Example 2 using R

Drawing age to death from homogeneous cohort

David Garibay, M.P.P.*   Hawre Jalal, MD, Ph.D.†

Fernando Alarid-Escudero, Ph.D.‡§

Code function

This document presents the code corresponding to the second example presented in the "A Fast Nonparametric Sampling (NPS) Method for Time-to-Event in Individual-Level Simulation Models." manuscript, all of them using R.

```r
# 01 Initial Setup ---------------------------------------------------------

# Clean global environment
remove(list = ls())

# Free unused R memory
gc()

# Load libraries
library(dplyr)
library(ggplot2)
library(tidyr)
library(tibble)
library(microbenchmark)
```

---

*School of Epidemiology and Public Health, Faculty of Medicine, University of Ottawa, Ottawa, ON, CA.

†School of Epidemiology and Public Health, Faculty of Medicine, University of Ottawa, Ottawa, ON, CA.

‡Department of Health Policy, Stanford University School of Medicine, Stanford, CA, USA.

§Center for Health Policy, Freeman Spogli Institute, Stanford University, Stanford, CA, USA.

```r
# 02 Define general parameters ------------------------------------------

# Number of samples to draw from the life table
n_samp_life_tables <- 1e5

# Number of iterations for microbenchmarking
n_samp_iter_life_tables <- 100

# Seed for reproducibility in random number generation
n_seed <- 10242022

# To print a specific number of digits in tibbles
options(pillar.sigfig = 4)


# 03 Load base data -----------------------------------------------------

#* Yearly USA data, from 2000 to 2019,
#* Mortality rate for males, females and total
#* Obtained from The Human Mortality Database:
#* https://www.mortality.org/cgi-bin/hmd/country.php?cntr=USA&level=1
load("../data/all_cause_mortality.rda")


# 04 Filter data --------------------------------------------------------

# For homogeneous population example
df_all_cause_mortality_filt <- all_cause_mortality %>%
  as_tibble() %>%
  filter(Year == 2015)


# 05 Data wrangling -----------------------------------------------------

#* Following Lee & Wang (2013)-Statistical methods for survival data analysis
#* 4th ed - chapter 2: Functions of survival time
df_lifetable <- df_all_cause_mortality_filt %>%
  dplyr::arrange(Sex, Year, Age) %>%
  dplyr::group_by(Sex) %>%
  dplyr::mutate(
    H_t = cumsum(Rate),        # H(t) - Cumulative hazard
    S_t = exp(-H_t),           # S(t) - Cumulative survival
    F_t = 1 - exp(-H_t),       # F(t) - Cumulative probability: 1 - S(t)
    p_t = c(F_t[1], diff(F_t)) # f(t) - Instantaneous probability
  ) %>%
```

```r
  ungroup()


# Calculate life expectancy from lifetables data
df_le_lifetable <- df_lifetable %>%
  group_by(Sex) %>%
  summarise(le = sum(S_t))


# Obtain life expectancy from lifetables
le_lifetable_homog <- df_le_lifetable[df_le_lifetable$Sex == "Total", ]$le
```

```r
# Set seed for reproducibility in random number generation
set.seed(n_seed)

# 06 Calculate life expectancy using nps method --------------------------

# Filter to have homogeneous population
df_lifetable_homog <- df_lifetable %>%
  filter(Sex == "Total")


#* Sample ages to death from a categorical sampling
v_cat_life_table_homog <- sample(x       = df_lifetable_homog$Age,
                                 size    = n_samp_life_tables,
                                 prob    = df_lifetable_homog$p_t,
                                 replace = TRUE)

#* Create vector of drawings following a uniform distribution
v_unif_life_table_homog <- runif(n = n_samp_life_tables, min = 0, max = 1)

#* Add this vector to the categorical sampling outputs
v_cat_life_table_corr_homog <- (v_cat_life_table_homog +
                                  v_unif_life_table_homog)

#* Life expectancy without continuous time correction
le_homog_uncorr <- mean(v_cat_life_table_homog)

#* Life expectancy with correction
le_homog_corr <- mean(v_cat_life_table_corr_homog)

# Measure mean execution time
```

```r
## Without continuous time correction
l_mbench_homog_uncorr <- microbenchmark::microbenchmark(
  sample(x       = df_lifetable_homog$Age,
         size    = n_samp_life_tables,
         prob    = df_lifetable_homog$p_t,
         replace = TRUE),
  times = n_samp_iter_life_tables,
  unit = "ms")


## With continuous time correction
l_mbench_homog_corr <- microbenchmark::microbenchmark(
  sample(x       = df_lifetable_homog$Age,
         size    = n_samp_life_tables,
         prob    = df_lifetable_homog$p_t,
         replace = TRUE) + runif(n = n_samp_life_tables, min = 0, max = 1),
  times = n_samp_iter_life_tables,
  unit = "ms")



# Remove seed
set.seed(NULL)
```