DEMG609: Problem Set 3

Nick Graetz

September 27, 2017

```r
library(knitr)
library(data.table)
library(ggplot2)
library(formatR)

# Load data for problem set
  raw_data <- fread("C:/Users/ngraetz/Documents/repos/demg609/hw3_data_clean.csv")
  raw_data[, x := as.numeric(x)]
  raw_data[, nNx := gsub(',','',nNx)]
  raw_data[, nNx := as.numeric(nNx)]
  raw_data[, nDx := gsub(',','',nDx)]
  raw_data[, nDx := as.numeric(nDx)]

####### calculate_life_table ###########################################
### Define function that calculates complete life table given arguments:
###   data = data.table with numeric columns:
###                     x = beginning of age interval
###                     nNx = total people reaching age interval
###                     nDx = total deaths in age interval
###   radix = numbers of survivors at age 0 (or births per year in a
###           stationary population); defaults to 100000
########################################################################
calculate_life_table <- function(data, radix = 100000) {
# 1. Calculate nmx
  data[, nmx := nDx / nNx]

# 2. Calculate nax given Coale and Demeny equations for ages <5 and n/2 for others.
  # For 1a0
  data[x == 0 & nmx >= .107, nax := 0.35]
  data[x == 0 & nmx < .107, nax := 0.053 + (2.8 * nmx)]
  # For 4a1
  m0 <- data[x == 0, nmx]
  data[x == 1 & shift(nmx, 1, type='lag') >= .107, nax := 1.361]
  data[x == 1 & shift(nmx, 1, type='lag') < .107, nax := 1.522 - (1.518 * m0)]
  # All other age groups
  data[x > 1, nax := 5 / 2]
  data[x == 85, nax := 1/nmx]

# 3. Calculate nqx
  data[x == 0, n := 1]
  data[x == 1, n := 4]
  data[x > 1, n := 5]
  data[, nqx := (n * nmx) / (1 + ((n - nax) * nmx))]
  data[x == 85, nqx := 1]

# 4. Calculate npx
  data[, npx := 1 - nqx]

# 5. Calculate lx
  data[, lx := radix]
  for(r in 2:length(data[, lx])) {
```

```r
    previous_lx <- data[r-1, lx]
    previous_npx <- data[r-1, npx]
    data[r, lx := previous_lx * previous_npx]
  }

# 6. Calculate ndx
  for(r in 1:length(data[, lx])) {
    lx_n <- data[r+1, lx]
    data[r, ndx := lx - lx_n]
    if(r == length(data[, lx])) data[r, ndx := lx]
  }

# 7. Calculate nLx
  for(r in 1:length(data[, lx])) {
    lx_n <- data[r+1, lx]
    data[r, nLx := (n * lx_n) + (nax * ndx)]
    if(r == length(data[, lx])) data[r, nLx := lx / nmx]
  }

# 8. Calculate Tx
  for(r in 1:length(data[, lx])) {
    data[r:length(data[, lx]), Tx := sum(nLx)]
  }

# 9. Calculate ex
  data[, ex := Tx / lx]

# Return complete life table
  return(data)

}

# Calculate standard life table with data for problem set and default radix
lt <- calculate_life_table(data = raw_data)
```

A.(1)

```r
# A.(1)
  lt[x == 0, ex]
```

```
## [1] 77.47132
```

A.(2)

```r
# A.(2)
  lt[x == 35, ex]
```

```
## [1] 44.2476
```

A.(3)

```r
# A.(3)
  # Given an individual survived until age 35, the expected value
```

```r
  # of the remaining length of their life in years is 44.2476
```

A.(4)

```r
# A.(4)
  prod(lt[x < 25, npx])
```

```
## [1] 0.9791639
```

```r
  lt[x == 25, lx] / lt[1, lx]
```

```
## [1] 0.9791639
```

A.(5)

```r
# A.(5)
  1 - prod(lt[x >= 25 & x < 50, npx])
```

```
## [1] 0.0241032
```

A.(6)

```r
# A.(6)
  sum(lt[x >= 15 & x < 65, nLx]) / lt[1, lx]
```

```
## [1] 47.9629
```

```r
  50 - sum(lt[x >= 15 & x < 65, nLx]) / lt[1, lx]
```

```
## [1] 2.037096
```

A.(7)

```r
# A.(7)
  lt[, nAx := nax * ndx]
  lt[x == 1, nax]
```

```
## [1] 1.501346
```

A.(8)

```r
# A.(8) = probability of surviving to age 65 * probability of dying between 65-69
  prod(lt[x < 65, npx]) * lt[x == 65, nqx]
```

```
## [1] 0.06768248
```

A.(9)

```r
# A.(9) probability(30-60) * probability(0-30)
  prod(lt[x < 30, npx]) * prod(lt[x >= 30 & x < 60, npx])
```

```
## [1] 0.916966
```

```r
(lt[x == 30, lx] / lt[x == 0, lx]) * (lt[x == 60, lx] / lt[x == 30, lx])
```

```
## [1] 0.916966
```

A.(10)

```r
# A.(10)
  # CBR = CDR
  1 / lt[1, ex]
```

```
## [1] 0.012908
```

```r
  # Death rate above 60
  1 / lt[x == 60, ex]
```

```
## [1] 0.04769172
```

```r
  # Mean age at death
  lt[1, ex]
```

```
## [1] 77.47132
```

```r
  # Given 56,059 births per year in this population, how many people turn 65 each year?
  stationary_lt <- calculate_life_table(data = copy(raw_data),
                                        radix = 56059)
  stationary_lt[x == 65, lx]
```

```
## [1] 49272.9
```