PIRLS-test.R

Carl Schmertmann

Tue Aug 13 13:34:43 2019

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.5.3
```

```
## -- Attaching packages ------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0       v purrr   0.3.0
## v tibble  2.0.1       v dplyr   0.8.0.1
## v tidyr   0.8.2       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.3.0
```

```
## Warning: package 'tibble' was built under R version 3.5.2
```

```
## Warning: package 'readr' was built under R version 3.5.2
```

```
## Warning: package 'purrr' was built under R version 3.5.2
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
## Warning: package 'stringr' was built under R version 3.5.2
```

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

```r
library(broom)
```

```
## Warning: package 'broom' was built under R version 3.5.3
```

```r
rm(list=ls())

#============== DATA ===========================
## Italy 1980 Female data from HMD (true e0 from HMD is 77.42)

ITA = read.csv(file='ITA-Female-1980.csv')

# standard schedule = smoothed CAN females 1959 log rates at 0,1,...99
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, -7.9004, -7.8088, -7.7101, -7.6113, -7.5195, -7.4415,
       -7.3823, -7.3393, -7.308, -7.2837, -7.2619, -7.238, -7.2082,
       -7.1711, -7.1264, -7.0735, -7.0118, -6.9414, -6.8648, -6.7849,
       -6.7047, -6.6272, -6.5544, -6.4845, -6.4147, -6.3423, -6.2644,
       -6.1791, -6.0872, -5.9904, -5.8903, -5.7887, -5.6869, -5.586,
       -5.4866, -5.3895, -5.2953, -5.205, -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, -3.7297, -3.6303,
       -3.5279, -3.4221, -3.3129, -3.2004, -3.0861, -2.9716, -2.8589,
       -2.7497, -2.6457, -2.5482, -2.4556, -2.3659, -2.2771, -2.187,
       -2.0942, -1.9991, -1.9027, -1.8062, -1.7105, -1.6164, -1.5242,
       -1.434, -1.3458, -1.2596, -1.1758, -1.0958, -1.0212, -0.9535,
       -0.8944, -0.8455)
```

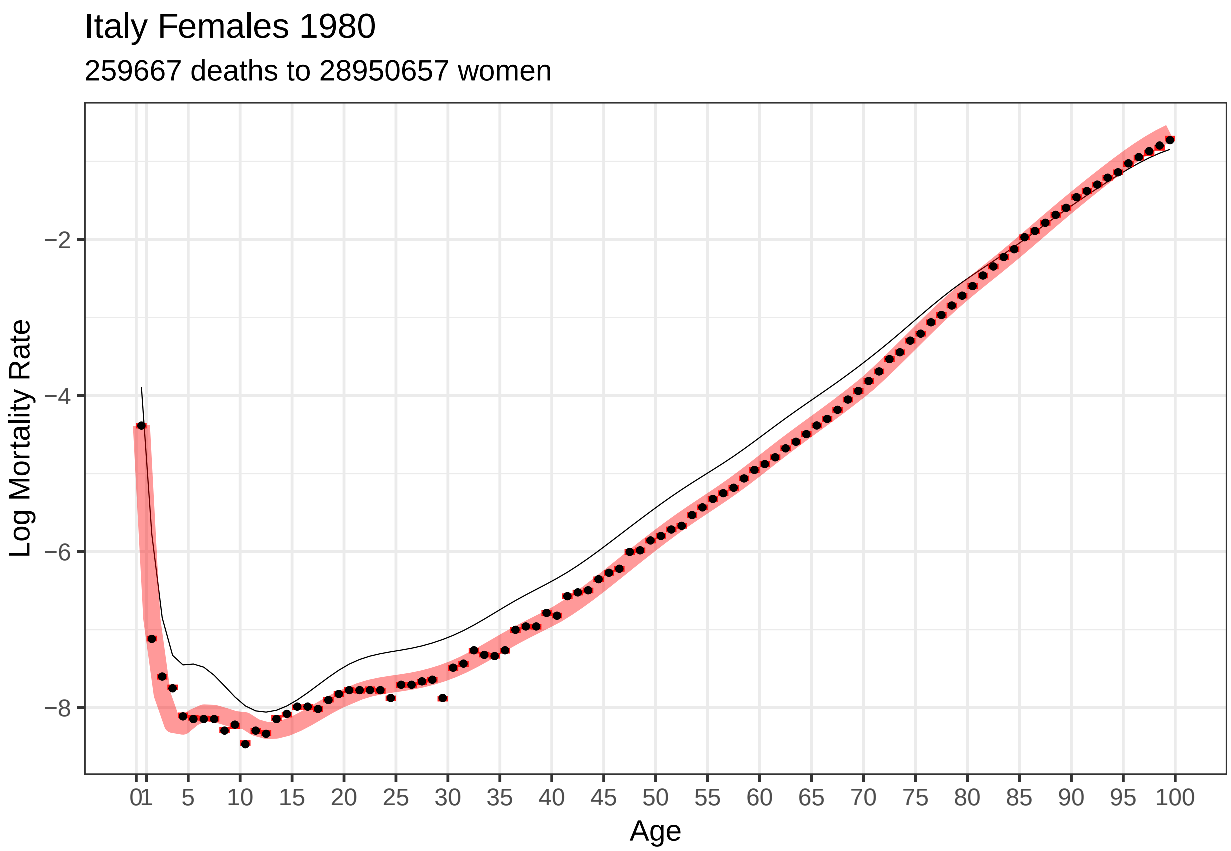
```
##################################################

# note that this sources TOPALS_fit.R (the grouped version)
# rather than TOPALS_fit function.R (the single-year version)

source('TOPALS_fit.R')
```

```
# some utility functions
```

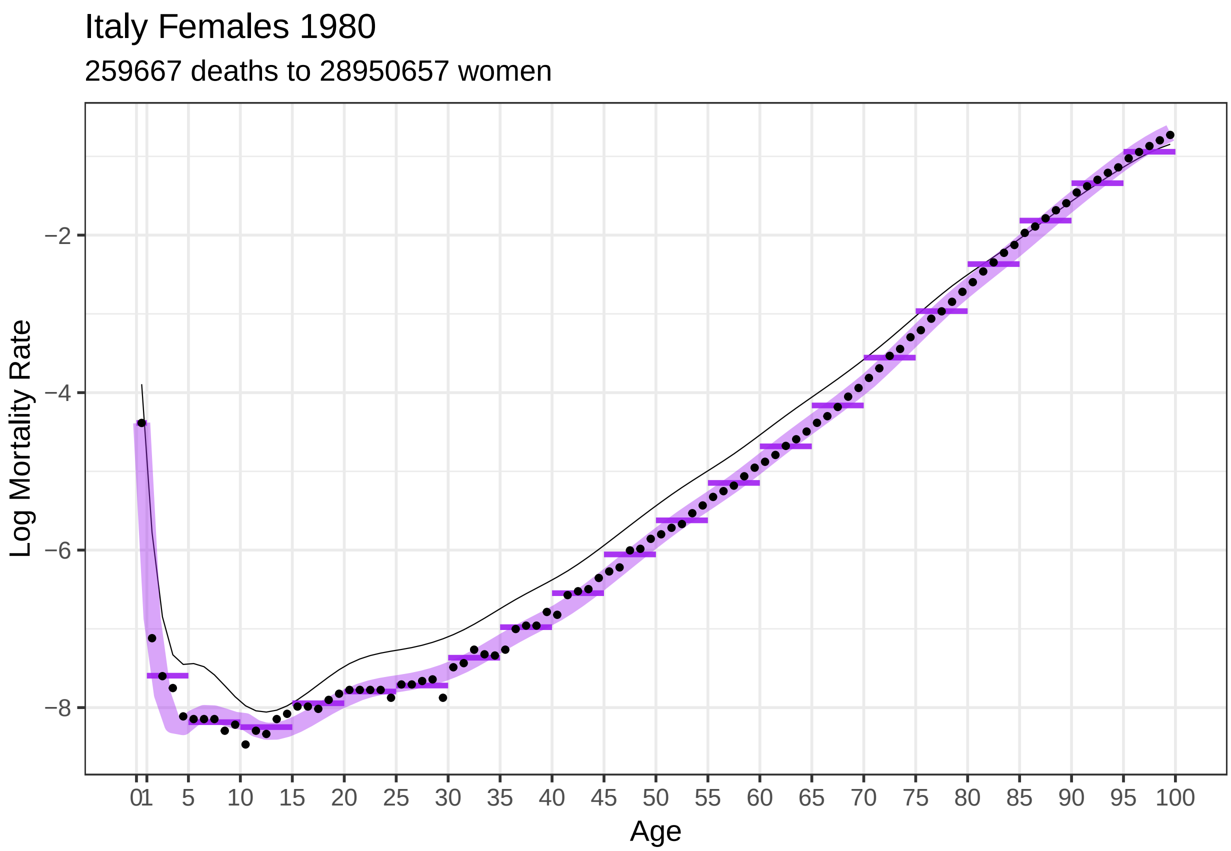
```
## age-grouping function
agg = function(x,bounds) {
  age = seq(x)-1  # 0,1,2,...
  L = head(bounds,-1)
  U = tail(bounds,-1)
  as.vector( tapply( x, cut(age, breaks=bounds, right=FALSE), sum))
}


## plotting function
show_fit = function(fit, true_schedule, fit_color='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(std)-0.5,
    std        = myfit$std,
    logmx_true = true_schedule,
    logmx_fit  = myfit$logm
  )

  this_plot =
    ggplot(data = df_single, aes(x=age,y=logmx_true)) +
    geom_line(aes(x=age,y=std), color='black', lwd=0.2) +
    geom_line(aes(x=age,y=logmx_fit), color=fit_color, lwd=3, alpha=.40) +
    geom_segment(data=df_grouped,
                 aes(x=L,xend=U,y=logmx_obs,yend=logmx_obs),
                 color=fit_color,lwd=1, alpha=.90) +
    geom_point(size=0.80) +
    labs(x='Age',y='Log Mortality Rate',
         title='Italy Females 1980',
         subtitle = paste(sum(D),'deaths to',round(sum(N)),'women')) +
    scale_x_continuous(breaks=c(0,1,seq(5,100,5)),minor_breaks = NULL) +
    theme_bw()

  print(this_plot)
} # show_fit

# trapez approx of life expectancy from a logmx schedule over ages 0..99
e0 = function(logmx) {
  mx = exp(logmx)
  px = exp(-mx)
  lx = c(1,cumprod(px))
  return( sum(head(lx,-1) + tail(lx,-1)) / 2)
}

#======================================
# FULL DATASET WITH 1-YEAR GROUPS
#======================================

## full dataset: 0,1,...99
N = ITA$N[1:100]
D = ITA$D[1:100]

bounds = 0:100

myfit = TOPALS_fit(D=D, N=N, std=std,
                   age_group_bounds = bounds,
                   details=TRUE)
```

```
## Loading required package: splines
```

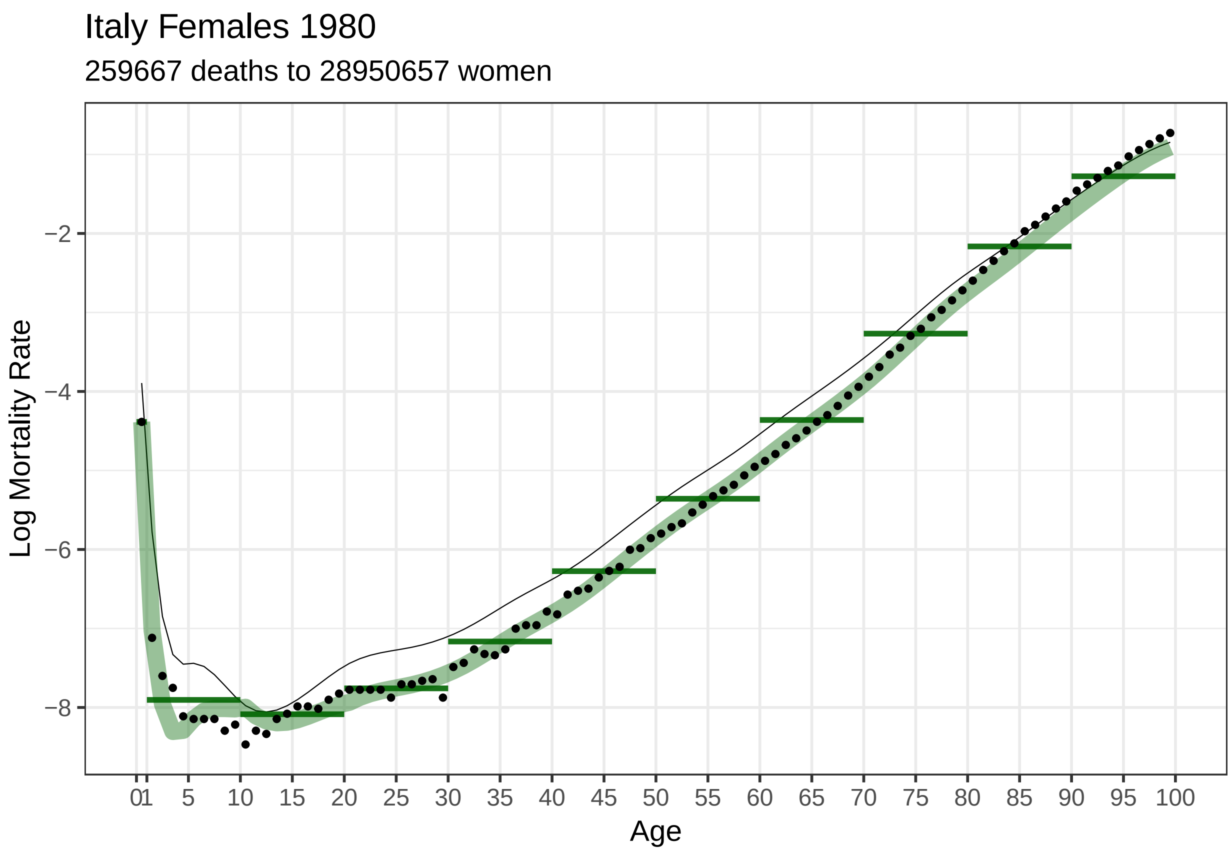
```
show_fit( myfit,  true_schedule = ITA$logmx[1:100],
          fit_color = 'red')
```



```
#========================================
# FULL DATASET WITH 5-YEAR GROUPS
#========================================

bounds = c(0,1,seq(5,100,5))

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

myfit = TOPALS_fit(D=D, N=N, std=std,
                   age_group_bounds = bounds,
                   details=TRUE)

show_fit( myfit,  true_schedule = ITA$logmx[1:100],
          fit_color = 'purple')
```



```
#========================================
# FULL DATASET WITH 10-YEAR GROUPS
#========================================

bounds = c(0,1, seq(10,100,10))

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

myfit = TOPALS_fit(D=D, N=N, std=std,
                   age_group_bounds = bounds,
                   details=TRUE)

show_fit( myfit,  true_schedule = ITA$logmx[1:100],
          fit_color = 'darkgreen')
```


Italy Females 1980

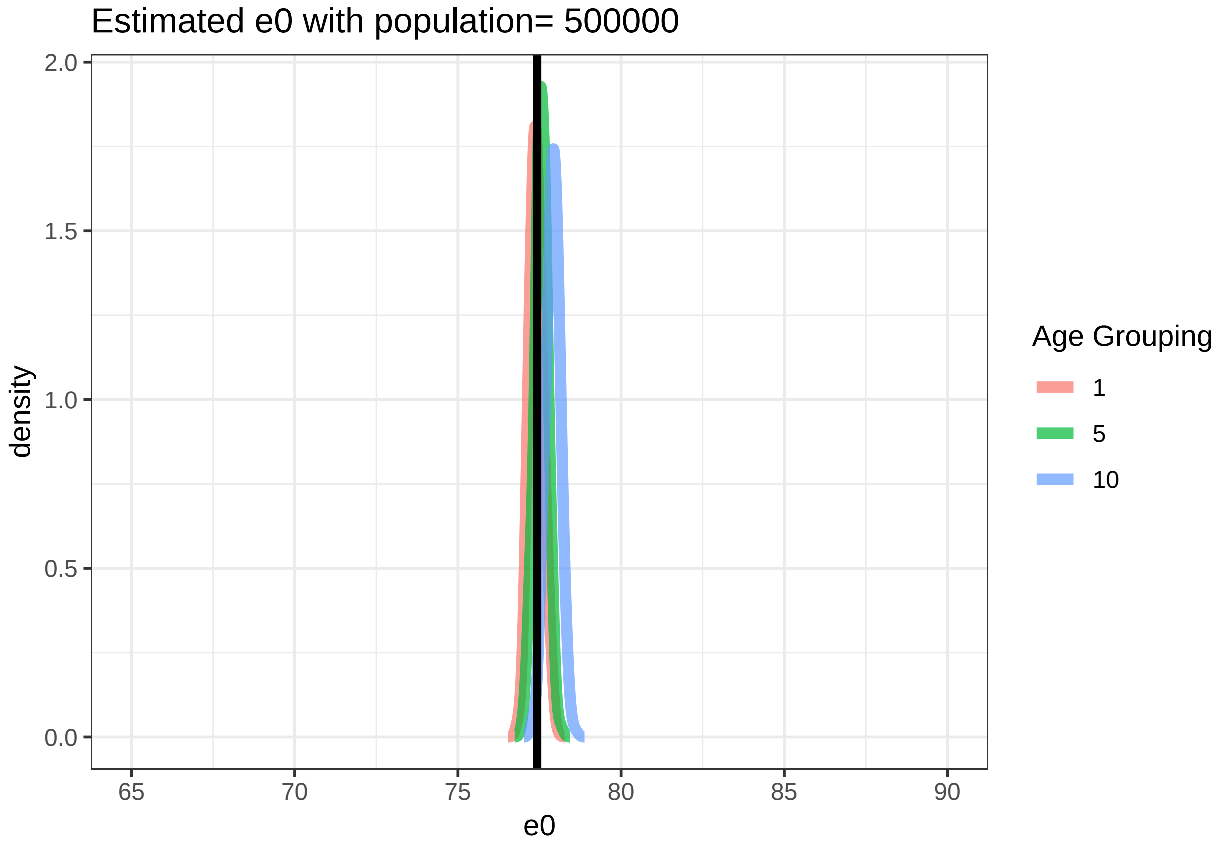
259667 deaths to 28950657 women

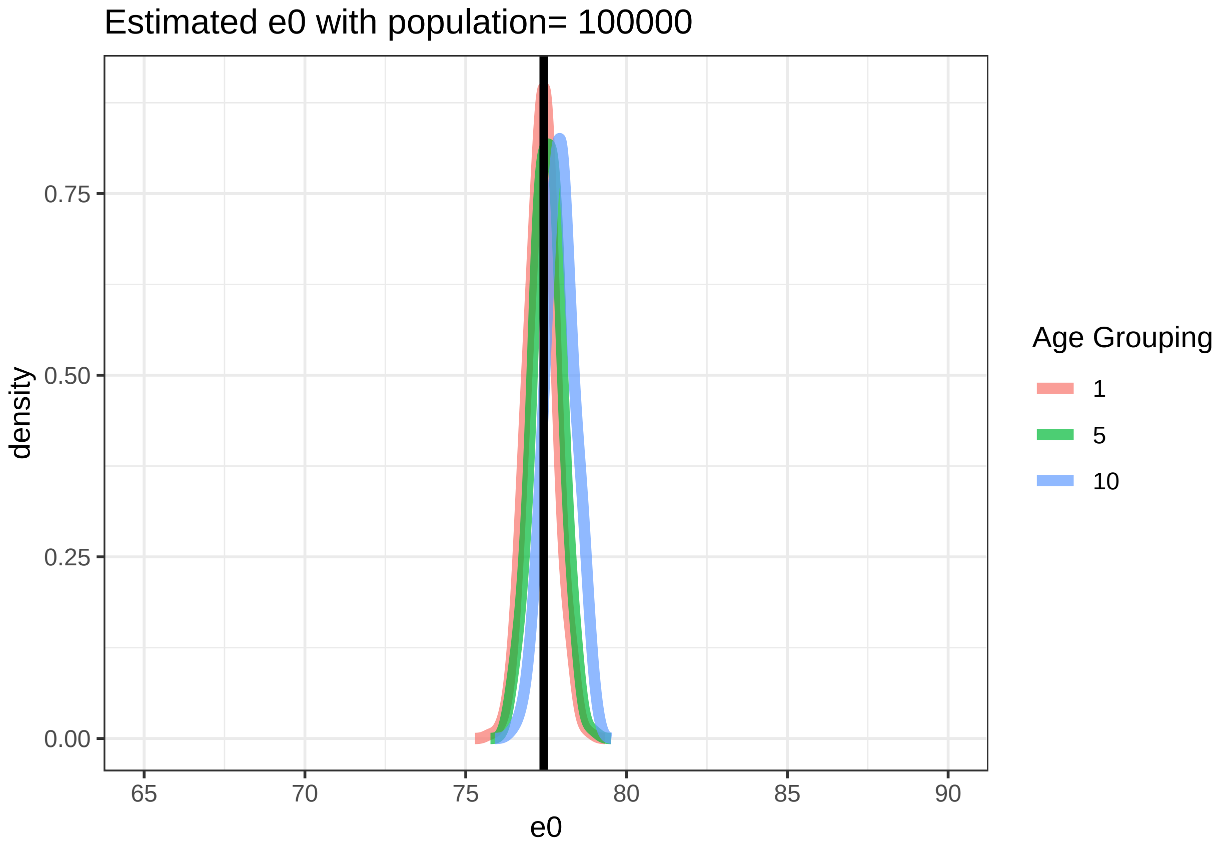
```
#  SMALL POPULATION SIMULATIONS
```

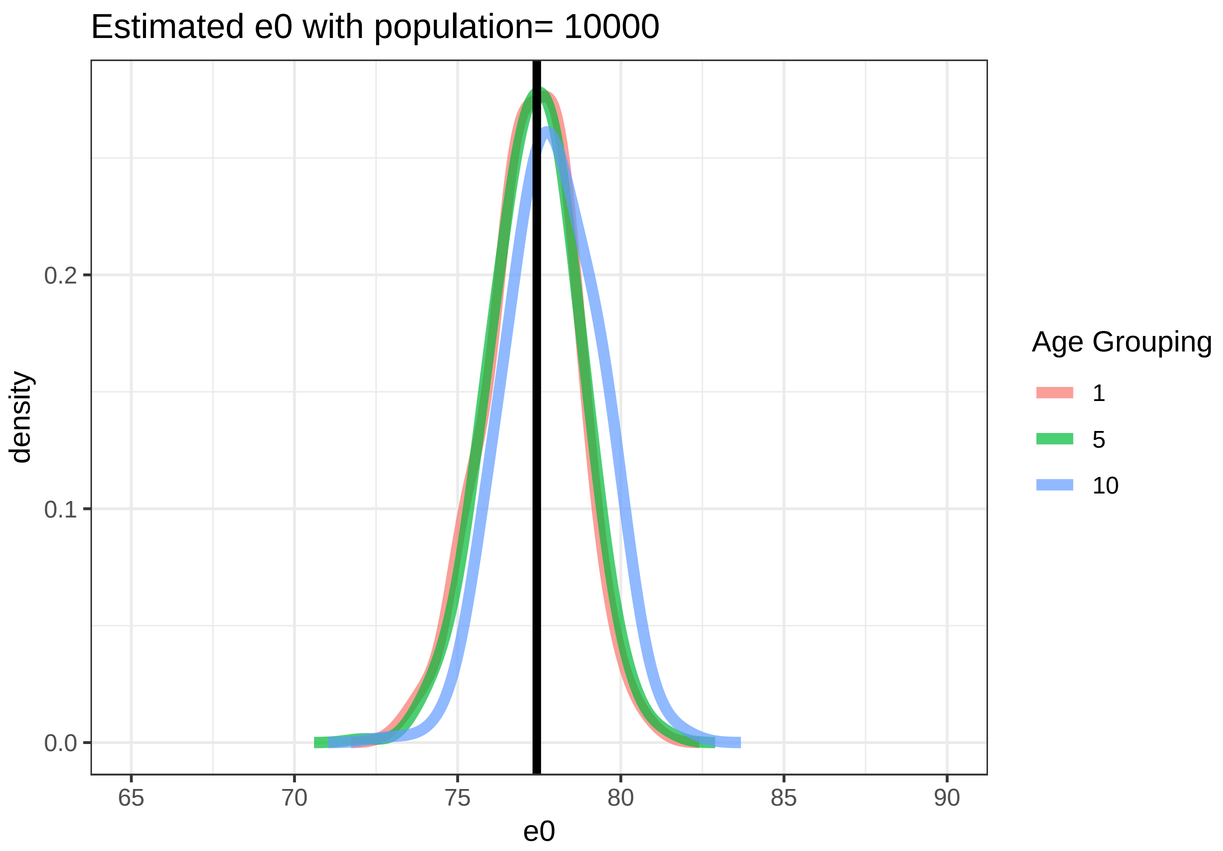
```
# scenario is a data frame with experimental parameters
# target_pop, L, U, nsim on each row

nsim = 500

pop_vals = c(5e5, 1e5, 1e4, 1e3)

bnd_vals = list( seq(0,100,1),
                 c(0,1,seq(5,100,5)),
                 c(0,1,seq(10,100,10))
           )


## MAE and density variables for each scenario will be calculated below
scenario = expand.grid( target_pop = pop_vals,
                        bounds     = bnd_vals,
                        MAE        = Inf) %>%
            as_tibble()

## add an empty LIST column to hold densities
scenario = scenario %>%
            add_column(e0_dens = list(NA))


for (s in 1:nrow(scenario)) {
  target_pop = scenario$target_pop[s]
  bounds     = unlist( scenario$bounds[s] )

  ## exposure and deaths for these age groups (all of Italy)
  bigN = agg(ITA$N, bounds)
  bigD = agg(ITA$D, bounds)

  e    = rep(NA,nsim)

  # small population with same age structure as ITA
  N          = bigN * target_pop/sum(bigN)

  for (i in 1:nsim) {

    # random deaths for this small popualtion at Italian rates
    D = rpois(length(N), N * bigD/bigN)

    myfit = TOPALS_fit(D=D, N=N, std=std,
                       age_group_bounds = bounds,
                       details=TRUE)

    e[i] = e0(myfit$logm)
  } # for i

  scenario$MAE[s]     = round(mean( abs(e-77.42)),2)
  scenario$e0_dens[s] = list( tidy(density(e, adj=1.5) ))

} # for s


# MAE report
matrix( round(scenario$MAE,2), nrow=4,
        dimnames=list(paste('Pop=',format(pop_vals,scientific = FALSE)),
                      paste0(c('1','5','10'),'-yr grp')))
```

```
##              1-yr grp 5-yr grp 10-yr grp
## Pop= 500000     0.17     0.19      0.50
## Pop= 100000     0.35     0.37      0.57
## Pop=  10000     1.08     1.08      1.19
## Pop=   1000     3.23     3.38      3.31
```

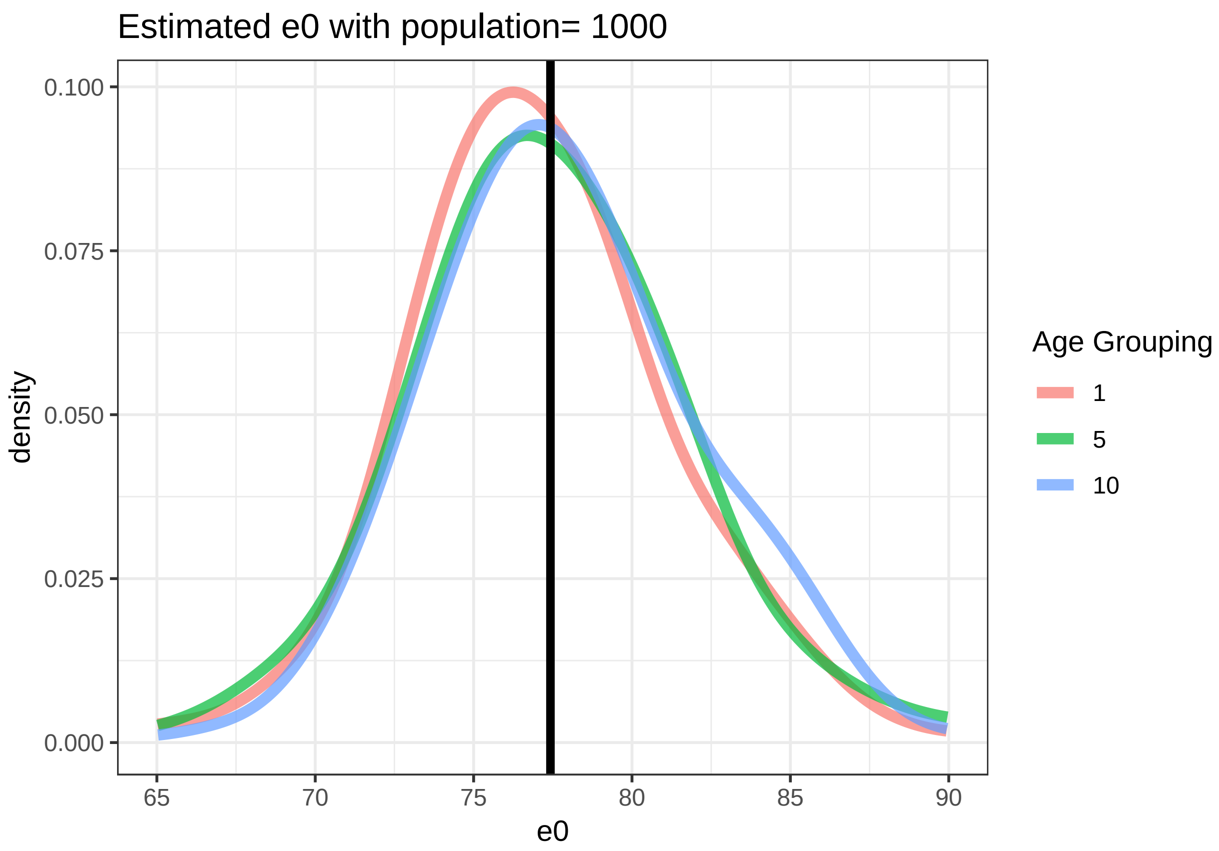
```
## e0 densities

for (p in unique(scenario$target_pop)) {

  tmp = filter( scenario, target_pop==p)

  df1 = as.data.frame(tmp$e0_dens[1]) %>%
          add_column(grouping=1)
  df2 = as.data.frame(tmp$e0_dens[2]) %>%
          add_column(grouping=5)
  df3 = as.data.frame(tmp$e0_dens[3]) %>%
          add_column(grouping=10)
  df = rbind(df1,df2,df3)

  this_plot =
    ggplot(data=df, aes(x=x,y=y, color=as.factor(grouping))) +
      geom_line(lwd=2, alpha=.70) +
      labs(title=paste('Estimated e0 with population=',
                       format(p,scientific = FALSE)),
           x='e0',y='density', color='Age Grouping') +
      geom_vline(xintercept = e0(ITA$logmx), lwd=1.5) +
      scale_x_continuous(limits=c(65,90)) +
      theme_bw()

  print(this_plot)

}
```




Estimated e0 with population= 100000
density
0.00
0.25
0.50
0.75
65
70
75
80
85
90
e0
Age Grouping
1
5
10



```
## Warning: Removed 543 rows containing missing values (geom_path).
```


Estimated e0 with population= 1000
density
0.100
0.075
0.050
0.025
0.000
65
70
75
80
85
90
e0
Age Grouping
1
5
10