

# Example 1 using python

## Time to event from parametric hazards

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 python code corresponding to the first example presented in the “A Fast Nonparametric Sampling (NPS) Method for Time-to-Event in Individual-Level Simulation Models.” manuscript. Since python has different parametrizations of the Gamma and Lognormal distributions, present in the example using R, this document only shows the code to replicate the example using the exponential function. .

```
# 01 Initial Setup -----  
  
# Import required modules  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import scipy.stats as stats  
import pandas as pd  
  
# 02 General parameters -----  
  
# Exponential rate  
rate = 0.1  
  
# Sample size  
n_samp = int(1e6)
```

---

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

<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.

```

# 03 Data wrangling -----

# Obtain analytical values
a_true_mean    = 1/rate          ## mean
a_true_median  = np.log(2)/rate  ## median
a_true_sd      = (1/(rate**2))*(1/2) ## SD

# Derive PDF from CDF
a_prob_exp_rates = (stats.expon.cdf(np.arange(1, 152), scale = 1/rate) -
                    stats.expon.cdf(np.arange(0, 151), scale = 1/rate))

# Normalize PDF
a_norm_exp_probs = a_prob_exp_rates/sum(a_prob_exp_rates)

# Sample values from normalized probabilities
a_random_exp_sample = np.random.choice(
    a      = np.arange(0, 151),
    size   = n_samp,
    replace = True,
    p      = a_norm_exp_probs)

# Add random number between 0 and 1 to approximate continuous time
a_random_exp_corr = (a_random_exp_sample +
                     np.random.random_sample(size = n_samp))

```