KOSTAT-UNFPA Summer Seminar on Population

Workshop 1. Demography in R

Day 3: Supplementary data preparation

Instructor: Tim Riffe
tim.riffe@gmail.com

Assistant: Rustam Tursun-Zade
rustam.tursunzade@gmail.com

28 July 2021

Contents

1 Summary

I decided to get abridged lifetables for countries in Africa for the Wednesday lesson, and this is taking more data wrangling that I originally thought, hence the separation of data prep into this script, for those that are interested to see what I did to the data. My goal was to give you rates ${}_nM_x$ and so-called ${}_nA_x$ (the average time spent in an age interval by those dying in the interval). However, I learned that the GHO data source I chose does not share the ${}_nA_x$ values used in their lifetables (or at least I couldn't find them), so I decided to derive them myself from the other columns given. The details are given in the following.

package prerequisites:

```
# install.packages("rgho")
# install.packages("countrycode")
library(rgho)
library(tidyverse)
library(countrycode)
library(readr)
library(here)
```

2 Reading in the data

I got the data from the WHO-GHO API using the `rgho` package (found via web search), and following the tutorial here: https://cran.r-project.org/web/packages/rgho/vignettes/a-intro.html with a little bit of data wrangling and tenacity.

```
# this is how I found out the data file codes.
# search_codes("nqx") %>% str()

# once I got the codes, each lifetable column needs to be downloaded
# separately
nMx <- get_gho_data(
  dimension = "GHO",
  code = "LIFE_0000000029",
  show_col_types = FALSE) %>%
  filter(REGION == "AFR")

nqx <- get_gho_data(
  dimension = "GHO",
  code = "LIFE_0000000030",
  show_col_types = FALSE) %>%
  filter(REGION == "AFR")

# Actually we just want these for the open
# age group, so we can filter to the relevant
# age 85+
lx <-  get_gho_data(
  dimension = "GHO",
  code = "LIFE_0000000031",
  show_col_types = FALSE) %>%
  filter(REGION == "AFR",
         AGEGROUP == "AGE85PLUS")

Tx <-  get_gho_data(
  dimension = "GHO",
  code = "LIFE_0000000034") %>%
  filter(REGION == "AFR",
         AGEGROUP == "AGE85PLUS")
```

3 Prepping the data

3.1 Join lx and Tx

T_x/l_x gives life expectancy e_x in general, but we here only need it for the open age interval, where it ought to be equal to ${}_\infty A_{85}$...

I found out the hard way that some country /year combinations are in the data twice, due to over-coding the `WORLDBANINCOMEGROUP` variable, which can be both missing and labelled, for the same location and year, even with different lifetable values! Very odd, and I don't know what to make of it. I decided to remove this redundancy by filtering out the `NA` entries. In each downloaded subset the `Numeric` column contains the respective lifetable column, so we need to be sure to rename. Countries are given using ISO3 codes, so we rename, and can derive country names using another look-up service later once everything is joined.

```
lx <-
  lx %>%
  filter(!is.na(WORLDBANKINCOMEGROUP)) %>%
  select(ISO3 = COUNTRY,
         YEAR,
         SEX,
         AGEGROUP,
         lx = Numeric,
         # saved just in case there were still redundancies,
         # but actually this is an extraneous column now.
         Inc = WORLDBANKINCOMEGROUP)
# same thing for Tx.
Tx <-
  Tx %>%
  filter(!is.na(WORLDBANKINCOMEGROUP)) %>%
  select(ISO3 = COUNTRY,
         YEAR,
         SEX,
         AGEGROUP,
         Tx = Numeric,
         Inc = WORLDBANKINCOMEGROUP)
# join together
Close <-
  lx %>%
  # inner, full, left, right would all give same result!
  inner_join(Tx, by = c("ISO3", "YEAR", "SEX", "AGEGROUP", "Inc")) %>%
  # equal to e85, same thing
  mutate(nAx = Tx / lx)
```

3.2 Back out nAx from ${}_nq_x$ and ${}_nM_x$

For ages < 85, we'll shuffle the common formula used to derive ${}_nq_x$ from ${}_nM_x$ and ${}_nA_x$. *Usually* we solve for ${}_nq_x$, having approximated ${}_nM_x$ directly from events and exposures, and having estimated ${}_nA_x$ from rough standard procedures.

$$ {}_nq_x = \frac{n * {}_nM_x}{1 - (n - {}_nA_x) * {}_nM_x} $$

Solving for ${}_nA_x$ we get:

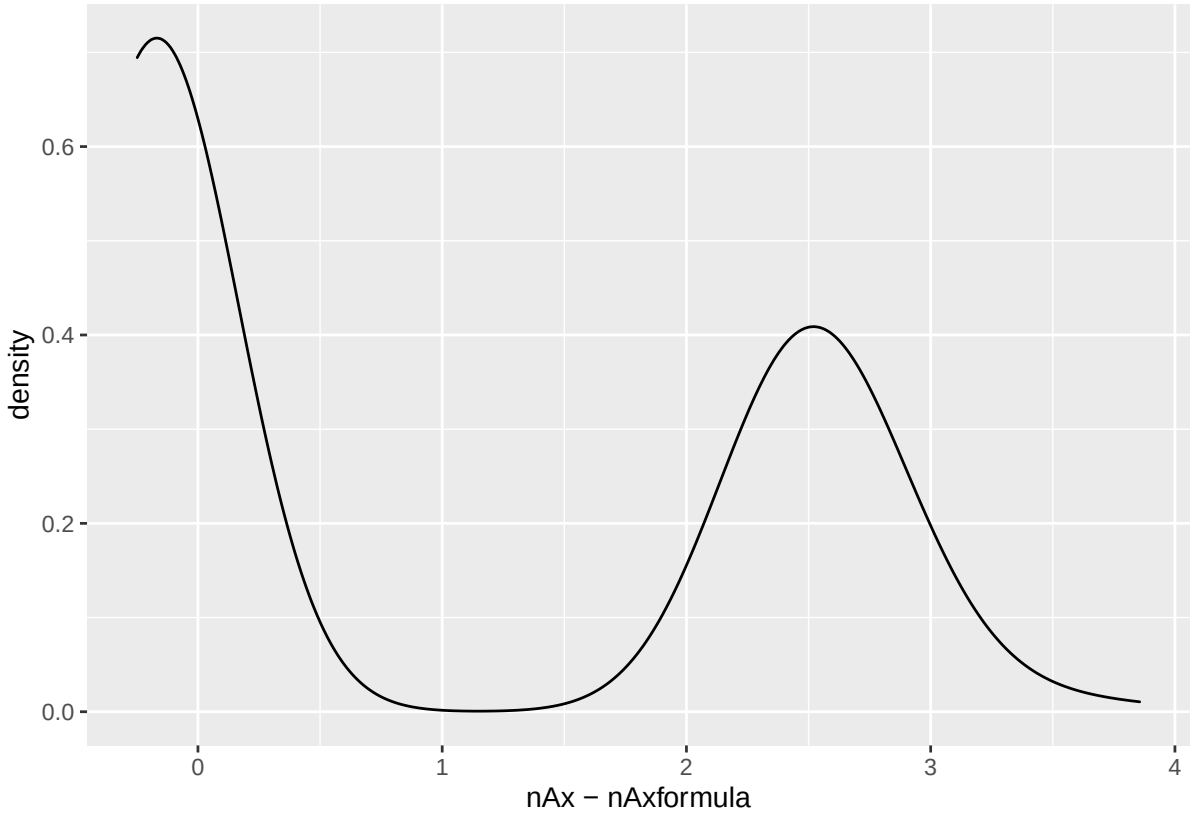
$$ {}_nA_x = \frac{1}{{}_nM_x} - \frac{n}{{}_nq_x} + n $$

Ready to join and derive ${}_nA_x$!

```
# same sequence as the above for these two columns
nqx <-
  nqx %>%
  filter(!is.na(WORLDBANKINCOMEGROUP)) %>%
  select(ISO3 = COUNTRY,
         YEAR,
         SEX,
         AGEGROUP,
         nqx = Numeric,
         Inc = WORLDBANKINCOMEGROUP)
nMx <-
  nMx %>%
  filter(!is.na(WORLDBANKINCOMEGROUP)) %>%
  select(ISO3 = COUNTRY,
         YEAR,
         SEX,
         AGEGROUP,
         nMx = Numeric,
         Inc = WORLDBANKINCOMEGROUP)
LT <-
  nqx %>%
  inner_join(nMx,
             by = c("ISO3", "YEAR", "SEX", "AGEGROUP", "Inc")) %>%
  mutate(
    n = case_when(AGEGROUP == "AGELT1" ~ 1,
                  AGEGROUP == "AGE1-4" ~ 4,
                  TRUE ~ 5),
    # solves nqx formula for a
    nAx = 1 / nMx - n / nqx + n,
    # clean up some pathological cases:
    nAx = ifelse(nAx > n | nAx < 0, n / 2, nAx))
```

Let's examine 3 versions of age 85 closeout to determine what WHO does. We determine they don't *merely* use $1/{}_nM_x$ to close out (only recommended/innocuous after age 100 according to me).

```
LT %>%
  filter(AGEGROUP == "AGE85PLUS") %>%
  rename(nAxformula = nAx) %>%
  mutate(nAxratio = 1 / nMx) %>%
  full_join(Close, by =  c("ISO3", "YEAR", "SEX", "AGEGROUP", "Inc")) %>%
  ggplot(aes(x = nAx - nAxformula))+
  geom_density()
```



We could do some forensics and figure out what closeout model they likely use, but let's just say "they do something that was considered" and be satisfied. Then, we can remove age 85 fro `LT` and bind on `Close`.

```
# name-selection as a recoding technique when
# you have lots of codes to handle
recvec        <- c(0,1,seq(5,85,by=5))
names(recvec) <- LT$AGEGROUP %>% unique()

LT_inputs <-
  Close %>%
  select(-lx, Tx) %>%
  rename(nAxnew = nAx) %>%
  right_join(LT, by = c("ISO3", "YEAR", "SEX", "AGEGROUP", "Inc")) %>%
  mutate(nAx = ifelse(AGEGROUP == "AGE85PLUS", NA_real_, nAx),
         # coalesce is awesome, check out ?coalesce
         nAx = coalesce(nAx, nAxnew)) %>%
  select(ISO3, Year = YEAR, SEX, AGEGROUP, nMx, nAx) %>%
  mutate(Sex = case_when(SEX == "MLE" ~ "m",
                         SEX == "FMLE" ~ "f",
                         SEX == "BTSX" ~ "t")) %>%
  mutate(Age = recvec[AGEGROUP],
         # countrycode package is a great resource!!!
         Country = countrycode(sourcevar = ISO3,
                               origin = "iso3c",
                               destination = "country.name")) %>%
  arrange(Country, Year, Sex, Age) %>%
  select(Country, ISO3, Year, Sex, Age, nMx, nAx)
```

```r
# save the results
write_csv(LT_inputs, file = here("Data","LT_inputs.csv"))
```

This resulting file will be what we use in class for lifetables.