

# Example 3 using R

## Drawing age to death from a heterogeneous cohort

David Garibay, M.P.P.\*      Hawre Jalal, MD, Ph.D.<sup>†</sup>  
Fernando Alarid-Escudero, Ph.D.<sup>‡§</sup>

### Code function

This document presents the code corresponding to the third 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.

```
# 01 Initial Setup -----  
  
## 01.01 Clean environment -----  
remove(list = ls())  
  
## Refresh environment memory  
gc()  
  
## 01.02 Load libraries -----  
library(dplyr)  
library(ggplot2)  
library(tidyr)  
library(tibble)  
library(microbenchmark)  
  
# Load function to implement multivariate categorical sampling  
source(file = "../R/nps_nhppp.R")
```

---

\*Health Research Consortium (CISIDAT), Cuernavaca, Morelos, Mexico.

<sup>†</sup>School of Epidemiology and Public Health, Faculty of Medicine, University of Ottawa, Ottawa, ON, CA.

<sup>‡</sup>Department of Health Policy, Stanford University School of Medicine, Stanford, CA, USA.

<sup>§</sup>Center for Health Policy, Freeman Spogli Institute, Stanford University, Stanford, CA, USA.

```

# 02 Define general parameters -----

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

# Number of samples, by sex, to draw from the life table
n_samp_by_sex <- 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
  ) %>%
  ungroup()

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

# Obtain life expectancy by sex from life tables
le_lifetable_fem <- df_le_lifetable[df_le_lifetable$Sex == "Female",]$le
le_lifetable_male <- df_le_lifetable[df_le_lifetable$Sex == "Male",]$le

# Seed for reproducibility in random number generation
set.seed(n_seed)

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

v_samp_sex <- c("Male", "Female")

# Filter to have heterogeneous population
df_lifetable_heterog <- df_lifetable %>%
  filter(Sex != "Total") %>%
  select(Year, Sex, Age, p_t) %>%
  # Normalize probabilities by sex
  group_by(Sex) %>%
  # Normalize instantaneous probabilities by sex
  mutate(p_t = p_t / sum(p_t)) %>%
  ungroup()

# Generate synthetic cohort with 50% males and 50% females
df_samp_raw <- tibble::tibble(
  Year = 2015,
  Sex = c(rep(x = v_samp_sex,
              each = n_samp_by_sex)))

# Convert lifetable data from long to wide
df_lifetable_probs_wide <- tidyr::pivot_wider(
  data = df_lifetable_heterog,

```

```

names_from = Age,
values_from = p_t,
names_prefix = "Age_")

# Generate dataset for Multivariate categorical sampling
df_samp_probs <- df_samp_raw %>%
  left_join(y = df_lifetable_probs_wide,
            by = join_by(Year, Sex))

## Extract probability matrix from `df_samp_probs`
m_probs <- df_samp_probs %>%
  select(-Year, -Sex) %>%
  as.matrix()

## Implement Multivariate NPS
### Without continuous time approximation
v_cat_life_table_heterog <- nps_nhppp(m_probs = m_probs,
                                     correction = "none")

### With continuous time approximation
v_cat_life_table_heterog_corr <- nps_nhppp(m_probs = m_probs,
                                     correction = "uniform")

# Create dataset with the age to death samples
df_heterog_samp <- df_samp_raw %>%
  mutate(age_death = v_cat_life_table_heterog,
         age_death_corr = v_cat_life_table_heterog_corr,
         .after = Sex)

## Obtain life expectancies by sex
df_le_nps_heterog <- df_heterog_samp %>%
  group_by(Sex) %>%
  summarise(le = mean(age_death),
            le_corr = mean(age_death_corr)) %>%
  ungroup()

# Extract values
le_nps_fem_uncorr <- filter(df_le_nps_heterog, Sex == "Female")$le
le_nps_male_uncorr <- filter(df_le_nps_heterog, Sex == "Male")$le

le_nps_fem_corr <- filter(df_le_nps_heterog, Sex == "Female")$le_corr

```

```

le_nps_male_corr    <- filter(df_le_nps_heterog, Sex == "Male")$le_corr

# Measure mean execution time
## Without continuous time correction
l_mbench_heterog_uncorr <- microbenchmark::microbenchmark(
  nps_nhppp(m_probs = m_probs, correction = "none"),
  times = n_samp_iter_life_tables,
  unit = "ms")

## With continuous time correction
l_mbench_heterog_corr <- microbenchmark::microbenchmark(
  nps_nhppp(m_probs = m_probs, correction = "uniform"),
  times = n_samp_iter_life_tables,
  unit = "ms")

# Remove seed
set.seed(NULL)

```