\mathbf{A} = \text{diag}(\frac{N_1}{M_1} \dots \frac{N_G}{M_G})$. Using this new abbreviation,

$$\frac{\partial Q}{\partial \boldsymbol{\alpha}} = \mathbf{X}' \mathbf{A} \text{diag}(\frac{1}{\mathbf{N}})(\mathbf{D}_{obs} - \hat{\mathbf{D}}) - \mathbf{P} \boldsymbol{\alpha} = \mathbf{0} \in \mathbb{R}^K \quad (4)$$

Rearranging Eq. (4) and placing hats over the $\boldsymbol{\alpha}$ -dependent terms yields a system of K equations in K variables $\boldsymbol{\alpha}$

$$\hat{\mathbf{X}}' \hat{\mathbf{A}} \text{diag}(\frac{\hat{\mathbf{D}}}{\mathbf{N}}) + \mathbf{P} \hat{\boldsymbol{\alpha}} = \hat{\mathbf{X}}' \hat{\mathbf{A}} \text{diag}(\frac{\mathbf{D}_{obs}}{\mathbf{N}}) \quad (5)$$

The iteratively reweighted least squares approach to solving the system of equations in (5) begins by approximating how predicted values $\hat{\mathbf{D}}$ would change if the vector of TOPALS parameters changed from $\hat{\boldsymbol{\alpha}}$ to some other value $\boldsymbol{\alpha}^*$, while (temporarily) keeping matrices $\hat{\mathbf{X}}$ and $\hat{\mathbf{A}}$ constant:

$$\hat{\mathbf{X}}' \hat{\mathbf{A}} \text{diag}(\frac{1}{\mathbf{N}}) \left[\hat{\mathbf{D}} + \frac{\partial \hat{\mathbf{D}}}{\partial \boldsymbol{\alpha}'} (\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}) \right] + \mathbf{P} \boldsymbol{\alpha}^* = \hat{\mathbf{X}}' \hat{\mathbf{A}} \text{diag}(\frac{\mathbf{D}_{obs}}{\mathbf{N}}) \quad (6)$$

which from Eq. (2) expands to

$$\hat{\mathbf{X}}' \hat{\mathbf{A}} \text{diag}(\frac{1}{\mathbf{N}}) \left[\hat{\mathbf{D}} + \text{diag}(\mathbf{N}) \hat{\mathbf{X}} (\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}) \right] + \mathbf{P} \boldsymbol{\alpha}^* = \hat{\mathbf{X}}' \hat{\mathbf{A}} \text{diag}(\frac{1}{\mathbf{N}}) \mathbf{D}_{obs} \quad (7)$$

This can be rearranged as

$$\hat{\mathbf{X}}' \hat{\mathbf{A}} \hat{\mathbf{X}} (\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}) + \mathbf{P} \boldsymbol{\alpha}^* = \hat{\mathbf{X}}' \hat{\mathbf{A}} \text{diag}(\frac{1}{\mathbf{N}})(\mathbf{D}_{obs} - \hat{\mathbf{D}})$$

and further simplified as

$$\left[\hat{\mathbf{X}}' \hat{\mathbf{A}} \hat{\mathbf{X}} + \mathbf{P} \right] \boldsymbol{\alpha}^* = \hat{\mathbf{X}}' \hat{\mathbf{A}} \left[\text{diag}(\frac{1}{\mathbf{N}})(\mathbf{D}_{obs} - \hat{\mathbf{D}}) + \hat{\mathbf{X}} \hat{\boldsymbol{\alpha}} \right]$$

and finally simplified to

$$\left[\hat{\mathbf{X}}' \hat{\mathbf{A}} \hat{\mathbf{X}} + \mathbf{P} \right] \boldsymbol{\alpha}^* = \hat{\mathbf{X}}' \hat{\mathbf{A}} \hat{\mathbf{y}} \quad (8)$$

where $\hat{\mathbf{y}} = \text{diag}(\frac{1}{\mathbf{N}})(\mathbf{D}_{obs} - \hat{\mathbf{D}}) + \hat{\mathbf{X}} \hat{\boldsymbol{\alpha}}$.

The solution $\boldsymbol{\alpha}^*$ to Eq. (8) is also the coefficient estimate from a (penalized) weighted least squares regression of $\hat{\mathbf{y}}$ on $\hat{\mathbf{X}}$ with weights $\hat{\mathbf{A}}^{-1}$.

Starting from an arbitrary vector $\hat{\boldsymbol{\alpha}}$, one can solve for the vector that maximizes penalized likelihood $Q(\boldsymbol{\alpha})$ by

1. calculating the values of $(\hat{\mathbf{X}}, \hat{\mathbf{A}}, \hat{\mathbf{D}})$ at the current parameter value
2. solving eq. (8) for $\boldsymbol{\alpha}^*$, and
3. repeating until convergence.

This is the algorithm used in the **TOPALS_fit** function in the next sections.

Example: A TOPALS fit from five-year age groups for Italian Females in 1980

Age groups for this exercise are 0, 1–4, 5–9, ..., 80–85.

```
library(tidyverse, quietly = TRUE)
library(splines, quietly = TRUE)
```

Display the code for the *TOPALS_fit()* function

```
#-----
# TOPALS fitting function for mortality schedules
#
# Carl Schmertmann
#   created 01 Mar 2018
#   edited 21 Aug 2019
#
# + added fitted log mort, basis, etc to detailed output
# + added ability to fit age-grouped as well as single-yr data
# + changed fitting algorithm from Newton-Raphson
#   to (penalized) IRLS
#
# Fits TOPALS parameters to single-year or age-group (D,N)
# data by penalized IRLS with analytical derivatives
#
#
# age_group_bounds is a (G+1) vector of ages v that define
# G closed age groups [v1,v2), [v2,v3)... [vG,vG+1)
# For ex. if age_group_bounds = c(0,1,5,10,15,...,90)
# then the age groups are [0,1), [1,5), [5,10), ..., [85,90)
#
# A more complete explanation is in
# https://github.com/schmert/TOPALS/blob/master/TOPALS_fitting_with_grouped_data.pdf
#-----

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         = 50,
```

```

        alpha_tol      = .00005,
        details         = FALSE) {

require(splines)

## single years of age from 0 to (A-1)
A  = length(std)
age = 0:(A-1)

## B is an AxK matrix. Each column is a linear B-spline basis function
B   = bs( age, knots=knot_positions, degree=1 )
K = ncol(B)

D1 = diff( diag(K), diff=1)
P  = penalty_precision * crossprod(D1)

## number and width of age groups
G   = length(age_group_bounds)-1
nages = diff(age_group_bounds)

## weighting matrix for mortality rates (assumes uniform
## distribution of single-year ages within groups)
W = matrix(0, nrow=G, ncol=A,
           dimnames=list(head(age_group_bounds,-1) , age))

offset = 0
for (g in 1:G) {
  W[g, offset + 1:nages[g]] = 1/nages[g]
  offset = offset + nages[g]
}

## penalized log lik function
Q = function(alpha) {
  M = W %*% exp( std + B %*% alpha)
  likelihood = sum(D * log(M) - N * M)
  penalty     = 1/2 * t(alpha) %*% P %*% alpha
  return( likelihood - penalty )
}

```

```

#-----
# iteration function:
# next alpha vector as a function of current alpha
#-----
next_alpha = function(alpha) {
  mu = as.vector( exp( std + B %%% alpha))
  M  = as.vector( W %%% mu)

  Dhat = N * M

  X = W %%% diag(mu) %%% B
  A = diag(N/M)

  y = (D-Dhat)/N + X %%% alpha

  updated_alpha = solve( t(X) %%% A %%% X + P, t(X) %%% A %%% y)
  return(as.vector(updated_alpha))
}

## main iteration:
a = rep(0, K)

niter = 0
repeat {
  niter      = niter + 1
  last_param = a
  a          = next_alpha( a ) # update
  change     = a - last_param

  converge = all( abs(change) < alpha_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')
}

```



```

mu    = as.vector( exp(std + B %*% a))
M     = as.vector( W %*% mu )
dhat  = N * M

X     = W %*% diag(mu) %*% B
A     = diag(N/M)

covar = solve( t(X) %*% A %*% X + P)

return( list( alpha      = a,
              D          = D,
              N          = N,
              age_group_bounds = age_group_bounds,
              knots       = knot_positions,
              std         = std,
              B           = B,
              logm        = std + B %*% a,
              covar       = covar,
              Qvalue      = Q(a),
              converge     = converge,
              maxiter      = overrun))
} else return( a)

} # TOPALS_fit

# include the fitting function
source('TOPALS_fit.R', echo=FALSE, local = knitr::knit_global())

## Italian female 1980 HMD data for age groups

boundaries = c(0,1,seq(5,85,5)) # last group is [80,85)

N = c(312106.85, 1423566.3, 2105814.63, 2249555.41, 2230885.74, 1983157.8,
1874479.58, 1989351.99, 1772133.73, 1896866.51, 1836597.36, 1834496.64,
1811178.38, 1192763.85, 1498384.03, 1223810.9, 863725.92, 537720.77)

```

```

D = c(3889, 716, 587, 589, 791, 816, 832, 1257, 1651, 2721, 4310,
6636, 10536, 11043, 23312, 34945, 44537, 50392)

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

## standard schedule is (very arbitrarily!) the 1959 period
## schedule for Canadian females at ages 0,1,...,99
## these are the log mortality rates
std = c(-3.8933, -5.7776, -6.8474, -7.3298, -7.4519, -7.4408, -7.4807,
-7.5845, -7.7219, -7.8628, -7.9771, -8.041, -8.0568, -8.0329,
-7.9779, -7.9005, -7.8088, -7.7101, -7.6113, -7.5195, -7.4415,
-7.3823, -7.3393, -7.308, -7.2837, -7.2619, -7.238, -7.2082,
-7.1712, -7.1264, -7.0735, -7.0118, -6.9414, -6.8648, -6.7849,
-6.7047, -6.6272, -6.5544, -6.4845, -6.4147, -6.3423, -6.2645,
-6.1791, -6.0872, -5.9904, -5.8903, -5.7887, -5.6869, -5.586,
-5.4866, -5.3895, -5.2953, -5.2049, -5.1186, -5.0347, -4.9513,
-4.8664, -4.778, -4.6847, -4.5877, -4.4887, -4.3895, -4.2918,
-4.1969, -4.1041, -4.0122, -3.9199, -3.8261, -3.7296, -3.6303,
-3.5278, -3.4221, -3.3129, -3.2004, -3.0861, -2.9716, -2.8589,
-2.7497, -2.6458, -2.5482, -2.4556, -2.3659, -2.2771, -2.187,
-2.0942, -1.9991, -1.9028, -1.8062, -1.7105, -1.6164, -1.5242,
-1.434, -1.3458, -1.2596, -1.1758, -1.0958, -1.0212, -0.9535,
-0.8944, -0.8454)

## single-year log mortality rates from HMD
## these are the targets for TOPALS estimation
ITA_HMD_logmx =
  c(-4.3852, -7.1185, -7.6009, -7.7517, -8.1117, -8.1456, -8.1456,
-8.1456, -8.294, -8.2171, -8.4684, -8.294, -8.3349, -8.1456,
-8.0789, -7.9866, -7.9866, -8.0164, -7.902, -7.824, -7.7753,
-7.7753, -7.7753, -7.7753, -7.8753, -7.7063, -7.7063, -7.6628,
-7.6417, -7.8753, -7.4876, -7.4354, -7.2644, -7.3233, -7.3385,
-7.2644, -7.0021, -6.959, -6.959, -6.7855, -6.8216, -6.5713,
-6.5225, -6.4956, -6.3539, -6.2712, -6.2196, -6.0035, -5.9835,
-5.8569, -5.7992, -5.7169, -5.6694, -5.5315, -5.433, -5.3247,
-5.2514, -5.1814, -5.0625, -4.9533, -4.8783, -4.7915, -4.6767,
-4.5923, -4.4945, -4.3836, -4.2992, -4.1825, -4.0513, -3.9409,
-3.8135, -3.6913, -3.5332, -3.4455, -3.2966, -3.2069, -3.0614,

```

```
-2.9677, -2.8466, -2.7201, -2.5974, -2.4617, -2.3462, -2.2249,
-2.1253, -1.9713, -1.8905, -1.7861, -1.6842, -1.5945, -1.4583,
-1.3792, -1.297, -1.2087, -1.1393, -1.0245, -0.9444, -0.8681,
-0.7958, -0.7276)
```

```
show = function(fit, 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_HMD_logmx,
    logmx_fit = fit$logm
  )
```

```
this_plot =
  ggplot(data = df_single, aes(x=age,y=logmx_true)) +
    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) +
    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()

    print(this_plot)
} # show

```

Now fit the data and show the result.

```

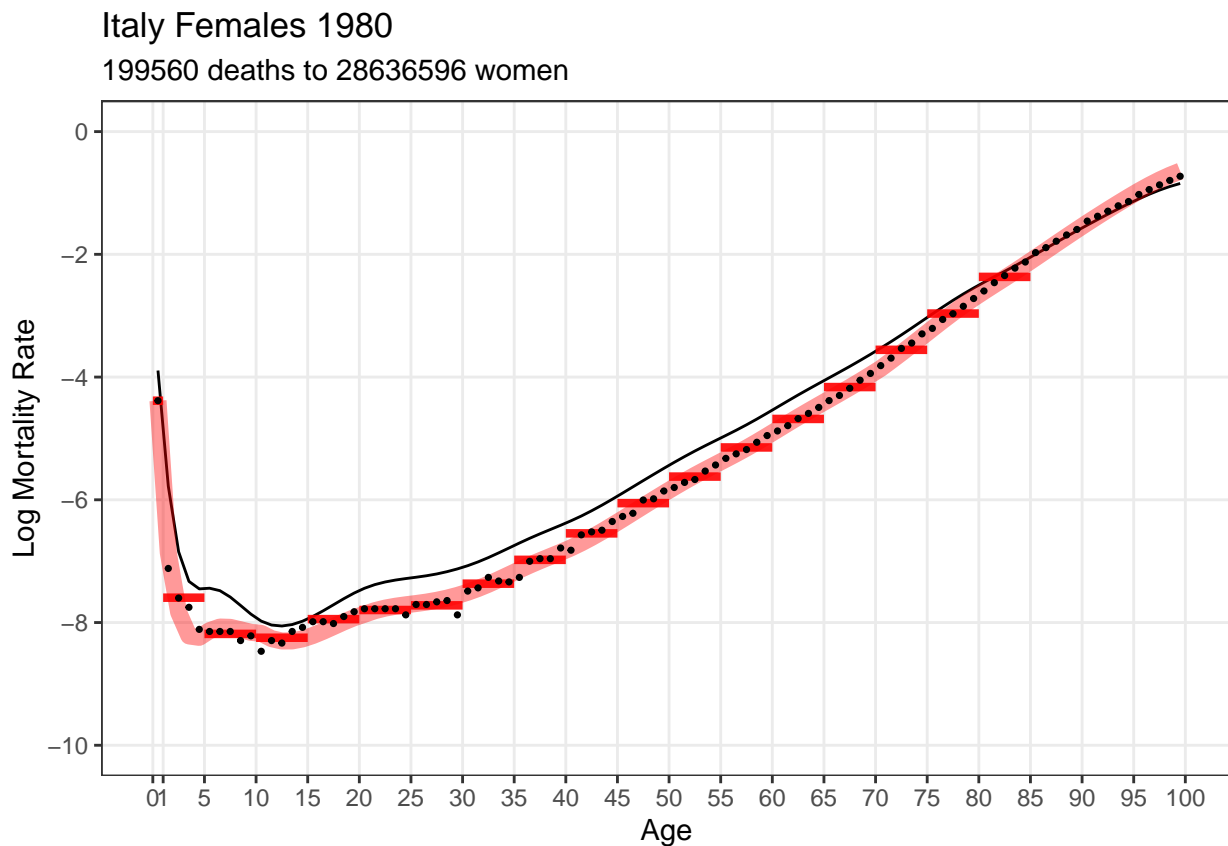
fit = TOPALS_fit(N,D,std,
                 age_group_bounds = boundaries,
                 details=TRUE)

str(fit)

## List of 12
## $ alpha          : num [1:7] -0.492 -1.086 -0.199 -0.424 -0.442 ...
## $ D              : Named num [1:18] 3889 716 587 589 791 ...
##   ..- attr(*, "names")= chr [1:18] "0" "1" "5" "10" ...
## $ N              : Named num [1:18] 312107 1423566 2105815 2249555 2230886 ...
##   ..- attr(*, "names")= chr [1:18] "0" "1" "5" "10" ...
## $ age_group_bounds: num [1:19] 0 1 5 10 15 20 25 30 35 40 ...
## $ knots          : num [1:6] 0 1 10 20 40 70
## $ std            : num [1:100] -3.89 -5.78 -6.85 -7.33 -7.45 ...
## $ B              : 'bs' num [1:100, 1:7] 1 0 0 0 0 0 0 0 0 0 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:7] "1" "2" "3" "4" ...
##   ..- attr(*, "degree")= int 1
##   ..- attr(*, "knots")= num [1:6] 0 1 10 20 40 70
##   ..- attr(*, "Boundary.knots")= int [1:2] 0 99
##   ..- attr(*, "intercept")= logi FALSE
## $ logm           : num [1:100, 1] -4.39 -6.86 -7.83 -8.22 -8.24 ...
## $ covar          : num [1:7, 1:7] 2.57e-04 8.72e-07 -2.53e-07 3.69e-08 -2.84e-09 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:7] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:7] "1" "2" "3" "4" ...
## $ Qvalue         : num [1, 1] -931444
## $ converge       : logi TRUE
## $ maxiter        : logi FALSE

```

```
show(fit)
```



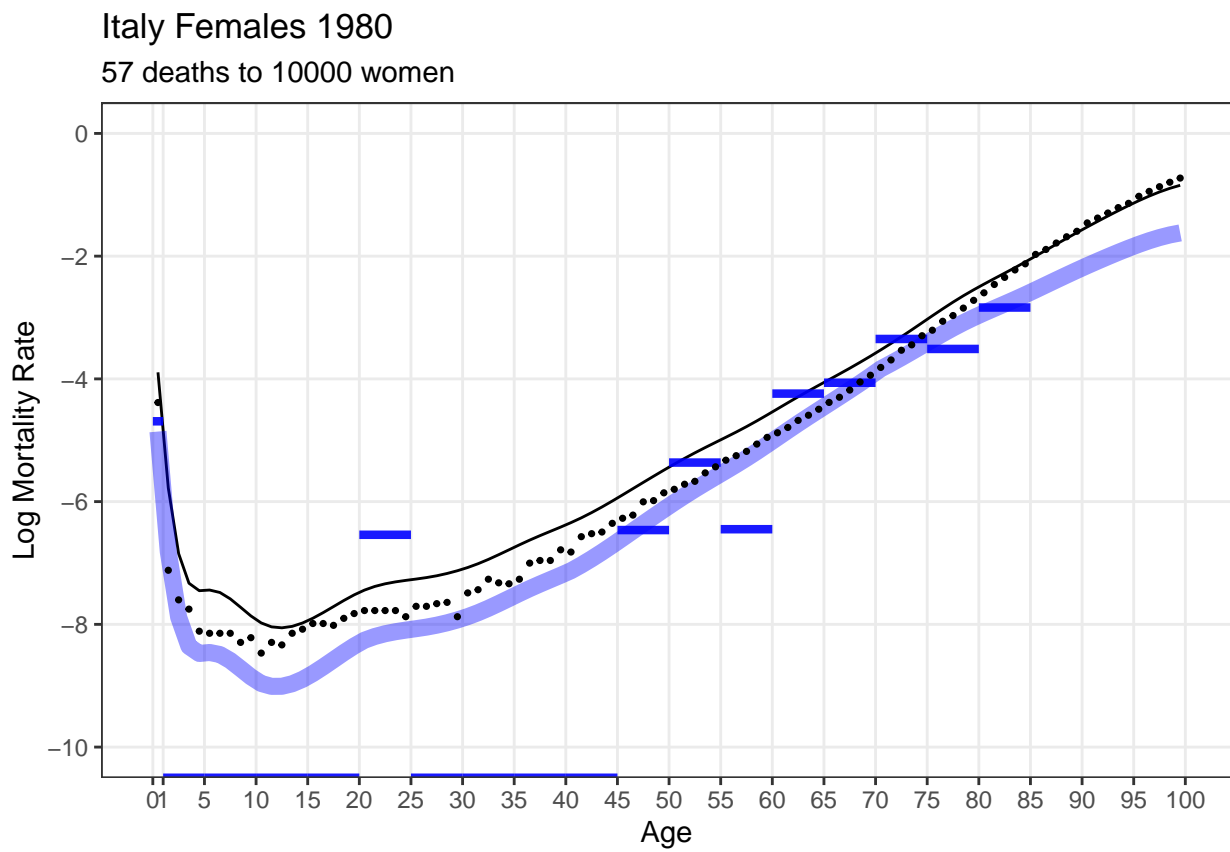
In this figure the dots are the true rates from the HMD, the horizontal bars are the $\log(N_g/D_g)$ values from the grouped data, the black curve is the standard, and the red curve is the TOPALS single-year fit from PIRLS.

Example: Very small populations with Italy 1980 rates

Repeat the exercise in the last section but with simulated deaths for a much smaller population subject to the same rates. Suppose there were only 10000 Italian women and simulate deaths:

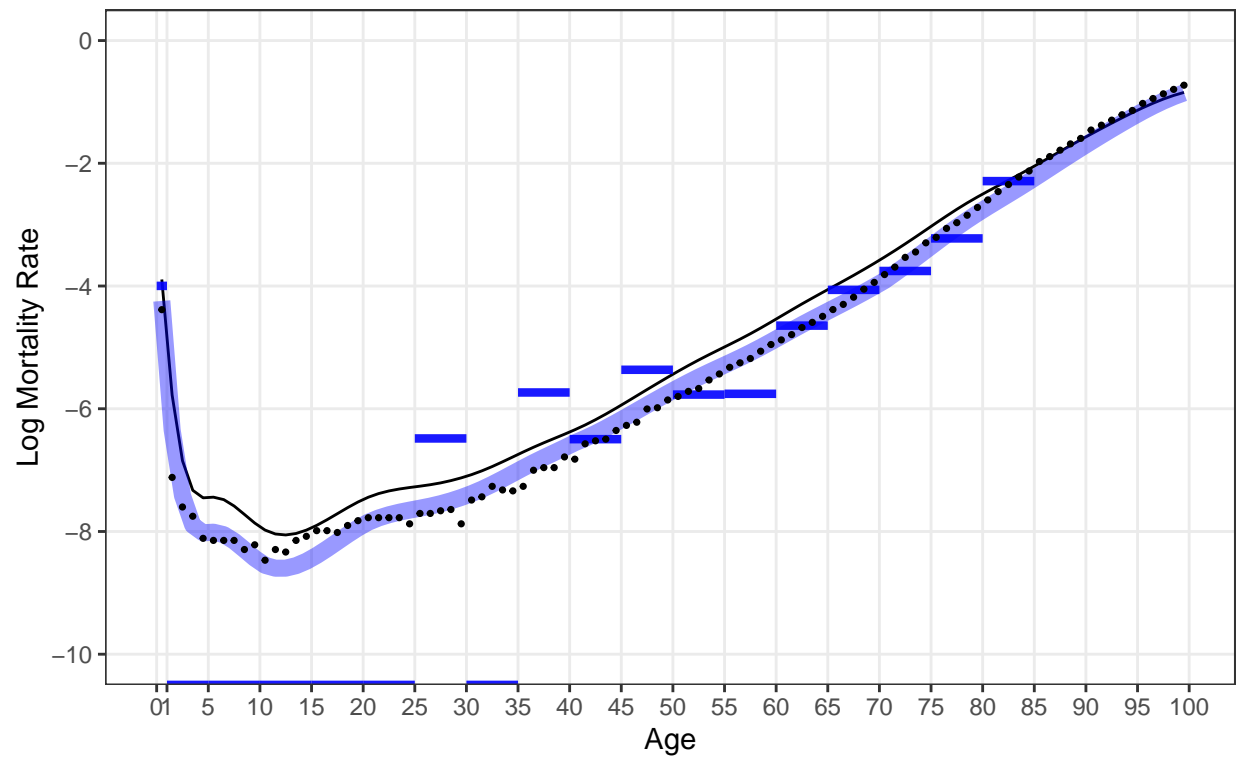
```
for (i in 1:4) {  
  ## simulate a small population  
  target_pop = 10000  
  smallN = N * target_pop/sum(N)  
  smallD = rpois(length(smallN), smallN* D/N)  
  
  this_fit = TOPALS_fit(smallN, smallD,std,  
                        age_group_bounds = boundaries,  
                        detail=TRUE)
```

```
show(this_fit, hue='blue')
}
```



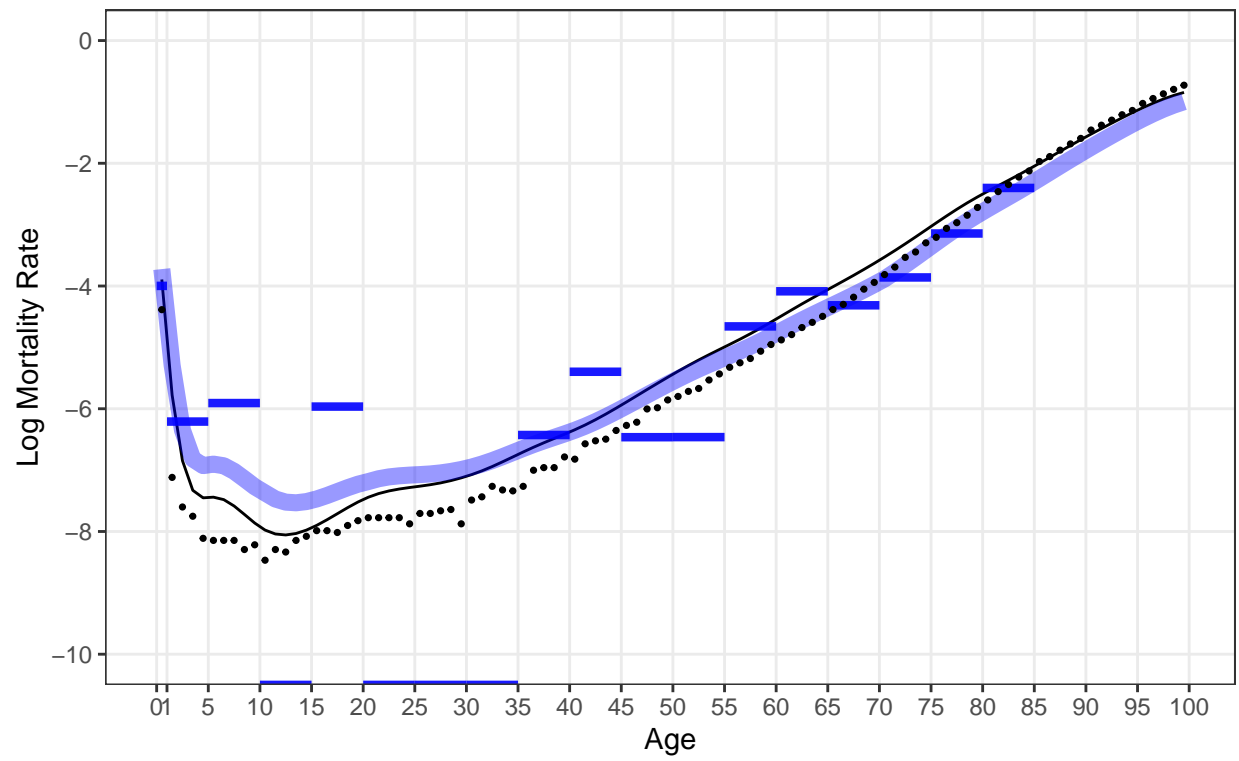
Italy Females 1980

67 deaths to 10000 women



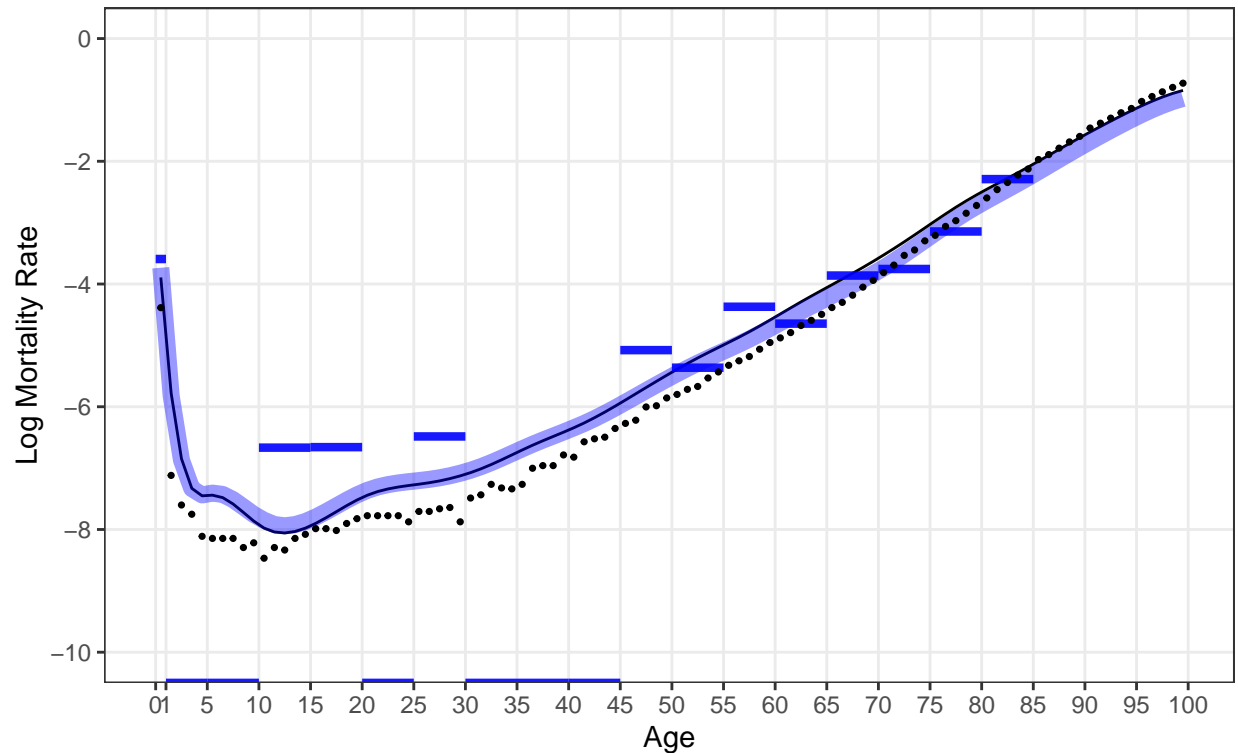
Italy Females 1980

72 deaths to 10000 women



Italy Females 1980

78 deaths to 10000 women

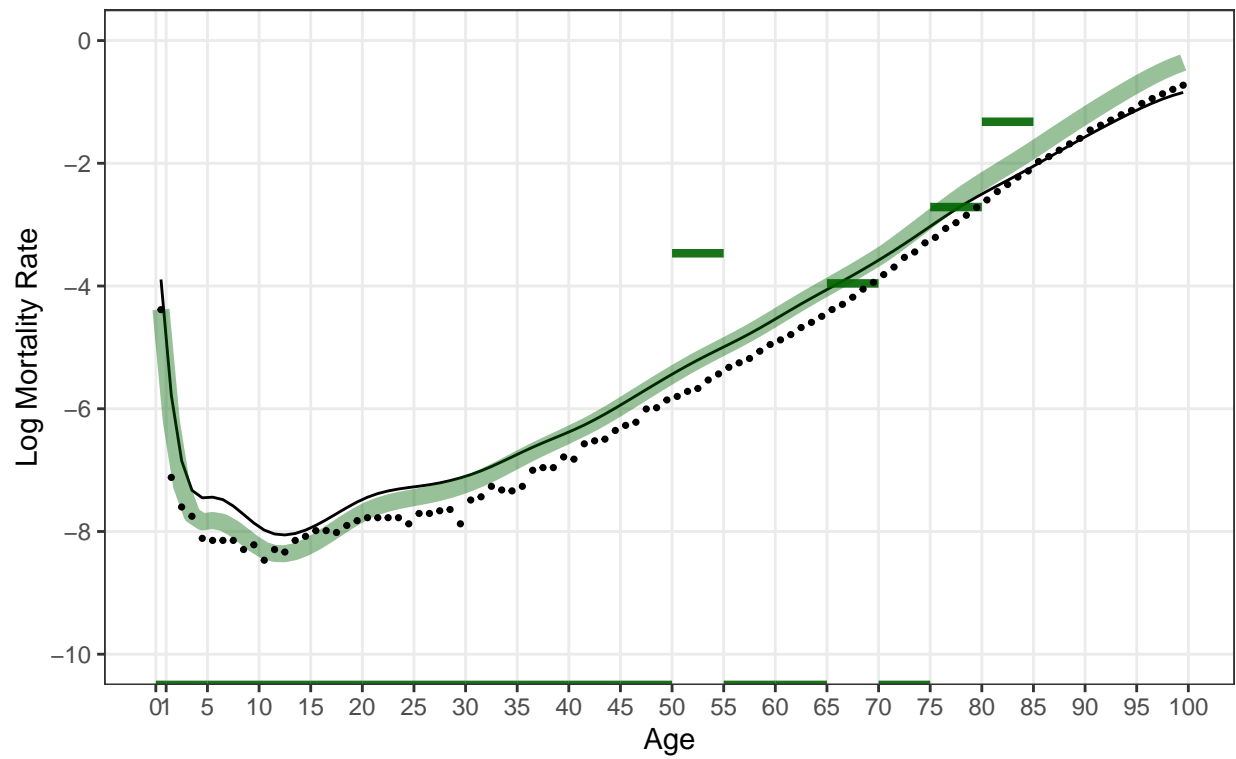


Example: Super-small populations with Italy 1980 rates

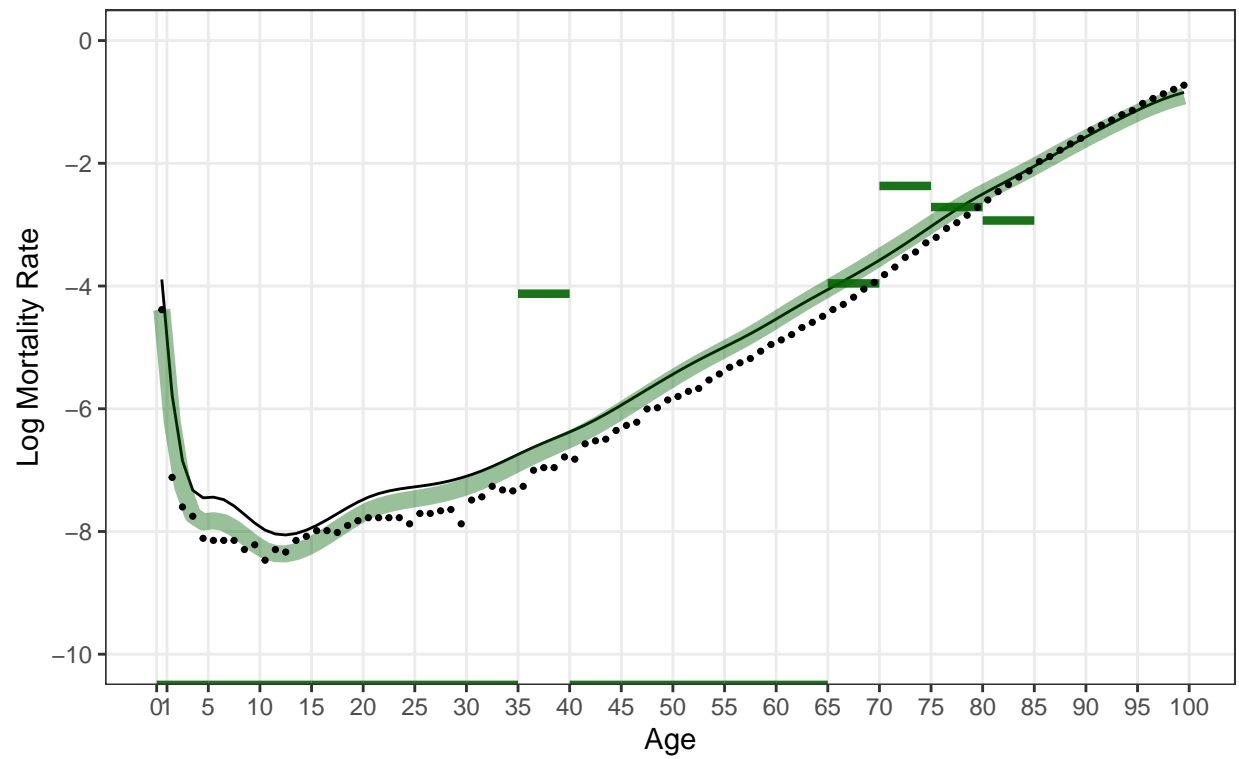
Repeat the exercise in the last section but with simulated deaths for a much smaller population subject to the same rates. Suppose there were only 1000 Italian women and simulate deaths:

```
for (i in 1:4) {  
  ## simulate a small population  
  target_pop = 1000  
  smallN = N * target_pop/sum(N)  
  smallD = rpois(length(smallN), smallN* D/N)  
  
  this_fit = TOPALS_fit(smallN, smallD,std,  
                        age_group_bounds = boundaries,  
                        detail=TRUE)  
  
  show(this_fit, hue='darkgreen')  
}
```

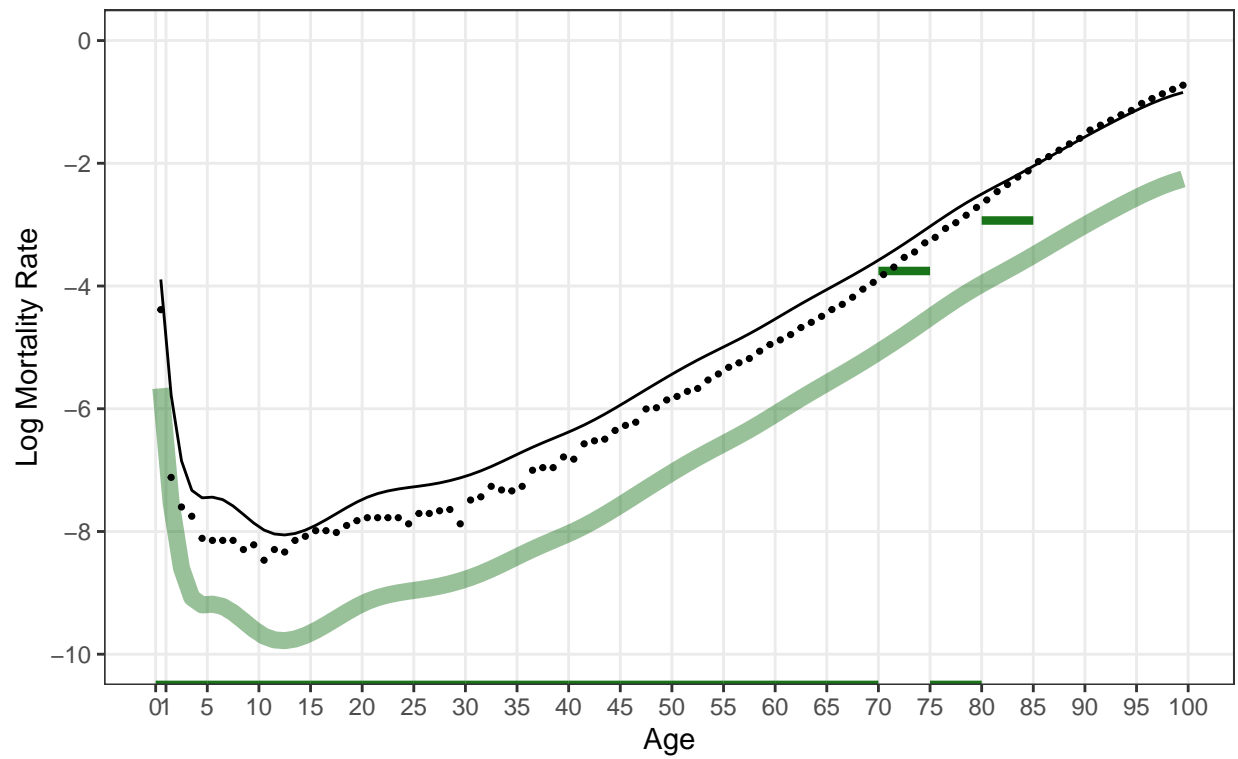
Italy Females 1980
10 deaths to 1000 women



Italy Females 1980
9 deaths to 1000 women



Italy Females 1980
2 deaths to 1000 women



Italy Females 1980
4 deaths to 1000 women

