# Day 3, exercsie 1: Age-specific decomposition of a difference between life expectancies and lifespan variations

## 2022-05-07

The aim of this exercise is to apply the linear integral model proposed by Horiuchi et al (2008). The example is applied to life expectancy and lifespan variation for Mexican males. The aim is to quantify the contribution of age-specific mortality changes to differences in both these measures between 2005 and 2015.

The dataset contains lifetable and cause of death data for Mexican men over age 15 for the years 2005 and 2015. Columns 1:9 contain data for ten exhaustive and exclusive, assumed to be independent, cause-of-death groups. However, in this exercise we will focus on all-cause mortality. The data and analyses come from Aburto and Beltrán-Sánchez (2019).

First load the data and the necessary packages. We also load a separate script, which contains some functions and information we will need later on.

```
load('MexicoMen.RData')

library(DemoDecomp)
library(tidyverse)

source("Functions_D3.R")
```

Following the method, we require two vectors of rates: one for 2005 and one for 2015.

```
# first we need to vectors of mortality rates
mx1 <- data %>%
  filter(year==2005) %>%
  pull(mx)
mx2 <- data %>%
  filter(year==2015) %>%
  pull(mx)
```

Luckily for us Tim Riffe released a package in CRAN (DemoDecomp), which contains a number of useful decomposition functions. You can check the algorithm for the continuous change method by calling the right function.

```
#load some functions and some info for graphs
horiuchi
```

```
## function (func, pars1, pars2, N, ...)
## {
##     y1 <- func(pars1, ...)
##     y2 <- func(pars2, ...)
##     d <- pars2 - pars1
##     n <- length(pars1)
##     delta <- d/N
##     x <- pars1 + d * matrix(rep(0.5:(N - 0.5)/N, n), byrow = TRUE,
##         ncol = N)
##     cc <- matrix(0, nrow = n, ncol = N)
##     zeros <- rep(0, n)
```

```
##      for (j in 1:N) {
##          DD <- diag(delta/2)
##          for (i in 1:n) {
##              cc[i, j] <- func((x[, j] + DD[, i]), ...) - func((x[,
##                  j] - DD[, i]), ...)
##          }
##      }
##      return(rowSums(cc))
## }
## <bytecode: 0x0000017ef1216778>
## <environment: namespace:DemoDecomp>
```

In order to use this function, we need to define another function, which estimates the summary indicator from a vector of covariates (age-cause specific rates in our case). Let us first start with life expectancy. Here is the needed function.

```
e0.frommx
```

```
## function (nmx = mx, sex = 1, nax = NULL)
## {
##      age <- 0:(length(nmx) - 1)
##      n <- c(diff(age), 999)
##      if (is.null(nax)) {
##          nax <- 0.5 * n
##          if (n[2] == 4) {
##              if (sex == 1) {
##                  if (nmx[1] >= 0.107) {
##                      nax[1] <- 0.33
##                      nax[2] <- 1.352
##                  }
##                  else {
##                      nax[1] <- 0.045 + 2.684 * nmx[1]
##                      nax[2] <- 1.651 - 2.816 * nmx[1]
##                  }
##              }
##              if (sex == 2) {
##                  if (nmx[1] >= 0.107) {
##                      nax[1] <- 0.35
##                      nax[2] <- 1.361
##                  }
##                  else {
##                      nax[1] <- 0.053 + 2.8 * nmx[1]
##                      nax[2] <- 1.522 - 1.518 * nmx[1]
##                  }
##              }
##          }
##      }
##      nqx <- (n * nmx)/(1 + (n - nax) * nmx)
##      nqx <- c(nqx[-(length(nqx))], 1)
##      nqx[nqx > 1] <- 1
##      npx <- 1 - nqx
##      lx <- cumprod(c(1, npx))
##      ndx <- -diff(lx)
##      lxpn <- lx[-1]
##      nLxpn <- n * lxpn + ndx * nax
```

```
##       nLx <- c(nLxpn[-length(nLxpn)], lxpn[length(lxpn) - 1]/nmx[length(nmx)])
##       Tx <- rev(cumsum(rev(nLx)))
##       lx <- lx[1:length(age)]
##       ex <- Tx/lx
##       e0 <- ex[1]
##       return(e0)
## }
```

Now we can perform the decomposition following the arguments of the function. We'll get a vector with the age-specific contributions to the change in life expectancy.

```
#Now we can perfom the decomposition
results <- horiuchi(func = e0.frommx, pars1 = mx1, pars2 = mx2, N = 100)
```

Let's have a look at the results.

```
results
```

```
##  [1] 1.198434e-03 1.080514e-03 9.569737e-04 3.244721e-04 1.105501e-03
##  [6] 1.455085e-03 1.708074e-03 2.194065e-03 2.236456e-03 2.760845e-03
## [11] 2.195083e-03 2.618506e-03 2.719373e-03 3.588402e-03 3.164393e-03
## [16] 4.215794e-03 4.284196e-03 4.826602e-03 5.068240e-03 5.870266e-03
## [21] 6.595605e-03 6.425887e-03 7.463080e-03 7.694838e-03 8.537161e-03
## [26] 9.337188e-03 1.016239e-02 1.125144e-02 1.203325e-02 1.262266e-02
## [31] 1.320610e-02 1.364143e-02 1.362174e-02 1.388062e-02 1.414974e-02
## [36] 1.400647e-02 1.394208e-02 1.430703e-02 1.430783e-02 1.467613e-02
## [41] 1.514574e-02 1.477909e-02 1.477408e-02 1.445811e-02 1.403288e-02
## [46] 1.365480e-02 1.300880e-02 1.255246e-02 1.181598e-02 1.108344e-02
## [51] 1.009441e-02 9.608552e-03 8.627848e-03 8.122388e-03 7.182437e-03
## [56] 5.857698e-03 5.385979e-03 5.073094e-03 4.763988e-03 4.857903e-03
## [61] 5.579635e-03 5.193740e-03 4.786113e-03 4.606401e-03 4.638417e-03
## [66] 5.149155e-03 4.153624e-03 3.104129e-03 2.366086e-03 2.344904e-03
## [71] 2.057427e-03 1.857535e-03 1.885238e-03 1.864765e-03 1.557502e-03
## [76] 1.098618e-03 1.306923e-03 8.679445e-04 7.258661e-04 5.860968e-04
## [81] 4.660079e-04 3.646099e-04 2.805289e-04 2.121049e-04 1.574960e-04
## [86] 1.147825e-04 8.206201e-05 5.752941e-05 3.953817e-05 2.664041e-05
## [91] 1.760677e-05 1.142807e-05 7.302920e-06 1.267375e-05
```

Check consistency with the original difference in life expectancy.

```
#original
(original <- e0.frommx(mx2) - e0.frommx(mx1))
```

```
## [1] 0.547894
```

```
#with decomp
(with.decomp <- sum(results))
```

```
## [1] 0.547894
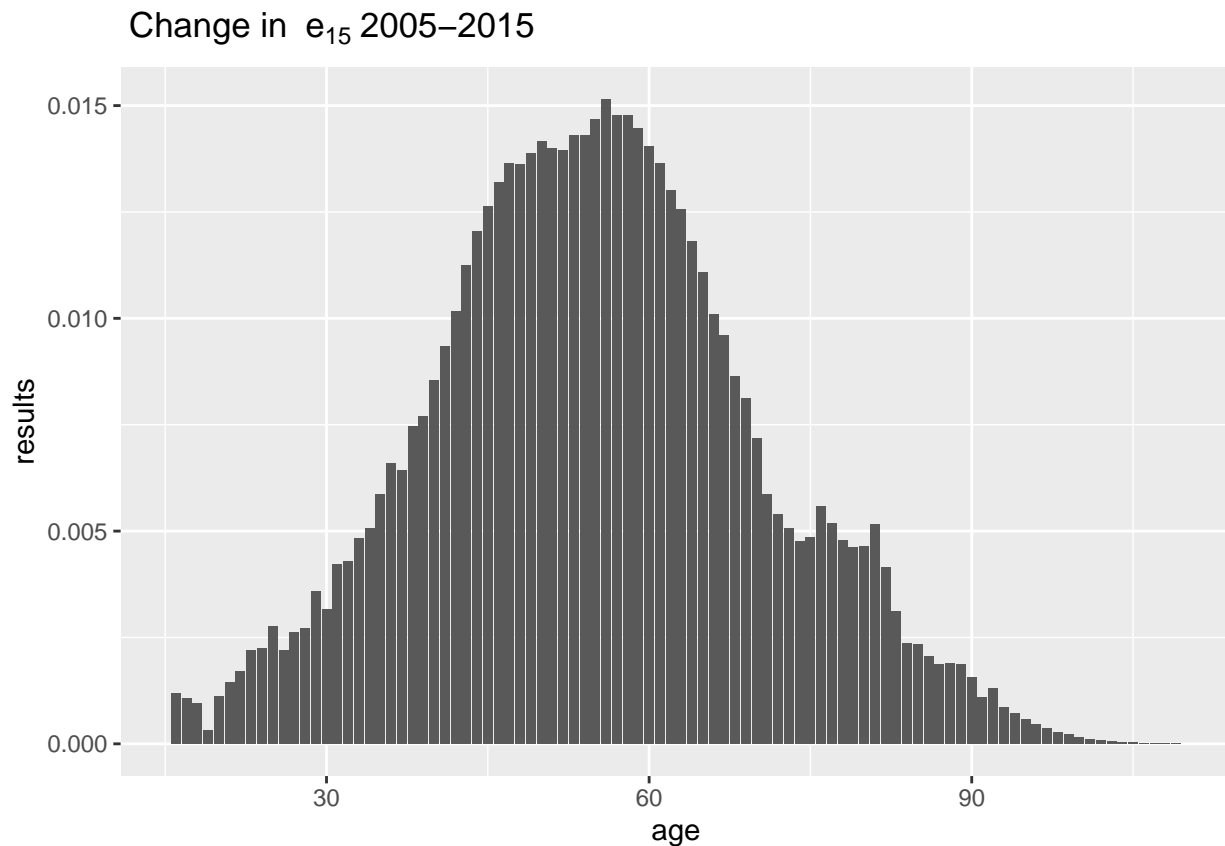```

Just do the difference and you will get the error term.

```
#error
with.decomp - original
```

```
## [1] -1.93009e-09
```

Finally, graph and interpret the results.

```r
#now graph results
age <- data %>%
  filter(year==2005) %>%
  pull(age)

ggplot()+
ggtitle(bquote(~'Change in '~ e[15] ~'2005-2015' ))+
geom_bar(aes(x = age, y= results), stat = "identity", position = "stack")
```



Change in $e_{15}$ 2005–2015

Let's do the same with lifespan disparity

```r
edagger.frommx
```

```
## function (mx, sex = 1)
## {
##     i.openage <- length(mx)
##     OPENAGE <- i.openage - 1
##     RADIX <- 1
##     if (mx[i.openage] < 0.5 | is.na(mx[i.openage]))
##         mx[i.openage] = mx[i.openage - 1] * 1.1
##     ax <- mx * 0 + 0.5
##     ax[1] <- AKm02a0(m0 = mx[1], sex = sex)
##     ax[i.openage] <- if (mx[i.openage] == 0)
##         0.5
##     else 1/mx[i.openage]
##     qx <- mx/(1 + (1 - ax) * mx)
##     qx[i.openage] <- ifelse(is.na(qx[i.openage]), NA, 1)
```

```
##       px <- 1 - qx
##       px[is.nan(px)] <- 0
##       lx <- c(RADIX, RADIX * cumprod(px[1:OPENAGE]))
##       dx <- lx * qx
##       Lx <- lx - (1 - ax) * dx
##       Lx[i.openage] <- dx[i.openage] * ax[i.openage]
##       Lx[is.na(Lx)] <- 0
##       Tx <- c(rev(cumsum(rev(Lx[1:OPENAGE]))), 0) + Lx[i.openage]
##       ex <- Tx/lx
##       ex[is.na(ex)] <- 0
##       ex[i.openage] <- if (ex[OPENAGE] == 0)
##           0
##       else ax[i.openage]
##       v <- (ax * c(ex[-1L], 0) + (1 - ax) * ex)
##       v[length(ex)] <- ex[length(ex)]
##       v <- dx * v
##       e.dagger <- rev(cumsum(rev(v)))/lx
##       e.dagger[16]
## }
```

```r
#Now we can perfom the decomposition
results_edag <- horiuchi(func = edagger.frommx, pars1 = mx1, pars2 = mx2, N = 100)
```

Let's have a look at the results.

```r
results_edag
```

```
##   [1] -1.065814e-14  1.421085e-14 -8.881784e-15  2.664535e-14  1.776357e-14
##   [6]  2.131628e-14  1.065814e-14 -7.105427e-15  7.105427e-15  8.881784e-15
##  [11]  2.131628e-14 -1.776357e-15  2.842171e-14 -2.486900e-14 -1.243450e-14
##  [16] -3.087381e-03 -3.107894e-03 -3.466866e-03 -3.602883e-03 -4.127929e-03
##  [21] -4.585414e-03 -4.414243e-03 -5.062565e-03 -5.150979e-03 -5.635414e-03
##  [26] -6.073067e-03 -6.507217e-03 -7.086131e-03 -7.446334e-03 -7.666263e-03
##  [31] -7.862262e-03 -7.950340e-03 -7.760010e-03 -7.716552e-03 -7.662084e-03
##  [36] -7.372541e-03 -7.117096e-03 -7.064479e-03 -6.813613e-03 -6.717775e-03
##  [41] -6.637951e-03 -6.174122e-03 -5.852433e-03 -5.397185e-03 -4.900099e-03
##  [46] -4.420137e-03 -3.860473e-03 -3.367117e-03 -2.812819e-03 -2.283946e-03
##  [51] -1.737941e-03 -1.308867e-03 -8.461830e-04 -4.677423e-04 -1.047757e-04
##  [56]  1.821711e-04  4.290892e-04  6.662107e-04  8.874265e-04  1.189021e-03
##  [61]  1.712987e-03  1.938702e-03  2.124290e-03  2.390744e-03  2.778744e-03
##  [66]  3.523980e-03  3.220351e-03  2.707709e-03  2.308841e-03  2.547404e-03
##  [71]  2.478077e-03  2.471711e-03  2.762751e-03  3.001371e-03  2.746471e-03
##  [76]  2.117777e-03  2.748403e-03  1.987428e-03  1.806562e-03  1.582852e-03
##  [81]  1.363582e-03  1.154282e-03  9.595396e-04  7.828210e-04  6.263846e-04
##  [86]  4.912879e-04  3.774832e-04  2.839859e-04  2.090914e-04  1.506145e-04
##  [91]  1.061239e-04  7.314978e-05  1.560930e-04  0.000000e+00
```

Check consistency with the original difference in life expectancy.

```r
#original
(original <- edagger.frommx(mx2) - edagger.frommx(mx1))
```

```
## [1] -0.1421836
```

```r
#with decomp
(with.decomp <- sum(results_edag))
```
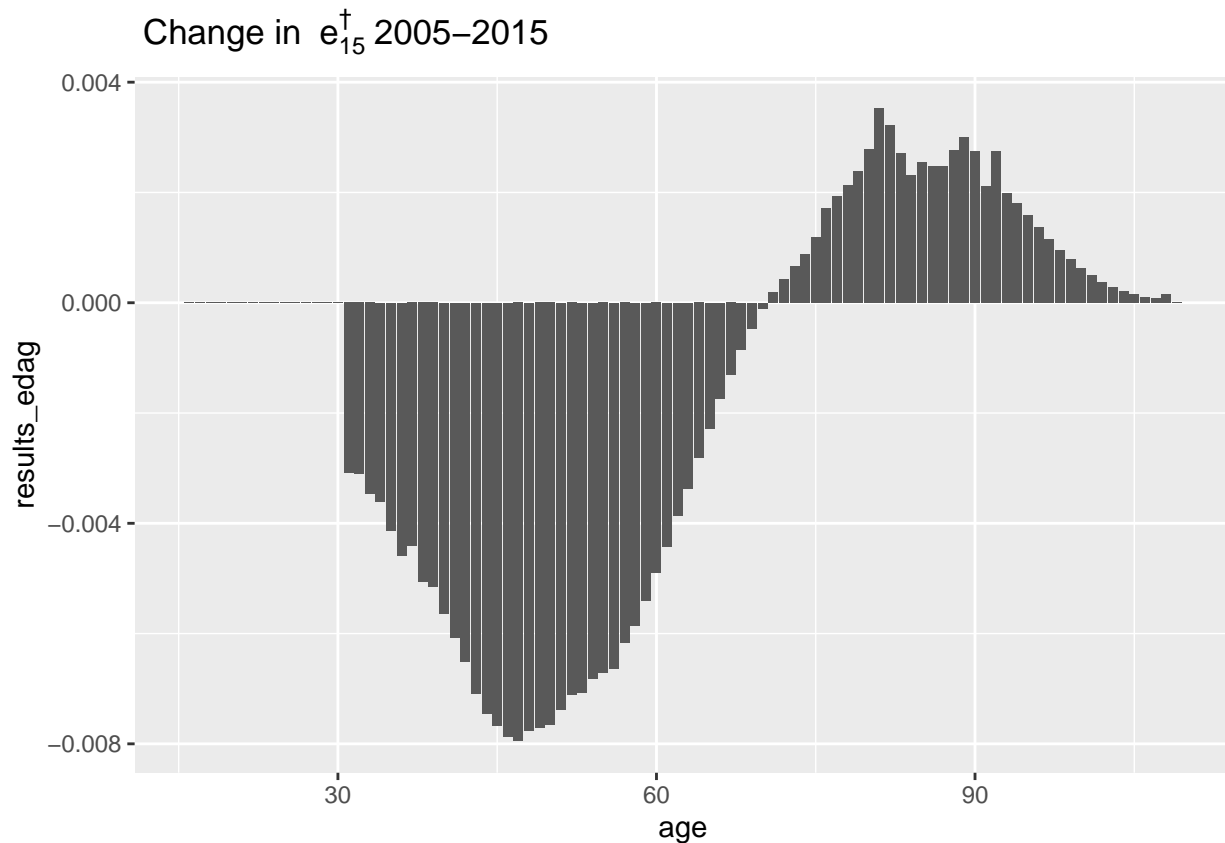
```
## [1] -0.1421836
```

```
#error
with.decomp - original
```

## [1] 2.427917e-09

Finally, graph and interpret the results.

```
#now graph results
ggplot()+
ggtitle(bquote(~'Change in '~ e[15]^"\u2020" ~'2005-2015' )) +
geom_bar(aes(x = age, y= results_edag), stat = "identity", position = "stack")
```



The continuous change method is just one of those that exist. We can obtain very similar results with the stepwise replacement method, also included in the DemoDecomp package. Let's look at the algorithm.

```
stepwise_replacement
```

```
## function (func, pars1, pars2, symmetrical = TRUE, direction = "up",
##     ...)
## {
##     direction <- tolower(direction)
##     stopifnot(direction %in% c("up", "down", "both"))
##     up <- direction %in% c("up", "both")
##     down <- direction %in% c("down", "both")
##     N <- length(pars1)
##     pars1Mat <- matrix(pars1, ncol = N + 1, nrow = N)
##     pars2Mat <- matrix(pars2, ncol = N + 1, nrow = N)
##     RM_1_2_up <- matrix(ncol = N + 1, nrow = N)
```

```
##       RM_1_2_down <- RM_1_2_up
##       RM_2_1_up <- RM_1_2_up
##       RM_2_1_down <- RM_1_2_up
##       r1ind <- lower.tri(pars1Mat, TRUE)
##       r2ind <- upper.tri(pars1Mat)
##       RM_1_2_up[r1ind] <- pars1Mat[r1ind]
##       RM_1_2_up[r2ind] <- pars2Mat[r2ind]
##       RM_1_2_down[r1ind[N:1, ]] <- pars1Mat[r1ind[N:1, ]]
##       RM_1_2_down[r2ind[N:1, ]] <- pars2Mat[r2ind[N:1, ]]
##       RM_2_1_up[r1ind] <- pars2Mat[r1ind]
##       RM_2_1_up[r2ind] <- pars1Mat[r2ind]
##       RM_2_1_down[r1ind[N:1, ]] <- pars2Mat[r1ind[N:1, ]]
##       RM_2_1_down[r2ind[N:1, ]] <- pars1Mat[r2ind[N:1, ]]
##       dec <- matrix(NA, nrow = N, ncol = 4)
##       if (up) {
##           dec[, 1] <- diff(apply(RM_1_2_up, 2, func, ...))
##       }
##       if (down) {
##           dec[, 2] <- diff(apply(RM_1_2_down, 2, func, ...))
##       }
##       if (symmetrical) {
##           if (up) {
##               dec[, 3] <- -diff(apply(RM_2_1_up, 2, func, ...))
##           }
##           if (down) {
##               dec[, 4] <- -diff(apply(RM_2_1_down, 2, func, ...))
##           }
##       }
##       dec_avg <- rowMeans(dec, na.rm = TRUE)
##       dec_avg
## }
## <bytecode: 0x0000017ef063f2e0>
## <environment: namespace:DemoDecomp>
```

Now let's get the results. We can use the same life expectancy and lifespan disparity functions as before.

```
# Life expectancy
results_step <- stepwise_replacement(e0.frommx,pars1 = mx1, pars2 = mx2)

#original
(original <- e0.frommx(mx2) - e0.frommx(mx1))
```

```
## [1] 0.547894
```

```
#with decomp
(with.decomp_step <- sum(results_step))
```
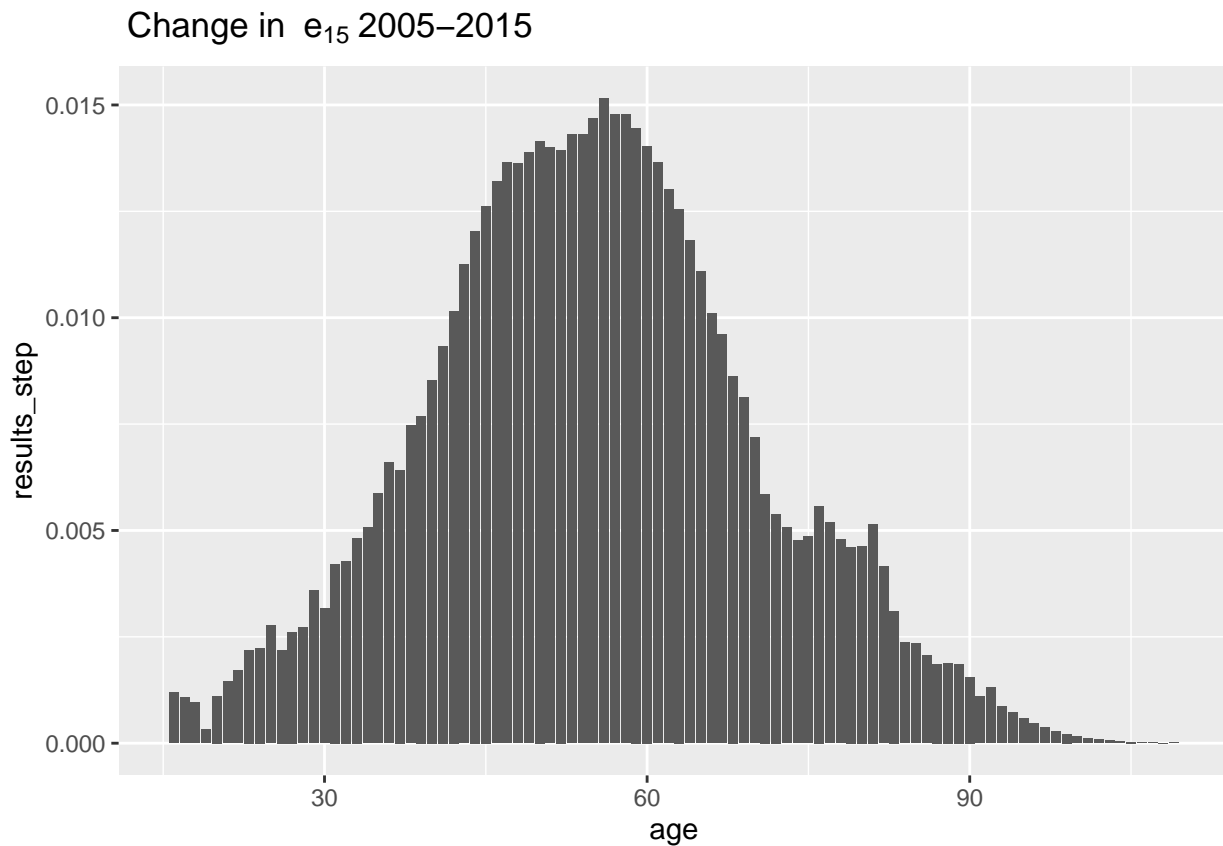
```
## [1] 0.547894
```

```
#error
with.decomp_step - original
```

```
## [1] 0
```

```
#now graph results
ggplot()+
ggtitle(bquote(~'Change in '~ e[15] ~'2005-2015' ))+
```

```
geom_bar(aes(x = age, y= results_step), stat = "identity", position = "stack")
```

## Change in $e_{15}$ 2005–2015



How different are these results from the ones we obtained with the continuous change method?

```
results - results_step
```

```
##  [1] -2.372842e-08 -2.166886e-08 -1.945230e-08 -6.697931e-09 -2.317218e-08
##  [6] -3.089745e-08 -3.669780e-08 -4.761945e-08 -4.897748e-08 -6.092551e-08
## [11] -4.882190e-08 -5.872264e-08 -6.140788e-08 -8.137388e-08 -7.196597e-08
## [16] -9.595846e-08 -9.724317e-08 -1.089907e-07 -1.135018e-07 -1.298109e-07
## [21] -1.430821e-07 -1.362628e-07 -1.538681e-07 -1.530627e-07 -1.625239e-07
## [26] -1.680680e-07 -1.704774e-07 -1.725383e-07 -1.646424e-07 -1.497708e-07
## [31] -1.308234e-07 -1.067658e-07 -7.745063e-08 -4.864986e-08 -1.796140e-08
## [36]  1.369971e-08  4.468338e-08  7.776366e-08  1.094923e-07  1.442438e-07
## [41]  1.808901e-07  2.056640e-07  2.318257e-07  2.491815e-07  2.597368e-07
## [46]  2.661053e-07  2.622207e-07  2.573876e-07  2.426008e-07  2.247054e-07
## [51]  1.999619e-07  1.839441e-07  1.581811e-07  1.416788e-07  1.188770e-07
## [56]  9.436485e-08  8.705182e-08  8.328406e-08  8.032666e-08  8.390568e-08
## [61]  9.350233e-08  7.966790e-08  6.612822e-08  5.510545e-08  4.175268e-08
## [66]  1.556674e-08 -2.049937e-08 -3.423873e-08 -3.351897e-08 -3.868829e-08
## [71] -3.991013e-08 -4.092685e-08 -5.002892e-08 -6.657936e-08 -7.499110e-08
## [76] -6.351226e-08 -9.483680e-08 -8.060768e-08 -7.970094e-08 -7.575560e-08
## [81] -7.029344e-08 -6.368656e-08 -5.633866e-08 -4.865407e-08 -4.100846e-08
## [86] -3.372298e-08 -2.704734e-08 -2.115087e-08 -1.612241e-08 -1.197869e-08
## [91] -8.677826e-09 -6.136094e-09 -4.245113e-09 -8.990447e-09
```

Let's do the same for lifespan disparity

```r
results_step_edag <- stepwise_replacement(edagger.frommx,pars1 = mx1, pars2 = mx2)

#original
(original <- edagger.frommx(mx2) - edagger.frommx(mx1))
```

```
## [1] -0.1421836
```

```r
#with decomp
(with.decomp_step <- sum(results_step_edag))
```

```
## [1] -0.1421836
```
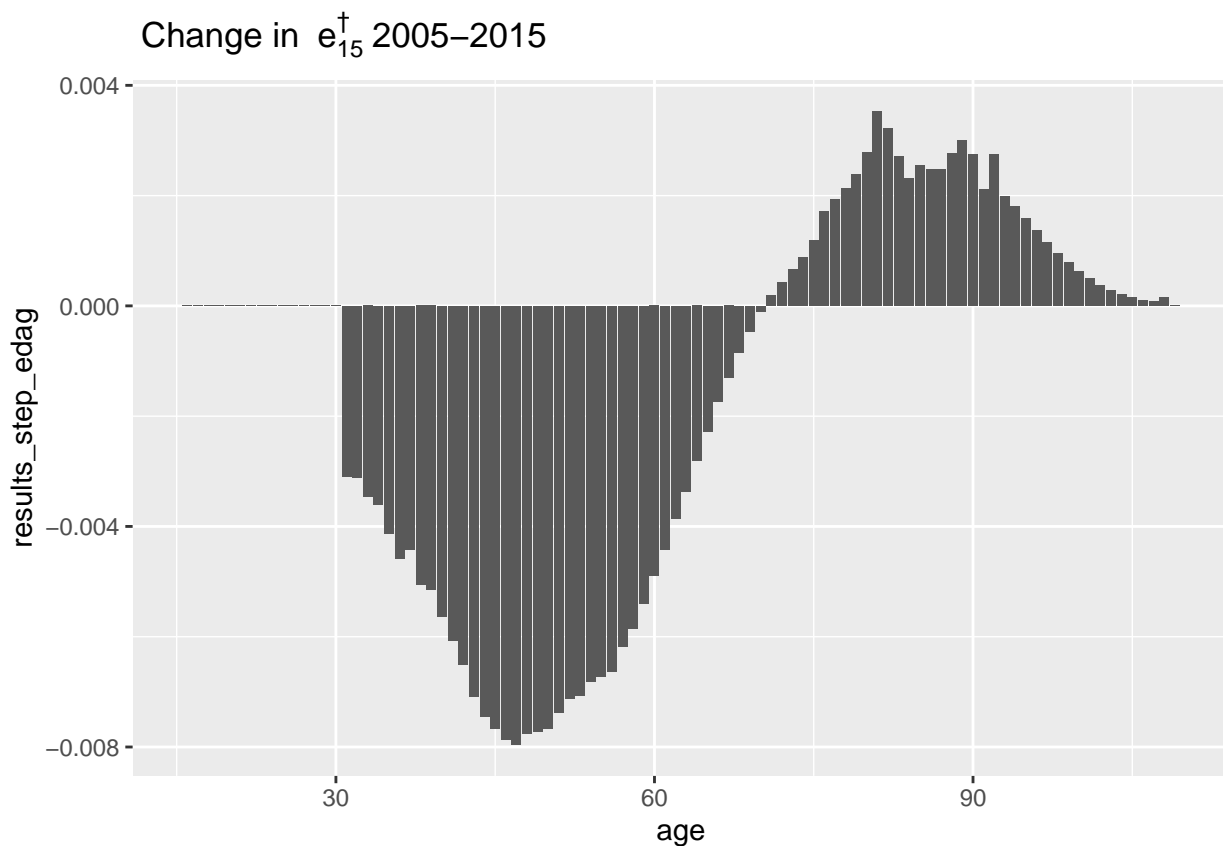
```r
#error
with.decomp_step - original
```

```
## [1] 0
```

```r
#now graph results

ggplot()+
ggtitle(bquote(~'Change in '~ e[15]^"\u2020" ~'2005-2015' ))+
geom_bar(aes(x = age, y= results_step_edag), stat = "identity", position = "stack")
```



Change in $e_{15}^{\dagger}$ 2005–2015

Difference between contiinuous change and stepwise replacement

```r
results_edag - results_step_edag
```

```
##  [1] -8.881784e-15  1.332268e-14 -9.769963e-15  2.753353e-14  1.776357e-14
##  [6]  2.131628e-14  1.154632e-14 -7.993606e-15  6.217249e-15  8.881784e-15
```

```
## [11]  2.220446e-14 -2.664535e-15  2.930989e-14 -2.575717e-14 -1.065814e-14
## [16]  1.970001e-07  1.986370e-07  2.213854e-07  2.290766e-07  2.600404e-07
## [21]  2.840334e-07  2.677528e-07  2.988016e-07  2.930653e-07  3.059734e-07
## [26]  3.097968e-07  3.059029e-07  2.987608e-07  2.714432e-07  2.302183e-07
## [31]  1.804182e-07  1.212787e-07  5.542174e-08 -1.036340e-08 -7.881499e-08
## [36] -1.442660e-07 -2.071814e-07 -2.758094e-07 -3.361805e-07 -4.025727e-07
## [41] -4.693553e-07 -5.025433e-07 -5.372068e-07 -5.489924e-07 -5.441221e-07
## [46] -5.286804e-07 -4.916370e-07 -4.517740e-07 -3.939061e-07 -3.320709e-07
## [51] -2.633506e-07 -2.087758e-07 -1.468852e-07 -9.859416e-08 -5.210486e-08
## [56] -2.028952e-08 -4.203667e-09  8.009838e-09  1.750861e-08  2.930602e-08
## [61]  5.880756e-08  8.806998e-08  1.123579e-07  1.403521e-07  1.818372e-07
## [66]  2.706246e-07  2.830397e-07  2.475331e-07  2.042182e-07  2.138684e-07
## [71]  1.977760e-07  1.854625e-07  1.959858e-07  2.041218e-07  1.767754e-07
## [76]  1.230444e-07  1.393171e-07  8.257971e-08  5.472386e-08  2.689222e-08
## [81]  2.056693e-09 -1.880262e-08 -3.504696e-08 -4.641895e-08 -5.303954e-08
## [86] -5.536326e-08 -5.409919e-08 -5.011027e-08 -4.430698e-08 -3.755028e-08
## [91] -3.057649e-08 -2.395109e-08 -7.190130e-08  0.000000e+00
```