Advanced Discrete Event Simulation in R

TA Trikalinos
(joint work with F Alarid-Escudero, Y Sereda, SA Chrysanthopoulou)

CISNET, 28/05/2025

Disclosures

- No financial or other conflicts
- Supported by the NCI CISNET Incubator (bladder, all) and CISNET programs (colorectal - FAE)
- Trikalinos and Sereda developed and maintain the **nhppp** R package
- Half day course at SMDM 2025 – Take it, it’s all fun and games.

Outline

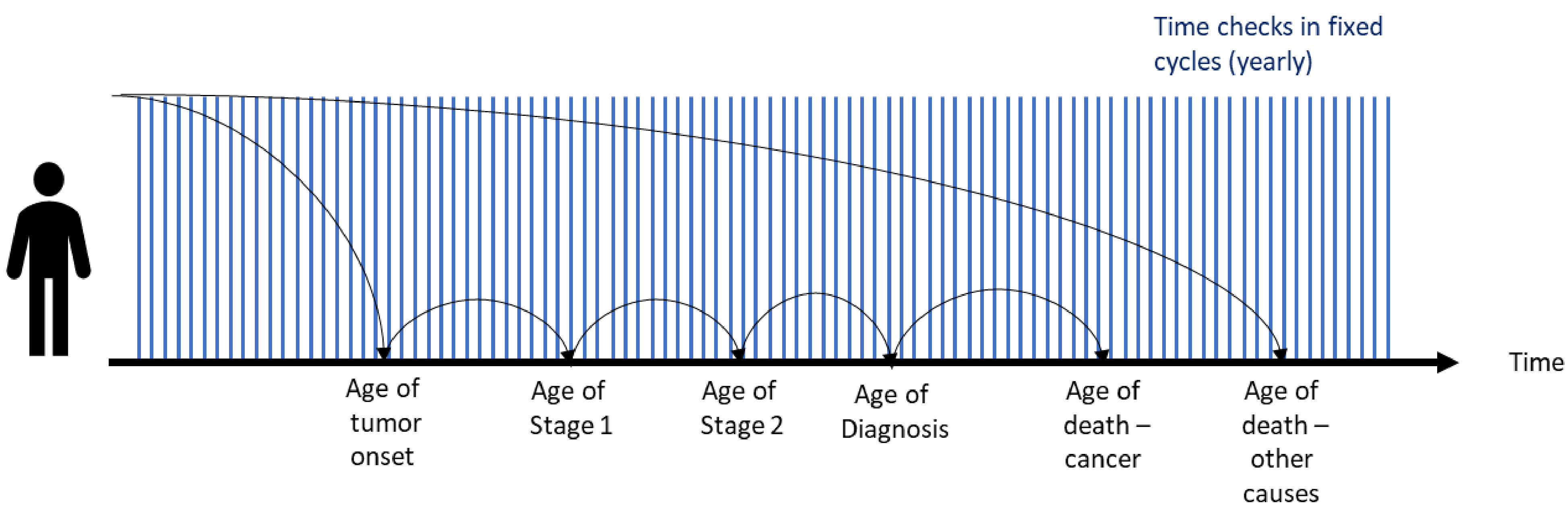
- Discrete event simulation as a convolution of point processes
- Non-homogeneous Poisson point processes (NHPPPs)
- Sampling from NHPPPs
- Demonstration

Outline

- Discrete event simulation as a convolution of point processes
- Non-homogeneous Poisson point processes (NHPPPs)
- Sampling from NHPPPs
- Demonstration

Background

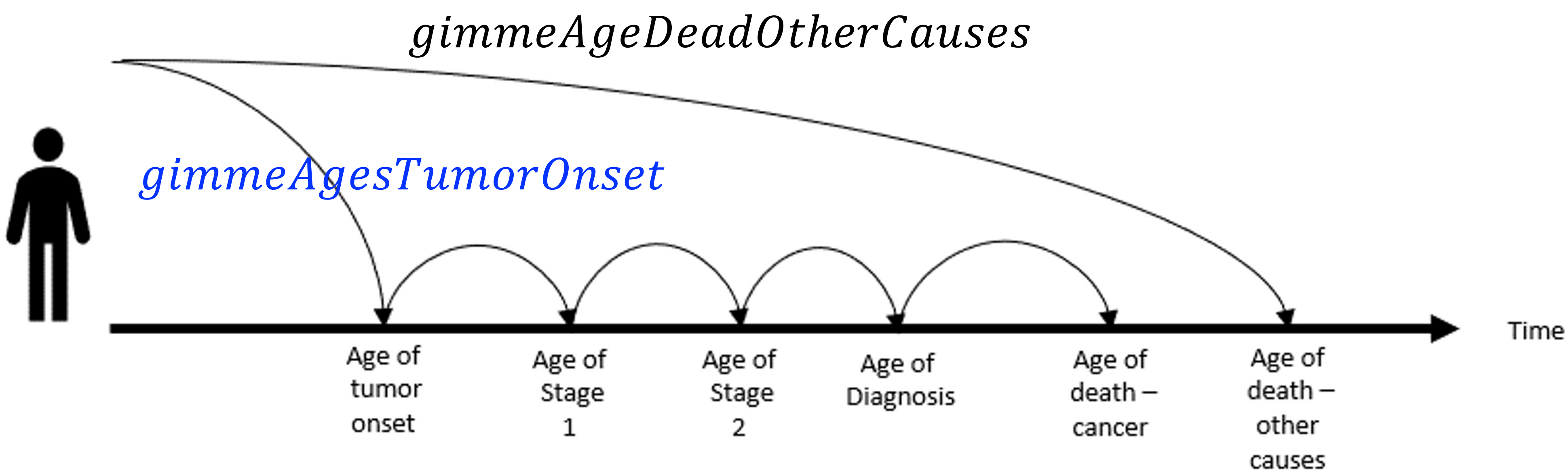
Discrete-time simulation samples events in each cycle



Thanks to Carlos Pineda

Background

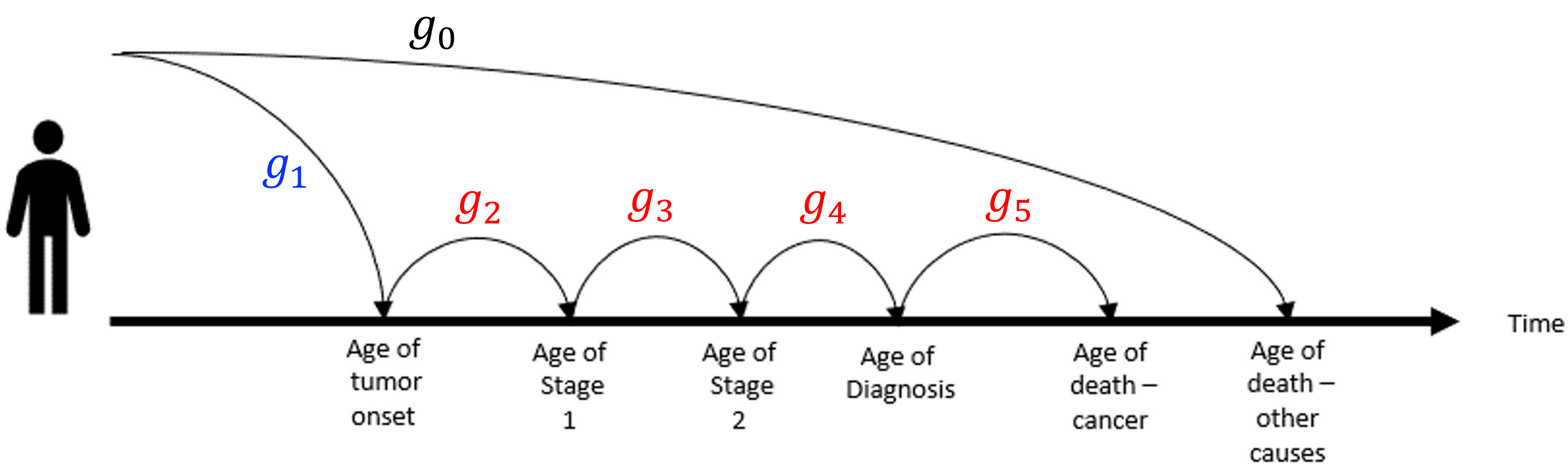
Discrete-event simulation works with event-generating processes



Thanks to Carlos Pineda

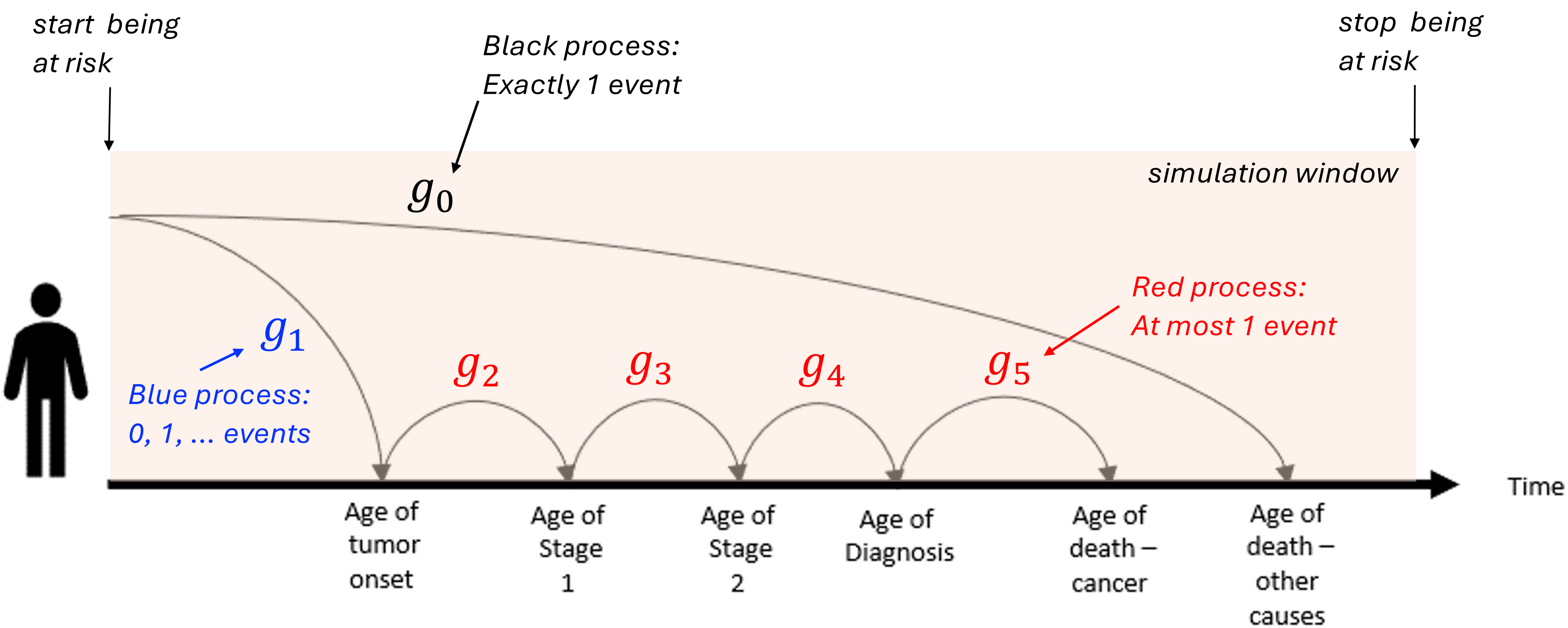
Background

Discrete-event simulation works with event-generating processes



Thanks to Carlos Pineda

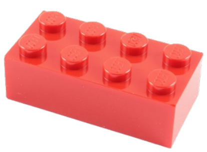
A typology of event-generating processes



Thanks to Carlos Pineda

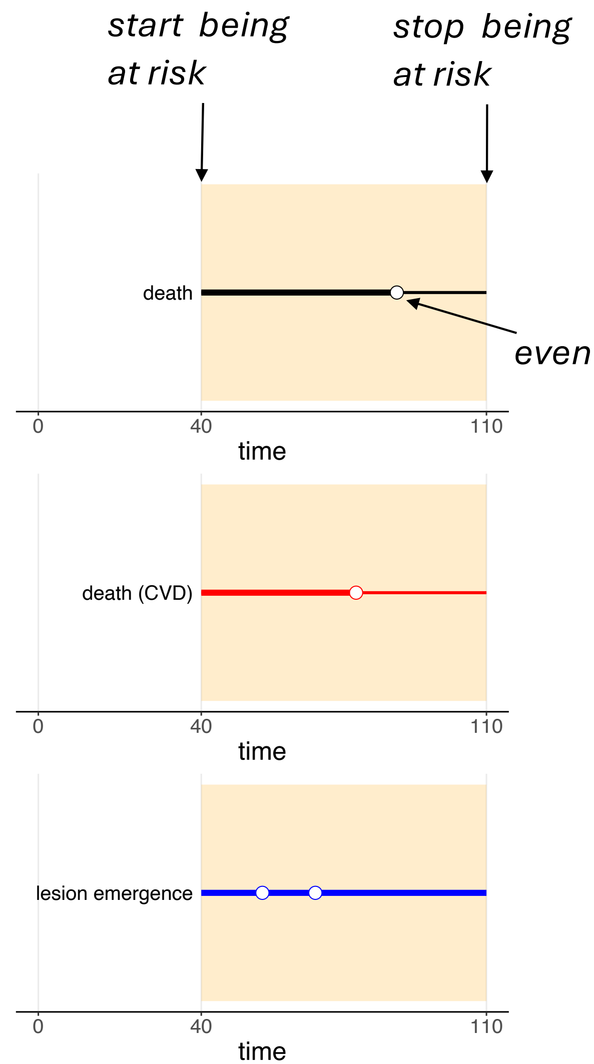
The building blocks of a DES

Events that happen exactly once



Events that happen 0 or 1 times

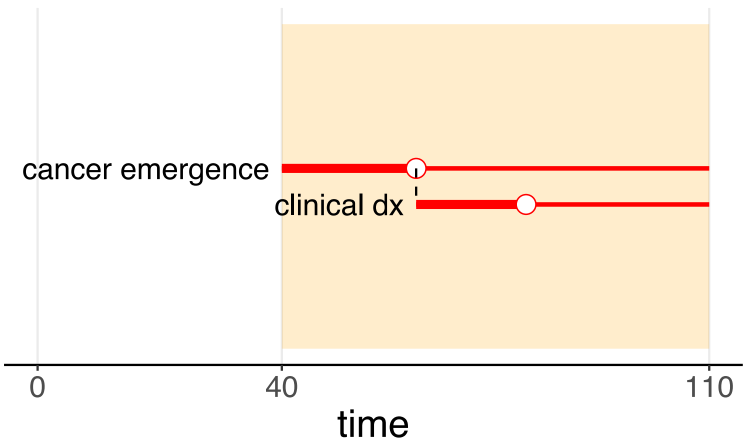
Events that happen 0, 1, ... times



Graphical notation: Chained events (in series)

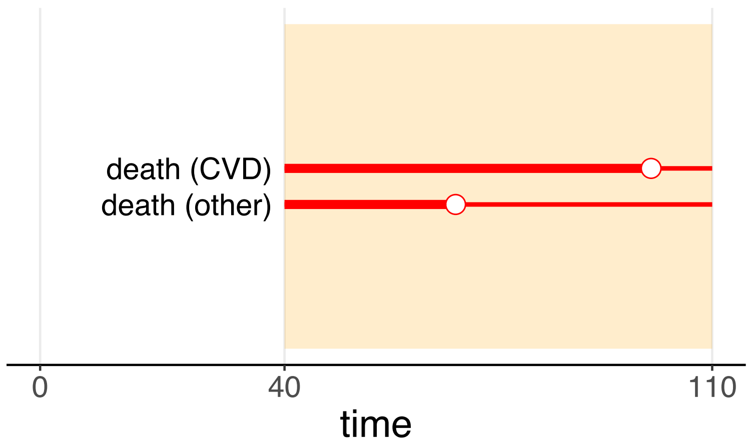
For chained processes, the next one starts once the preceding one realizes an event.

imagine simulating the first row first, etc.



Graphical notation: Competing events (parallel)

Competing event processes run parallel to each other.



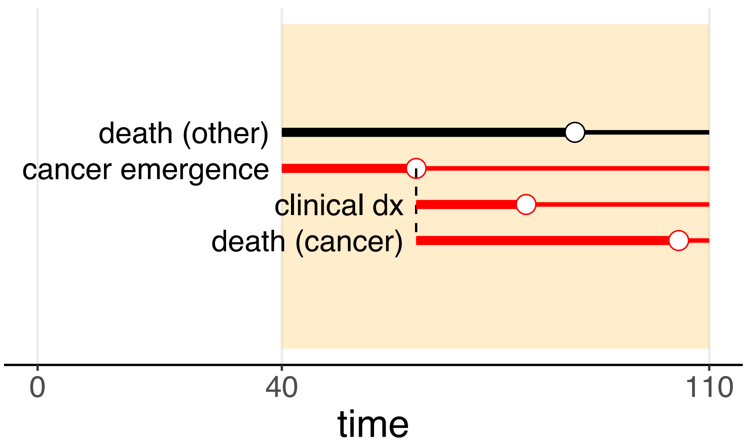
A simple DES model

All shall die.

Some may develop a cancer.

Some cancers will

- be clinically diagnosed, or
- cause deaths



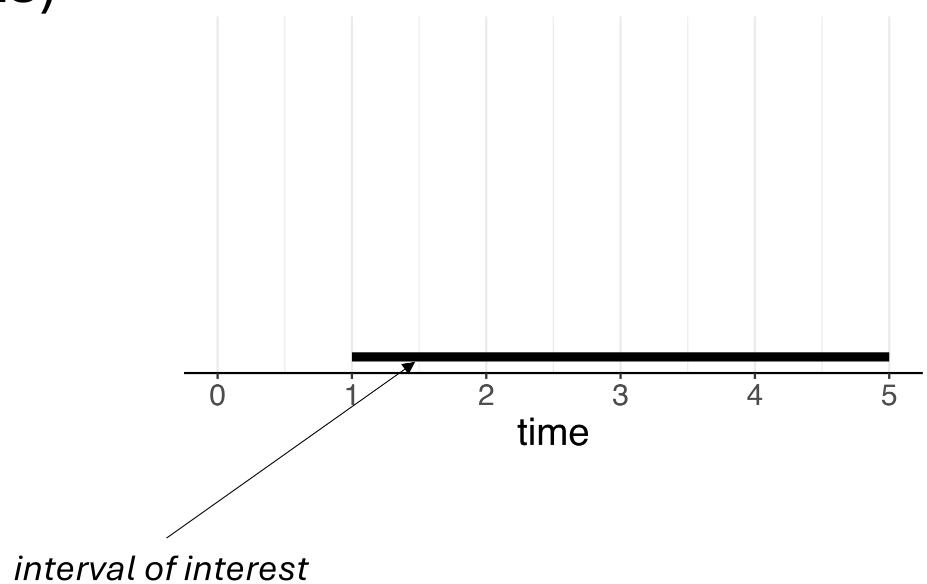
Outline

- Discrete event simulation as a convolution of point processes
- Non-homogeneous Poisson point processes (NHPPPs)
- Sampling from NHPPPs
- Demo

The building block

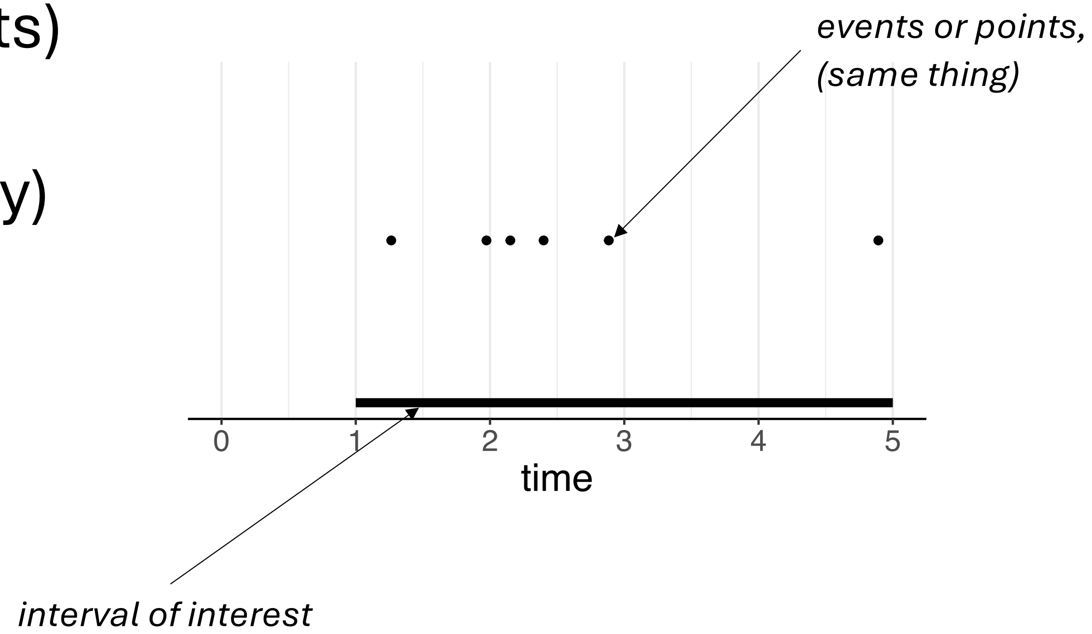
The point process

- A scheme that generates a sequence of events (points) over a time interval



The point process

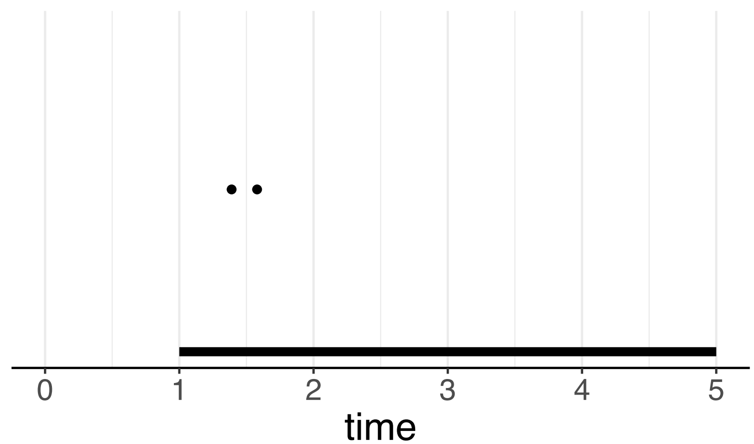
- A scheme that generates a sequence of events (points) over time
- An instantiation (trajectory) of the process is a sequence of 0, 1 or more events in the interval, but none outside it



The point process

- A scheme that generates a sequence of events (points) over time
- Each instantiation is random

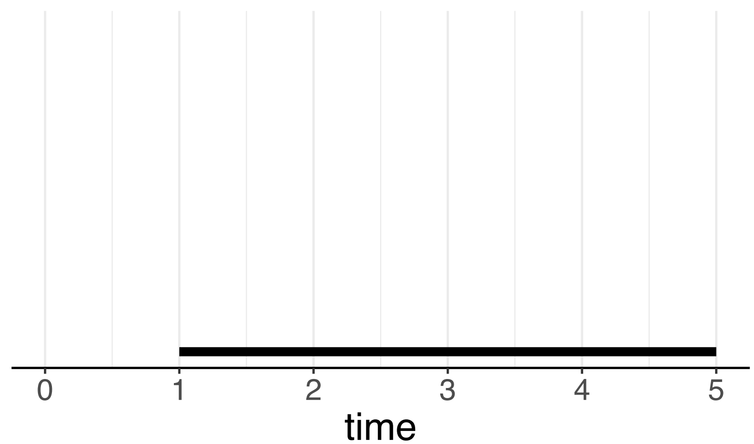
another instantiation (trajectory)



The point process

- A scheme that generates a sequence of events (points) over time
- Each instantiation is random

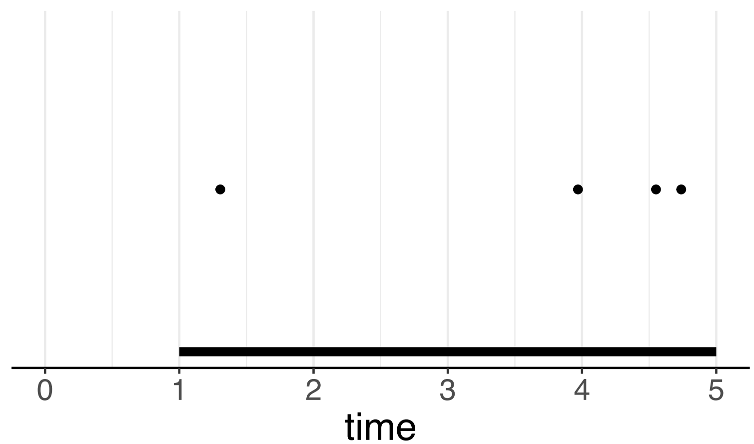
another instantiation (with no events)



The point process

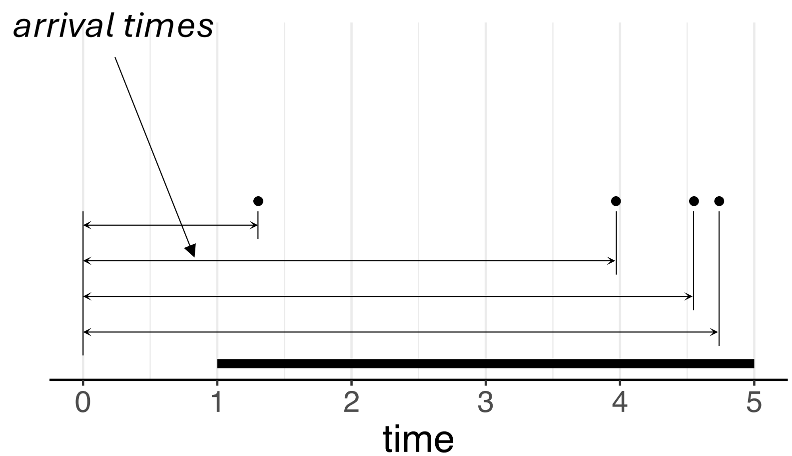
- A scheme that generates a sequence of events (points) over time
- Each instantiation is random

yet another instantiation



The point process

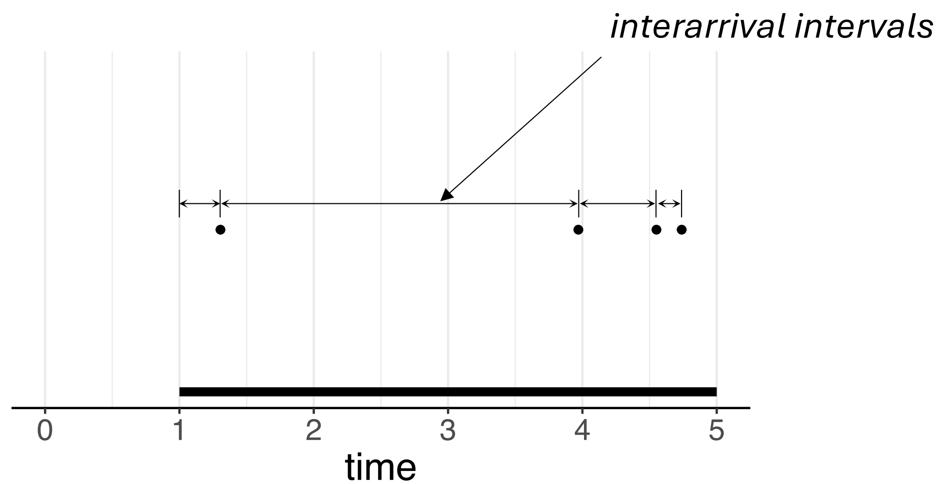
- The *arrival times* (times of the events) are random
- They start from whenever we zeroed the clock



The point process

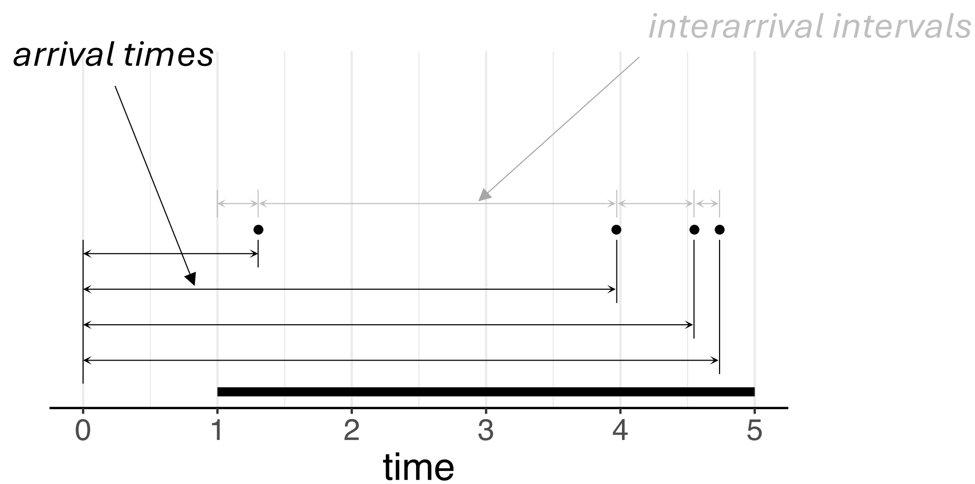
- The interarrival times are the lengths of the interarrival time intervals
- The arrival times and interarrival times give the same information

  (… thus, the interarrival times are random)



The point process

Hereon, we refer only to arrival times

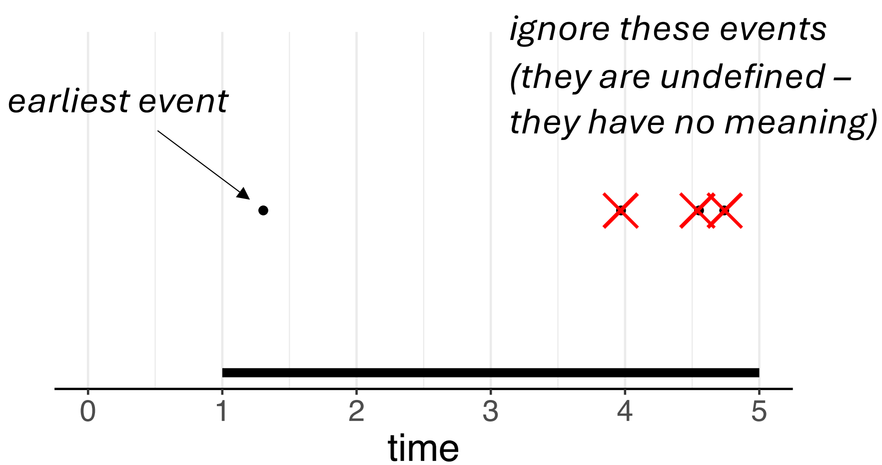


Modeling non-repeatable events

If the point process models a *nonrepeatable* event, we care only about the **earliest event.**

Will it occur in the interval, and, and if so, when?

Example: model a cause of death

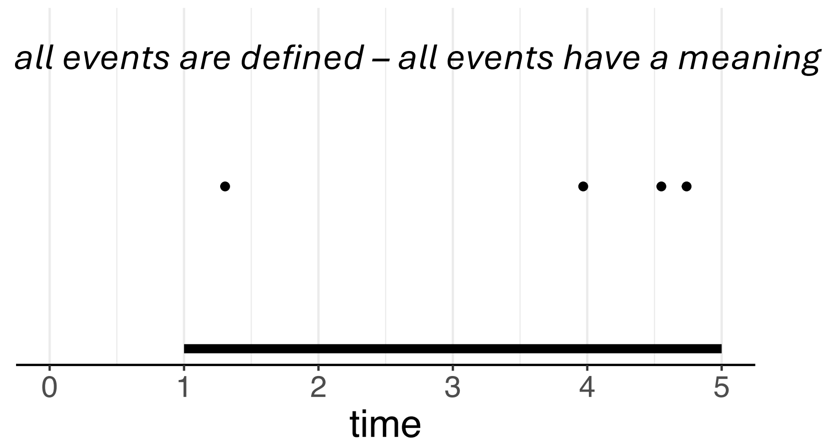


Modeling repeatable events

If the point process models a *repeatable* event, we care are about **all events**.

Will any occur in the interval, and, and if so, when?

Example: model the emergence of tumors, or the start of symptomatic episodes



The Poisson point process

- There are many types of point processes
- We will consider only a one type – the Poisson point process

The Poisson point process

Number of events between t and t + Δt

If for a sequence of events

$o(\Delta t)$ becomes 0 ***very fast***

$$
\begin{aligned}
\Pr[\boxed{N(t, t+\Delta t)} = 0] &= 1 - \lambda\Delta t + o\cancel{(\Delta t)},\\
\Pr[N(t, t+\Delta t) = 1] &= \lambda\Delta t + o\cancel{(\Delta t)},\\
\Pr[N(t, t+\Delta t) > 1] &= o\cancel{(\Delta t)}, \text{ and}\\
N(t, t+\Delta t) &\perp\!\!\!\perp N(0, t),
\end{aligned}
$$

for some $\lambda > 0$ and as $\Delta t \to 0$, then that sequence is a Poisson point process

The Poisson point process (in English)

If for a sequence of events

$$\Pr[N(t, t+\Delta t) = 0] = 1 - \lambda \Delta t$$
$$\boxed{\Pr[N(t, t+\Delta t) = 1] = \lambda \Delta t}$$
$$\Pr[N(t, t+\Delta t) > 1] = 0 \quad \text{and}$$
$$N(t, t+\Delta t) \perp\!\!\!\perp N(0, t),$$

for some $\lambda > 0$ and as $\Delta t \rightarrow 0$, then that sequence is a Poisson point process

Over a vanishingly small interval

- *you may get 1 event with probability $\lambda \Delta t$ …*

The Poisson point process (in English)

If for a sequence of events

$$\boxed{\Pr[N(t, t+\Delta t) = 0] = 1 - \lambda \Delta t}$$
$$\Pr[N(t, t+\Delta t) = 1] = \lambda \Delta t$$
$$\Pr[N(t, t+\Delta t) > 1] = 0 \qquad \text{and}$$
$$N(t, t+\Delta t) \perp\!\!\!\perp N(0, t),$$

for some $\lambda > 0$ and as $\Delta t \to 0$, then that sequence is a Poisson point process

Over a vanishingly small interval

- *you may get 1 event with probability $\lambda \Delta t$ ...*
- *otherwise, you'll get 0 events;*

The Poisson point process (in English)

If for a sequence of events

$$\Pr[N(t, t+\Delta t) = 0] = 1 - \lambda\Delta t$$
$$\Pr[N(t, t+\Delta t) = 1] = \lambda\Delta t$$
$$\boxed{\Pr[N(t, t+\Delta t) > 1] = 0 \quad \text{and}}$$
$$N(t, t+\Delta t) \perp\!\!\!\perp N(0, t),$$

for some $\lambda > 0$ and as $\Delta t \to 0$, then that sequence is a Poisson point process

Over a vanishingly small interval

- *you may get 1 event with probability $\lambda\Delta t$ …*
- *otherwise, you'll get 0 events;*
- *you'll never get many concurrent events*

The Poisson point process (in English)

If for a sequence of events

$$\Pr[N(t, t + \Delta t) = 0] = 1 - \lambda \Delta t$$
$$\Pr[N(t, t + \Delta t) = 1] = \lambda \Delta t$$
$$\Pr[N(t, t + \Delta t) > 1] = 0 \quad \text{and}$$
$$\boxed{N(t, t + \Delta t) \perp\!\!\!\perp N(0, t),}$$

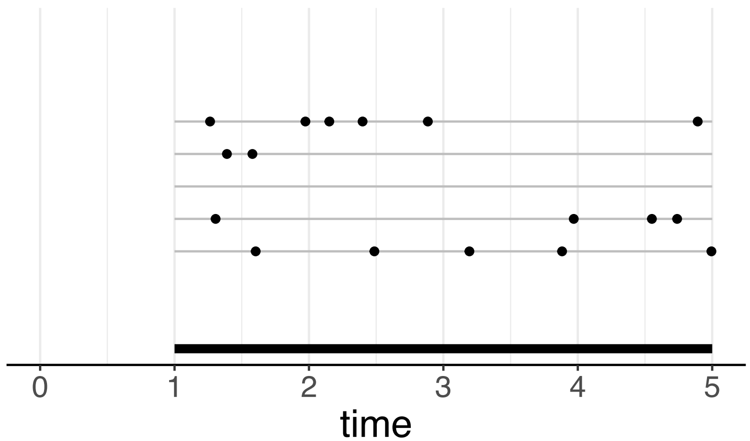
for some $\lambda > 0$ and as $\Delta t \to 0$, then that sequence is a Poisson point process

Over a vanishingly small interval

- *you may get 1 event with probability $\lambda\Delta t$ …*
- *otherwise, you'll get 0 events;*
- *you'll never get many concurrent events*
- *and it does not matter what happened in the past*

The intensity function λ in the example

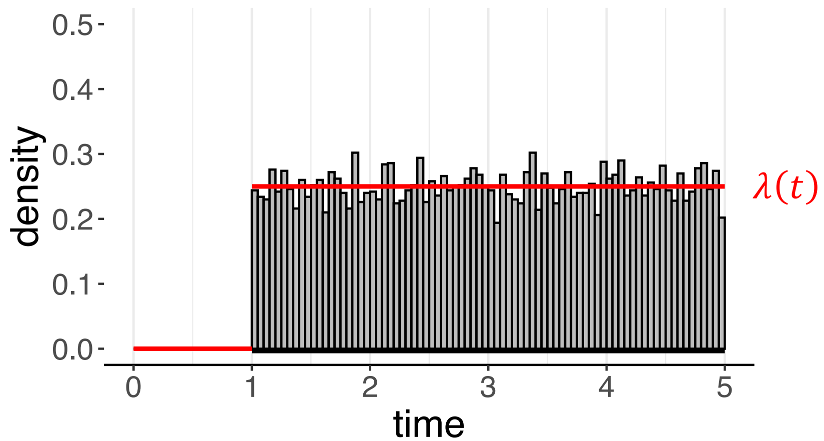
Event times for five instantiations



The intensity function λ in the example

As the number of instantiations increases, the histogram approaches the shape of the intensity function $\lambda(t)$.

The intensity function governs event occurrence.



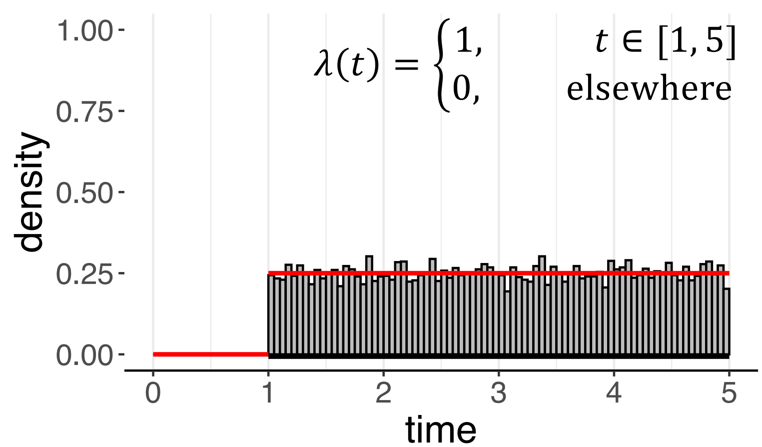
The intensity function is scaled by the expected number of events in the interval to be on the same plot

Time-homogeneous and non-homogeneous

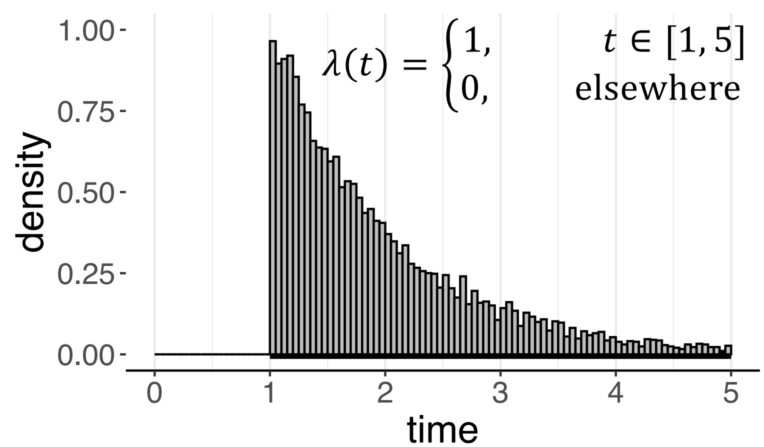
- $\lambda(\mathrm{t}) = \mathrm{constant}$: the Poisson point process (PPP) is called time-homogeneous
- Otherwise, it is called a non-homogeneous PPP (NHPPP)

All events vs earliest event in the example

All events, 10K instantiations



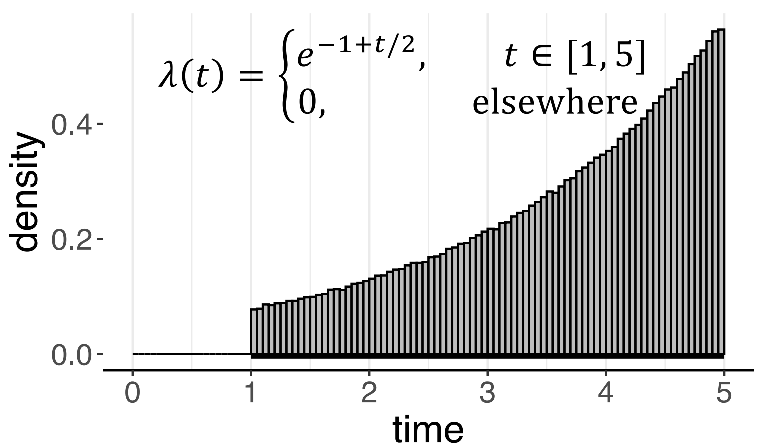
Earliest event, 10K instantiations



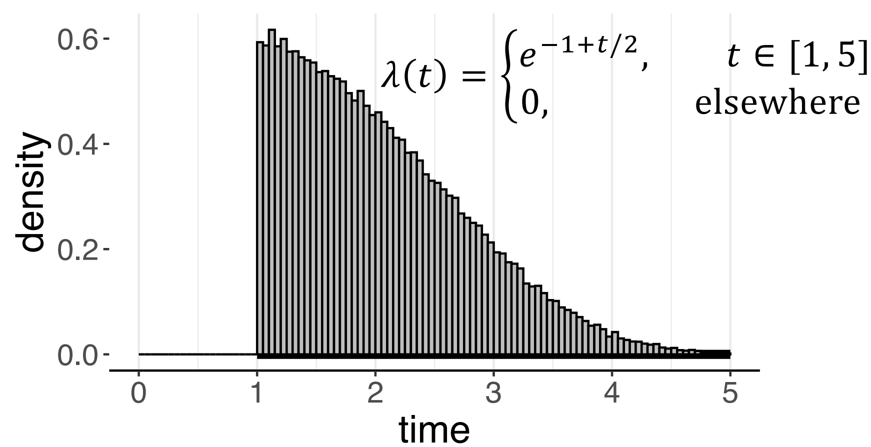
The histogram of the earliest event times does not approach the shape of the intensity function

All events vs earliest event, different example

All events, 100K instantiations



Earliest event, 100K instantiations



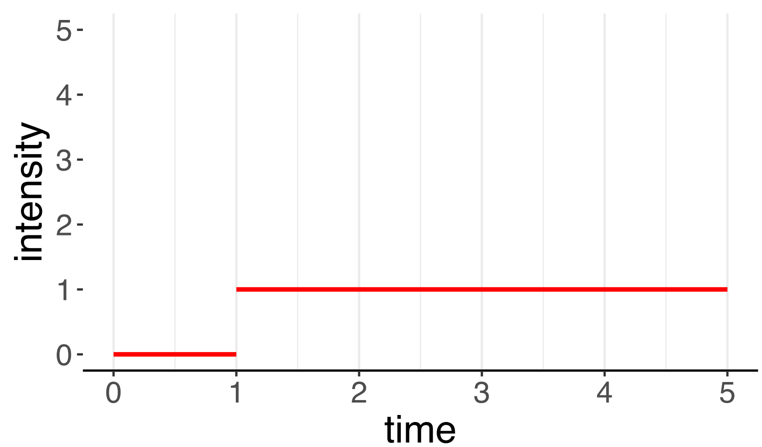
The three important functions

- Intensity function $\lambda(t)$
  - *Always available*
  - *Sufficient to sample from any NHPPP efficiently and accurately*

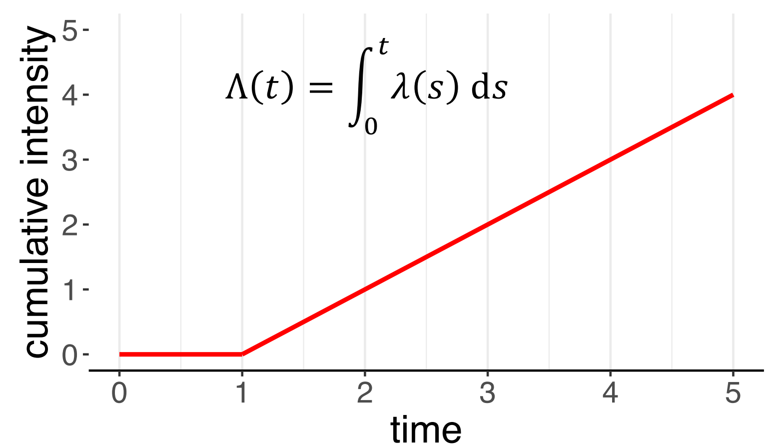
- Cumulative intensity function $\Lambda(t) = \int_0^t \lambda(s)\, \mathrm{d}s$
- Inverse cumulative intensity function $\Lambda^{-1}(z)$, defined so that $\Lambda^{-1}(\Lambda(t)) = t$
  - *Not always available*
  - *If available, you accelerate sampling by several times*

Intensity and cumulative intensity functions

Intensity function $\lambda(t)$

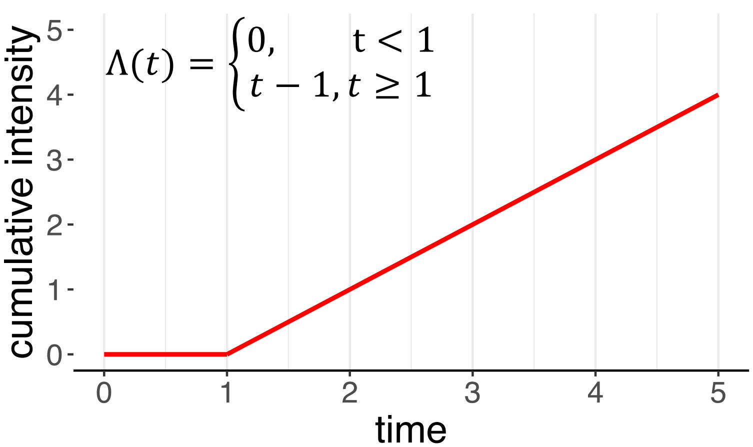


Cumulative intensity function $\Lambda(\mathrm{t})$

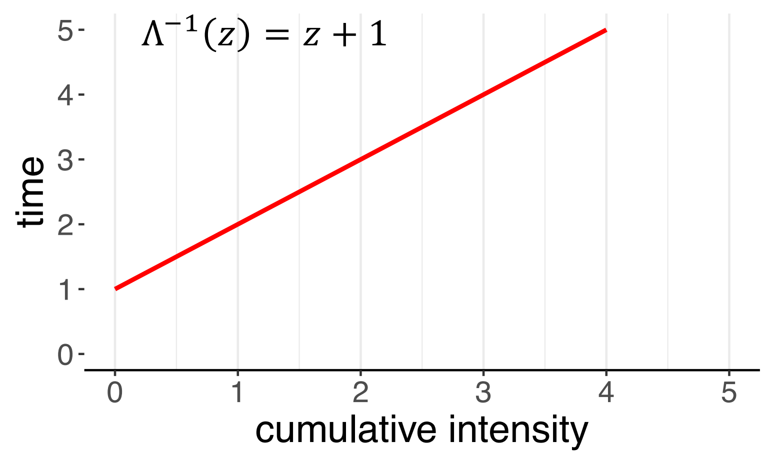


Cumulative intensity function and its inverse

Cumulative intensity function $\Lambda(\mathrm{t})$



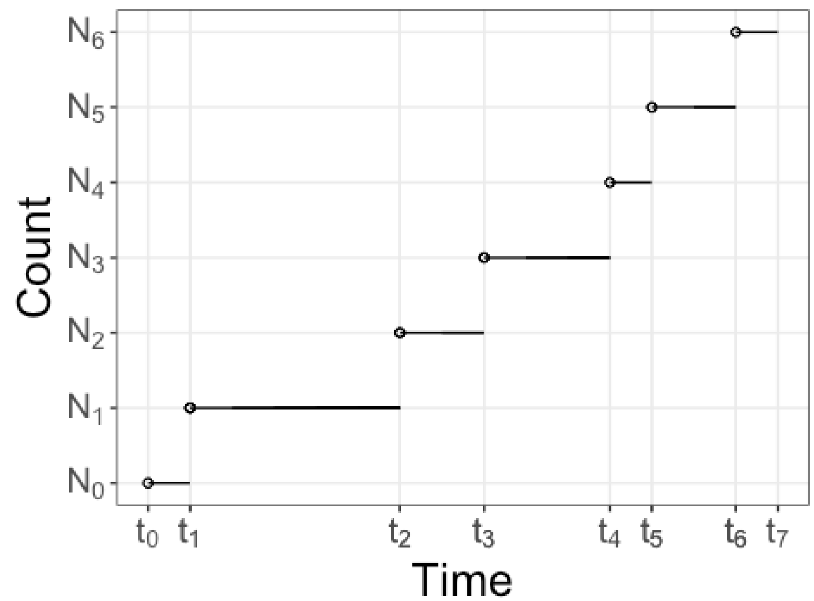
Inverse cumulative intensity function $\Lambda^{-1}(\mathrm{z})$



Duality with the Poisson counting process

Poisson *counting process*

$N_0, N_1, \ldots$
Cumulative number events over time



Poisson *point process*

$t_0, t_1, \ldots$

Outline

- Discrete event simulation as a convolution of point processes
- Non-homogeneous Poisson point processes (NHPPPs)
- Sampling from NHPPPs
- Demonstration

Three important properties for sampling

Memorylessness

You can ignore what happens outside your interval

Composability

You can merge two NHPPPs with intensities λ_1, λ_2 to get a new NHPPP with intensity $\lambda_1 + \lambda_2$.

Transmutability (time warping)

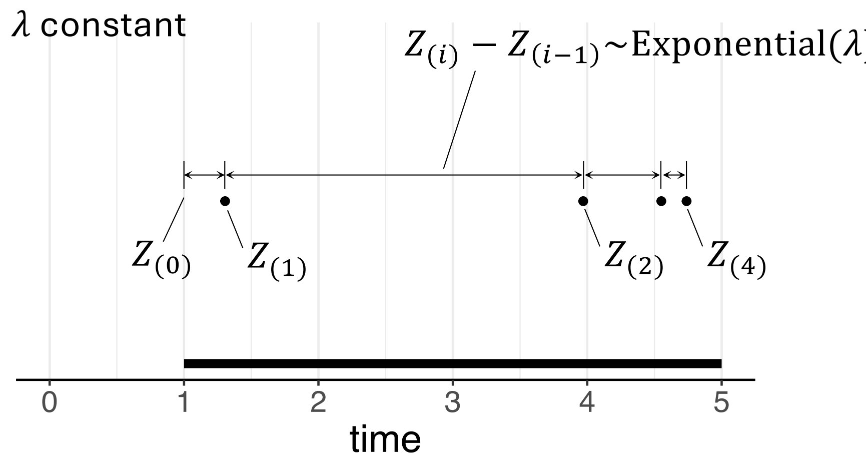
Any one-to-one transformation of the intensity function results in a unique NHPPP in the transformed time axis

1. Sampling from a PPP is easy

Constant intensity function (homogeneous PPP)

Sampling from a constant intensity function is easy.

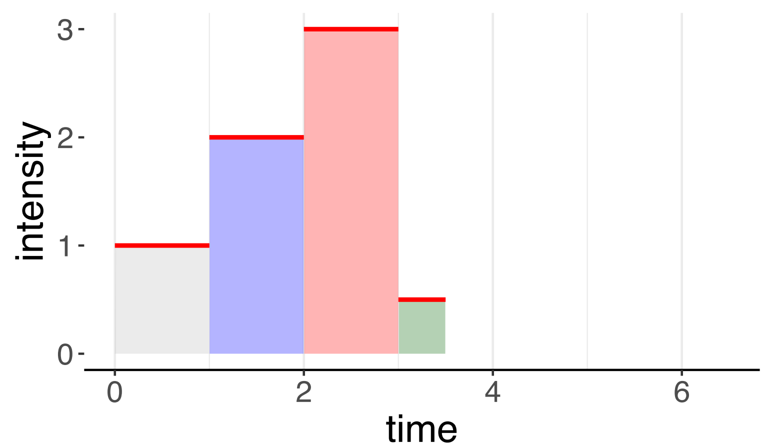
The interarrival times have an exponential distribution.



2. Memorylessness: Sampling from piecewise constant NHPPP is peasy

Piecewise constant intensity function (NHPPP)

- Look at each piecewise constant interval separately
- In each interval you have a constant intensity (easy)
- Return the union of all events

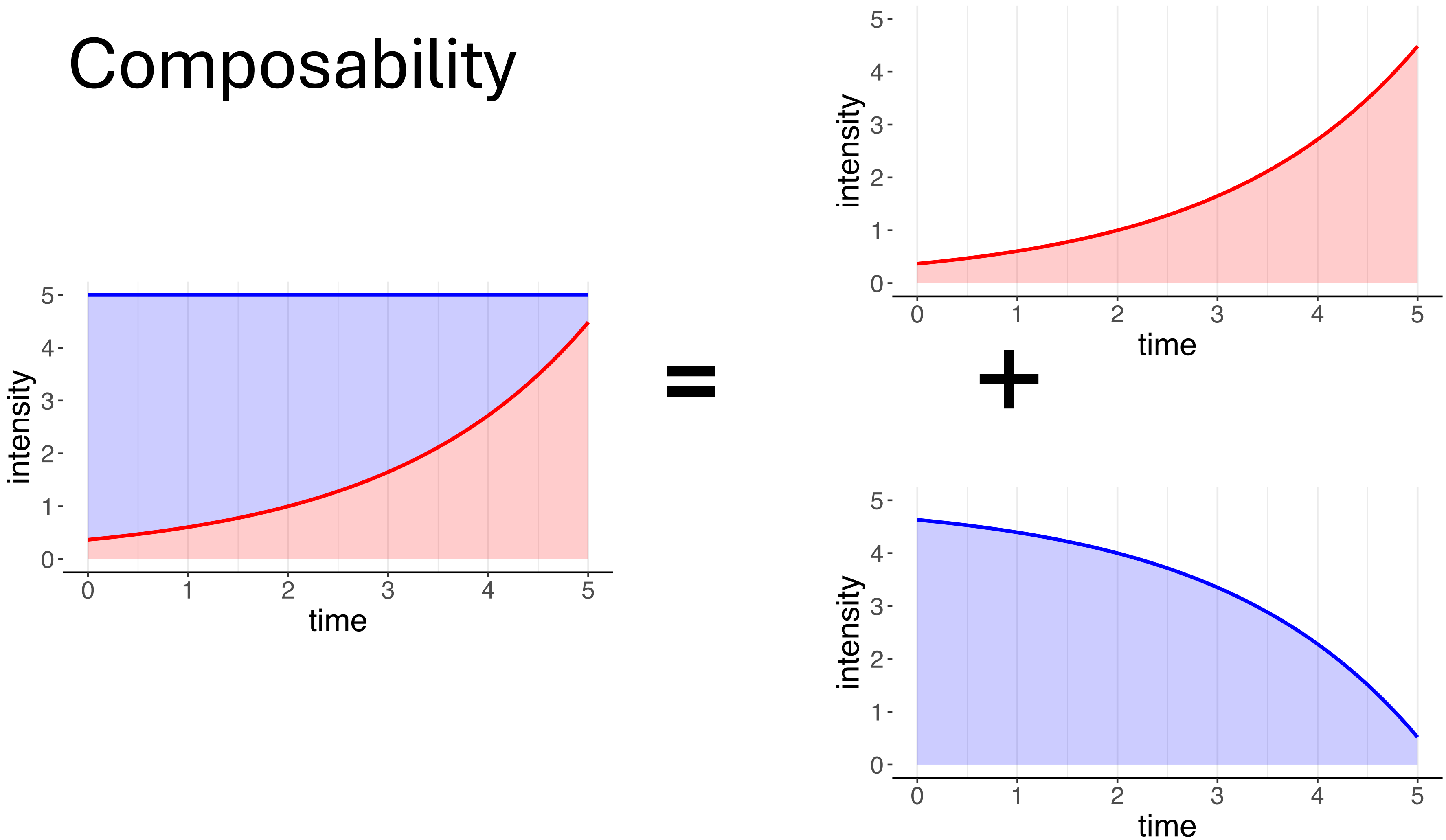


Sampling from piecewise constant intensities is easy (**memorylessness**)

3. Composability: Sampling NHPPPs when you know $\lambda(t)$ reduces to sampling from a PPP (#1) or piecewise constant NHPPP (#2)*

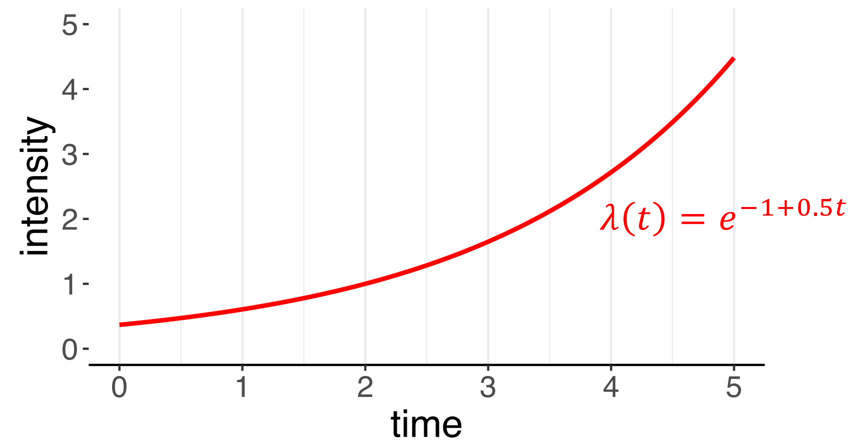
** You still need to find a constant or piecewise constant majorizer $\lambda_*(t)$, whose choice determines your efficiency .*
You cannot get achieve something difficult with zero effort.
You will put in some work.
Other terms and conditions may apply.

Composability



NHPPP, where you know $\lambda(t)$: Thinning

The general case is more challenging



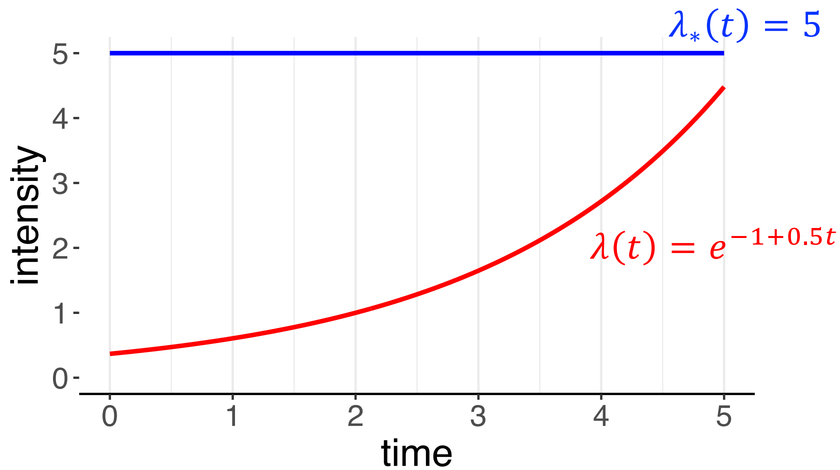
NHPPP, where you know $\lambda(t)$: Thinning

- Find a majorizer function λ_* that's easy to sample

Majorizer: any function that is "taller" that λ

$$\lambda_* \geq \lambda$$

(and has the same support as λ)



NHPPP, where you know $\lambda(t)$: Thinning

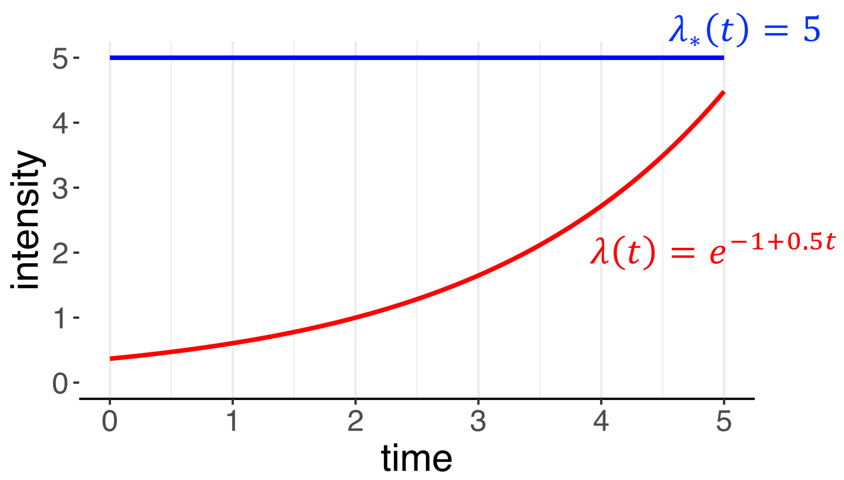
- Find a majorizer function λ_* that's easy to sample

$$\lambda_*(t) = \lambda(t) + [\lambda_*(t) - \lambda(t)]$$

Sample proposals from here

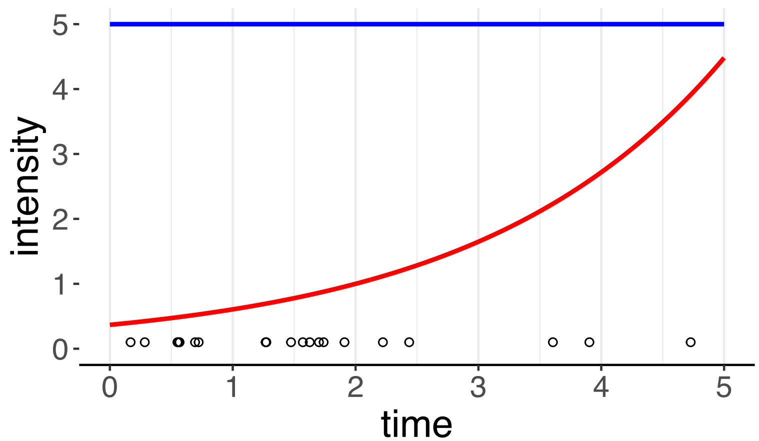
Keep proposals conforming to this part

Reject the rest



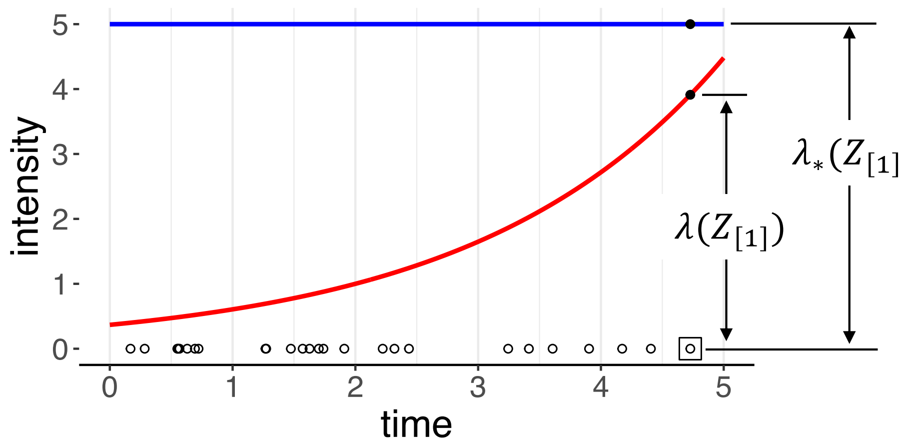
NHPPP, where you know $\lambda(t)$: Thinning

- Find a majorizer function λ_* that's easy to sample
- Draw events $\{Z_{*1}, \dots\}$ from λ_*



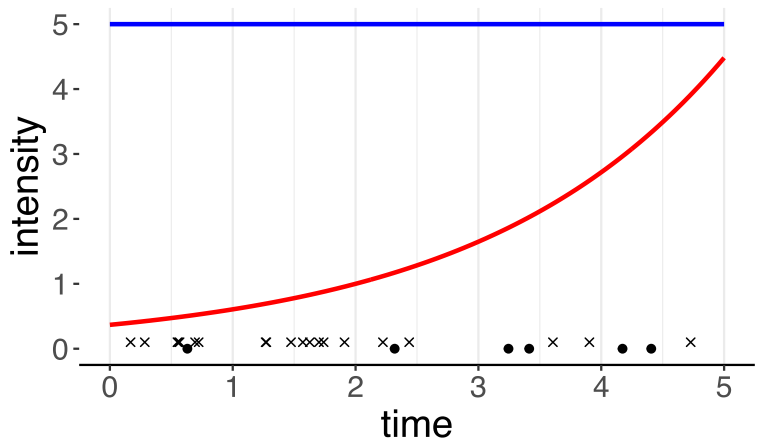
NHPPP, where you know $\lambda(t)$: Thinning

- Find a majorizer function λ_* that's easy to sample
- Draw events $\{Z_1, \ldots\}$ from λ_*
- Accept event i with probability $\frac{\lambda(Z_i)}{\lambda_*(Z_i)}$



NHPPP, where you know $\lambda(t)$: Thinning

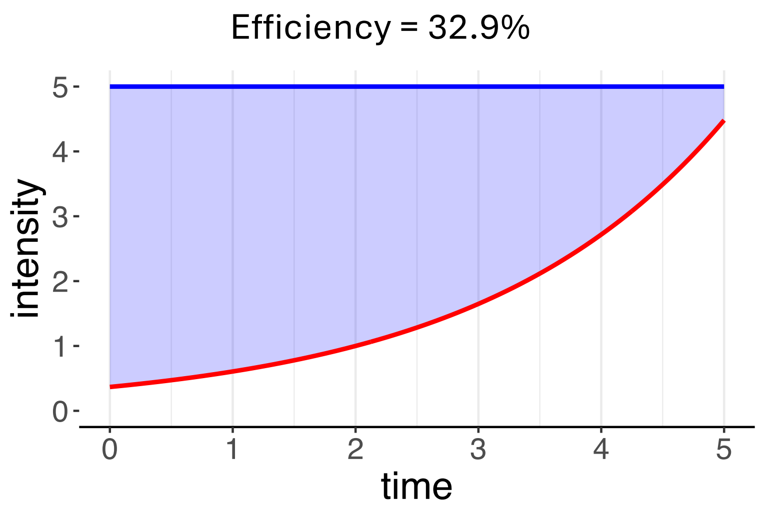
- Find a majorizer function λ_* that's easy to sample
- Draw events $\{Z_{*1}, \ldots\}$ from λ_*
- Accept event i with probability $\frac{\lambda(Z_i)}{\lambda_*(Z_i)}$
- The set of accepted points is an instantiation from $\lambda(t)$



(composability)

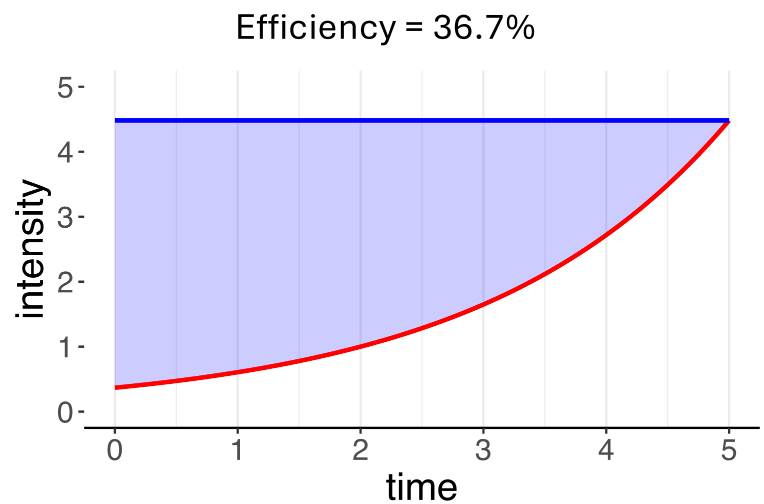
Thinning, efficiency

- Thinning efficiency: average fraction of proposals that are accepted
- Depends on the choice of λ_*
- The smaller the blue area, the better the efficiency



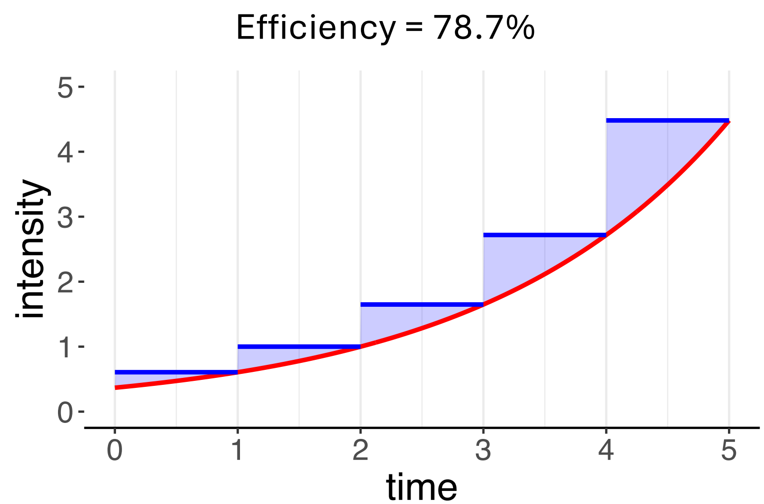
Thinning, efficiency

- Thinning efficiency: average fraction of proposals that are accepted
- Depends on the choice of λ_*
- The smaller the blue area, the better the efficiency



Thinning, efficiency

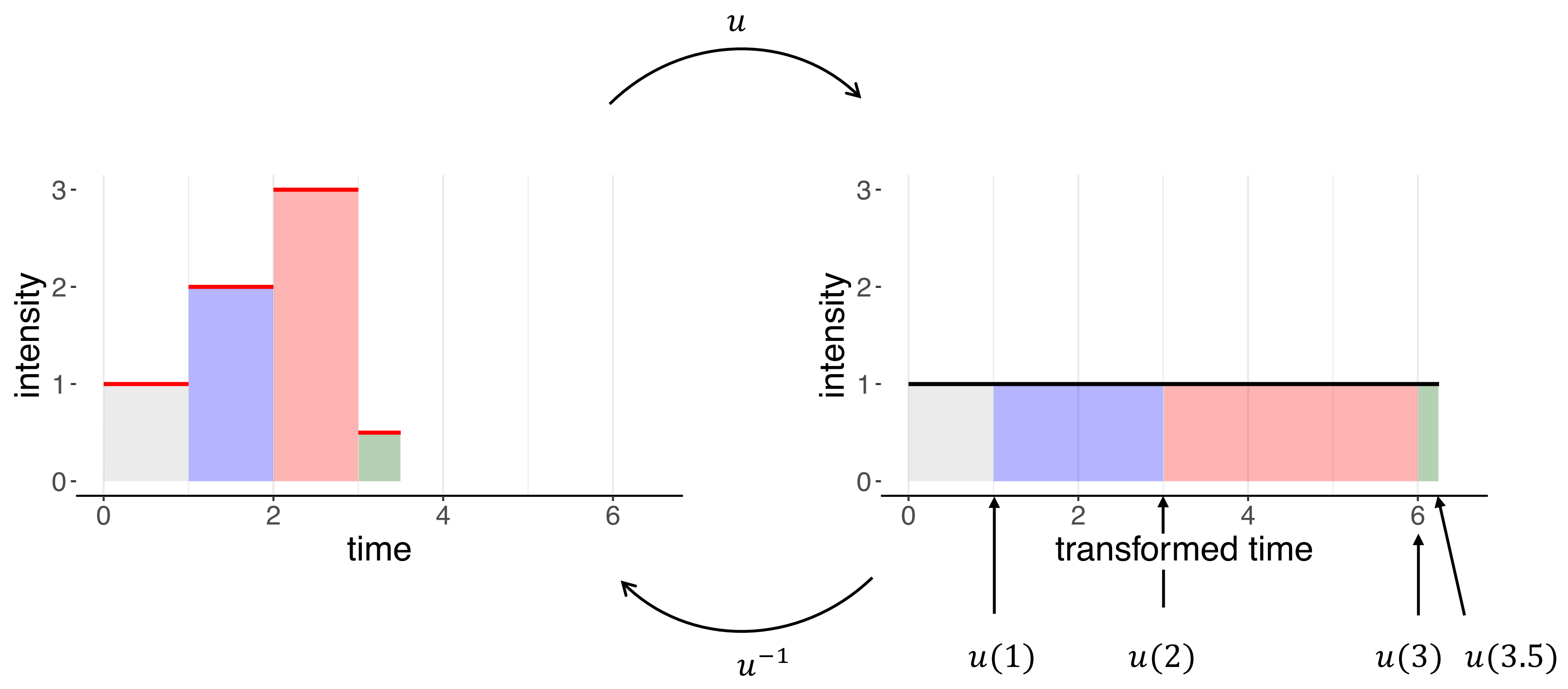
- Thinning efficiency: average fraction of proposals that are accepted
- Depends on the choice of λ_*
- The smaller the blue area, the better the efficiency



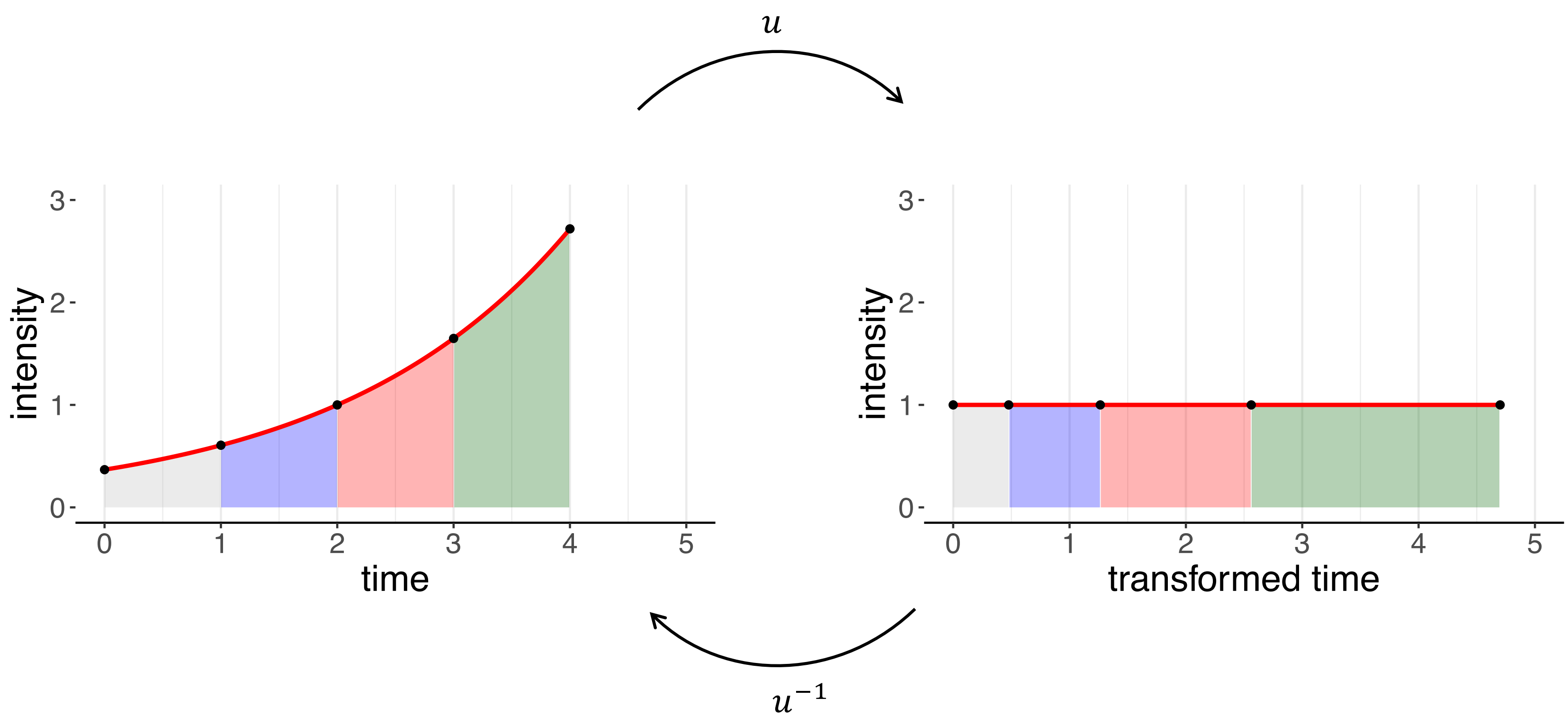
4. Transmutability of time: Sampling NHPPPs when you know Λ, Λ^{-1} reduces to sampling from a PPP with rate one (#1) *

** You will need to do some maths to get Λ, Λ^{-1}. It may not be practical to do so, or even possible. In such a case, back to (#3). Even if you have Λ, you may not have a cheap Λ^{-1}.*
You cannot achieve something difficult with zero effort. You will put in some work. Other terms and conditions may apply.

Transmutability



Transmutability



A nice u is Λ (and then u^{-1} is Λ^{-1})

Change of variable from s to u

$$\Lambda(t) = \int_a^t \lambda(s)\,\mathrm{d}s = \int_{u(a)}^{u(t)} \frac{\lambda(s)}{u'(s)}\,\mathrm{d}u$$

Pick u so that $u' = \lambda$. Any antiderivative of λ works. Using $u \coloneqq \Lambda$, transforms time to scale where the process has constant rate 1,

$$\int_{\Lambda(a)}^{\Lambda(t)} \frac{\lambda(s)}{\Lambda'(s)}\,\mathrm{d}u = \int_{\Lambda(a)}^{\Lambda(t)} 1\,\mathrm{d}u\,.$$

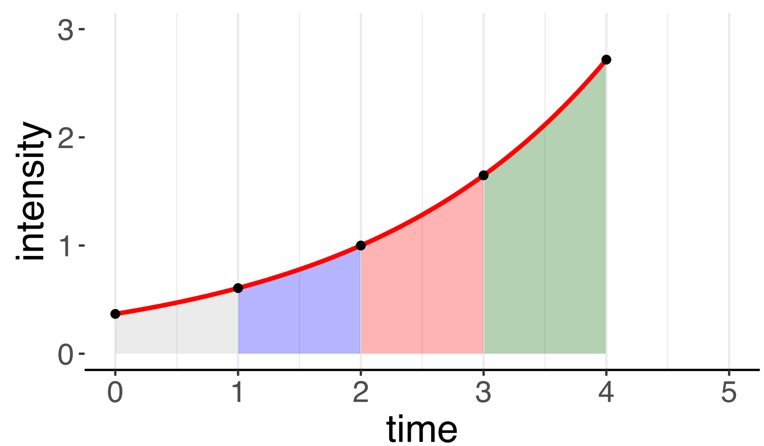
This is a sketch of the formal proof – omitting the rigorous bits

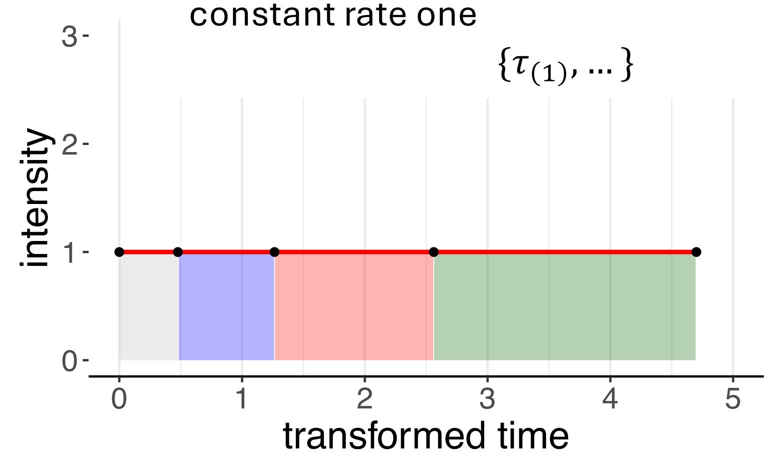
Transmutability

1. Find the start and stop of the transformed time interval
$\tau_{start} = \Lambda(t_{start})$ and $\tau_{stop} = \Lambda(t_{stop})$

Λ

2. Sample transformed times from a PPP with constant rate one
$\{\tau_{(1)}, \dots\}$





Λ^{-1}

3. Back-transform the instantiation to the original time scale
$\{\Lambda^{-1}(\tau_{(1)}), \dots\}$

More in these works…

PLOS ONE

RESEARCH ARTICLE

The nhppp package for simulating non-homogeneous Poisson point processes in R

Thomas A. Trikalinos[1,2,3]*, Yuliia Sereda[1]

Original Research Article

MDM
Medical Decision Making

Medical Decision Making
2025, Vol. 45(2) 205–213

DOI: 10.1177/0272989X241308768
journals.sagepub.com/home/mdm
Sage

A Fast Nonparametric Sampling Method for Time to Event in Individual-Level Simulation Models

David U. Garibay-Treviño, Hawre Jalal, and Fernando Alarid-Escudero

Outline

- Discrete event simulation as a convolution of point processes
- Non-homogeneous Poisson point processes (NHPPPs)
- Sampling from NHPPPs
- Demonstration

`nhppp` from CRAN

- Fast vectorized implementations of the presented algorithms
- Wrapper f

Define a vectorized λ and piecewise constant majorizer λ_*

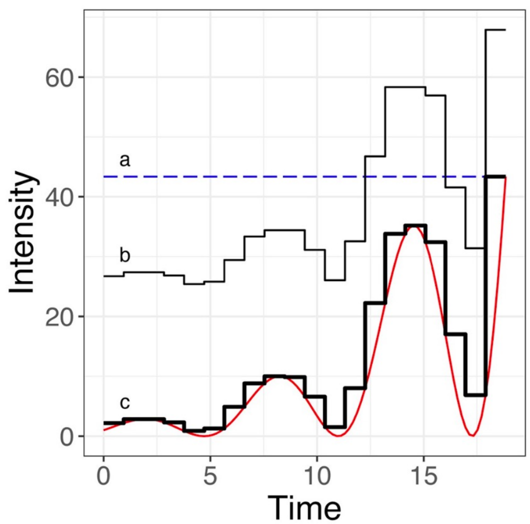
Simulation window varies by person, time on the simulation clock

Select black, red, or blue process

```r
an_intensity_fun <- function(t) { ... }
majorizer_matrix <- matrix( ... )

vdraw(
  lambda = an_intensity_function,
  lambda_maj_matrix = majorizer_matrix,
  t_min = rep(40, N),
  t_max = 40 + runif(N, 1, 60),
  atmost1 = TRUE,
  atleast1 = TRUE
)
```

nhppp simulates correctly



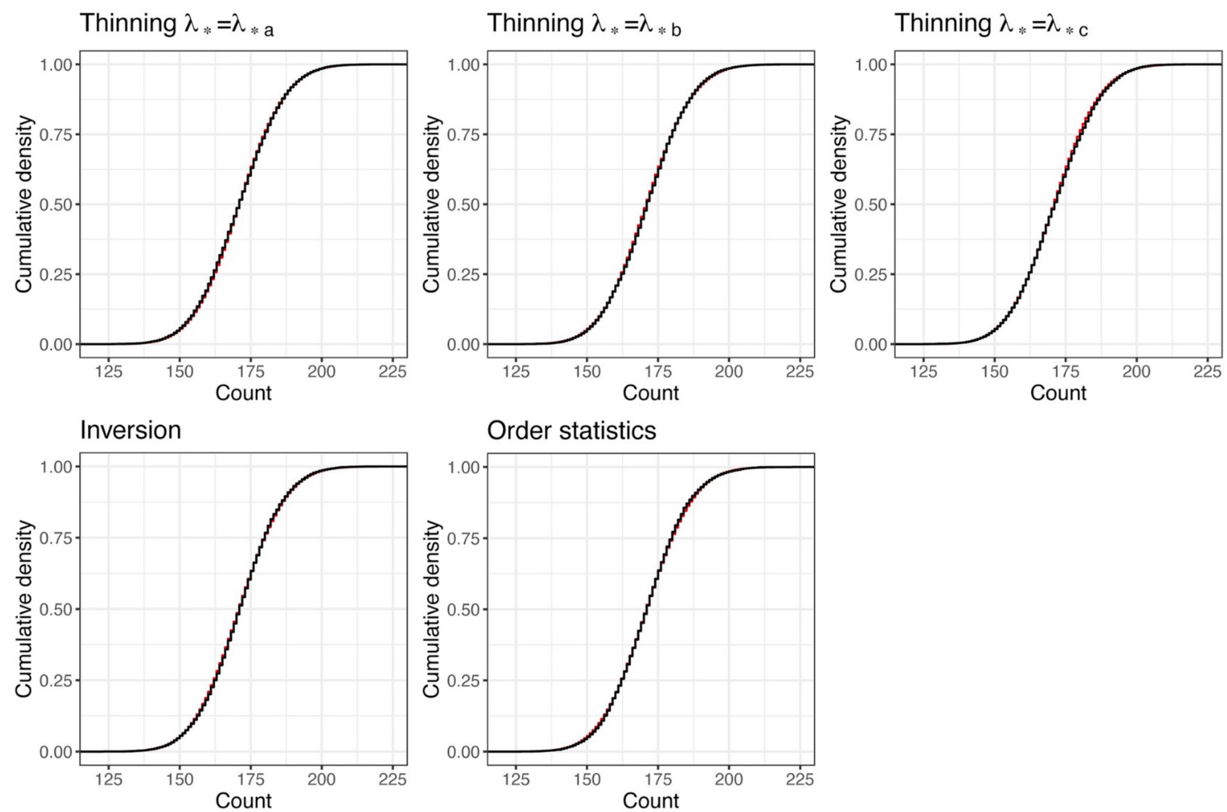
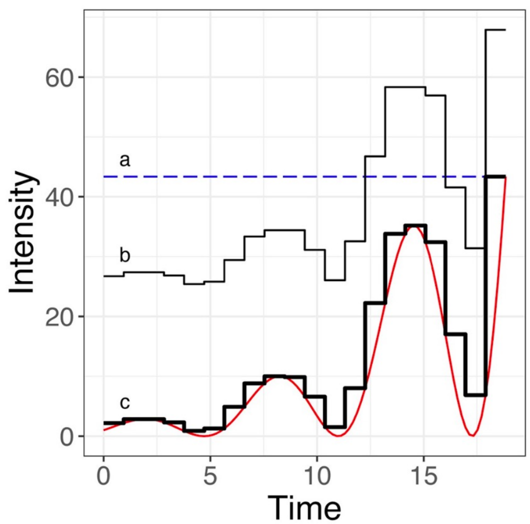


Fig 2. Theoretical (red) and empirical (black) cumulative distribution functions for event counts in the illustration example with nhppp functions. The unsigned area between the theoretical and empirical curves equals the Wasserstein-1 distance in Table 5.

PLoS ONE 2024

…but other packages do not



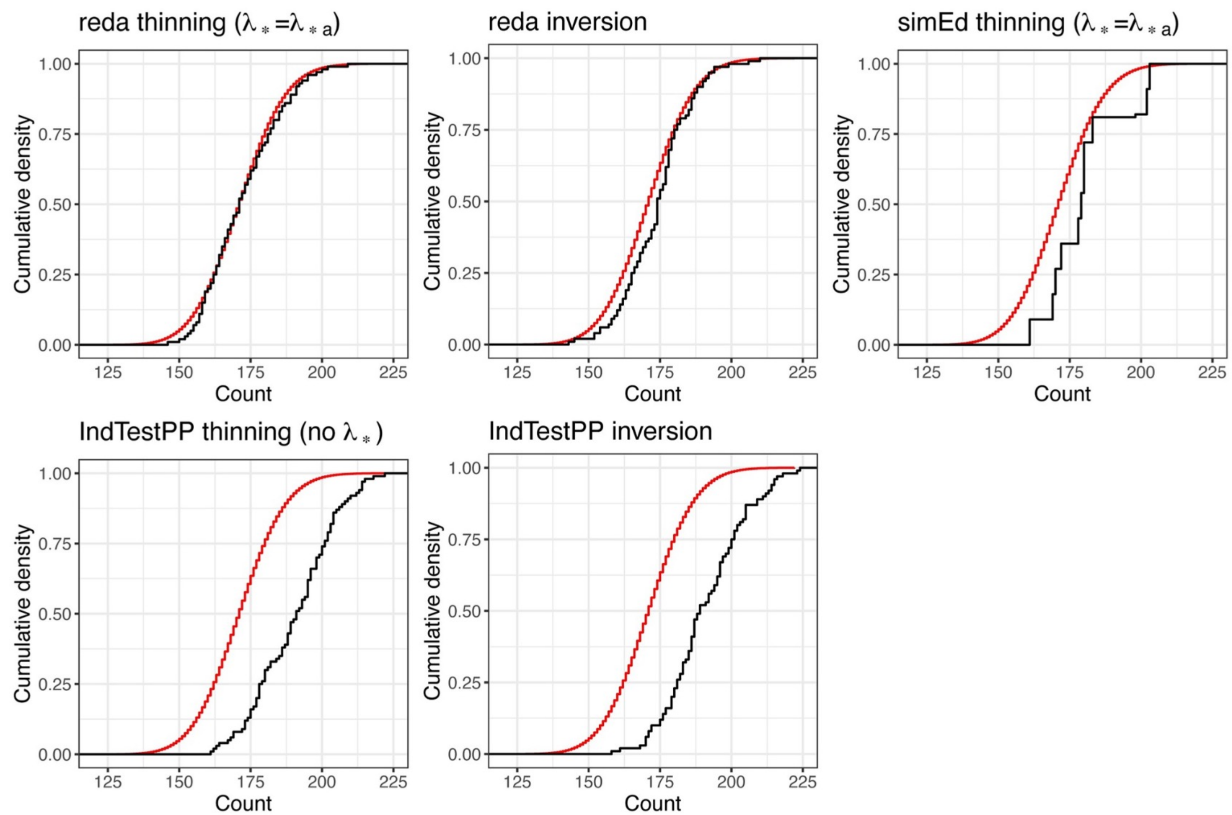


Fig 3. Theoretical (red) and empirical (black) cumulative distribution functions for event counts in the illustration example with the R packages in Table 3. The unsigned area between the theoretical and empirical curves equals the Wasserstein-1 distance in Table 5.

PLoS ONE 2024

... and are slower

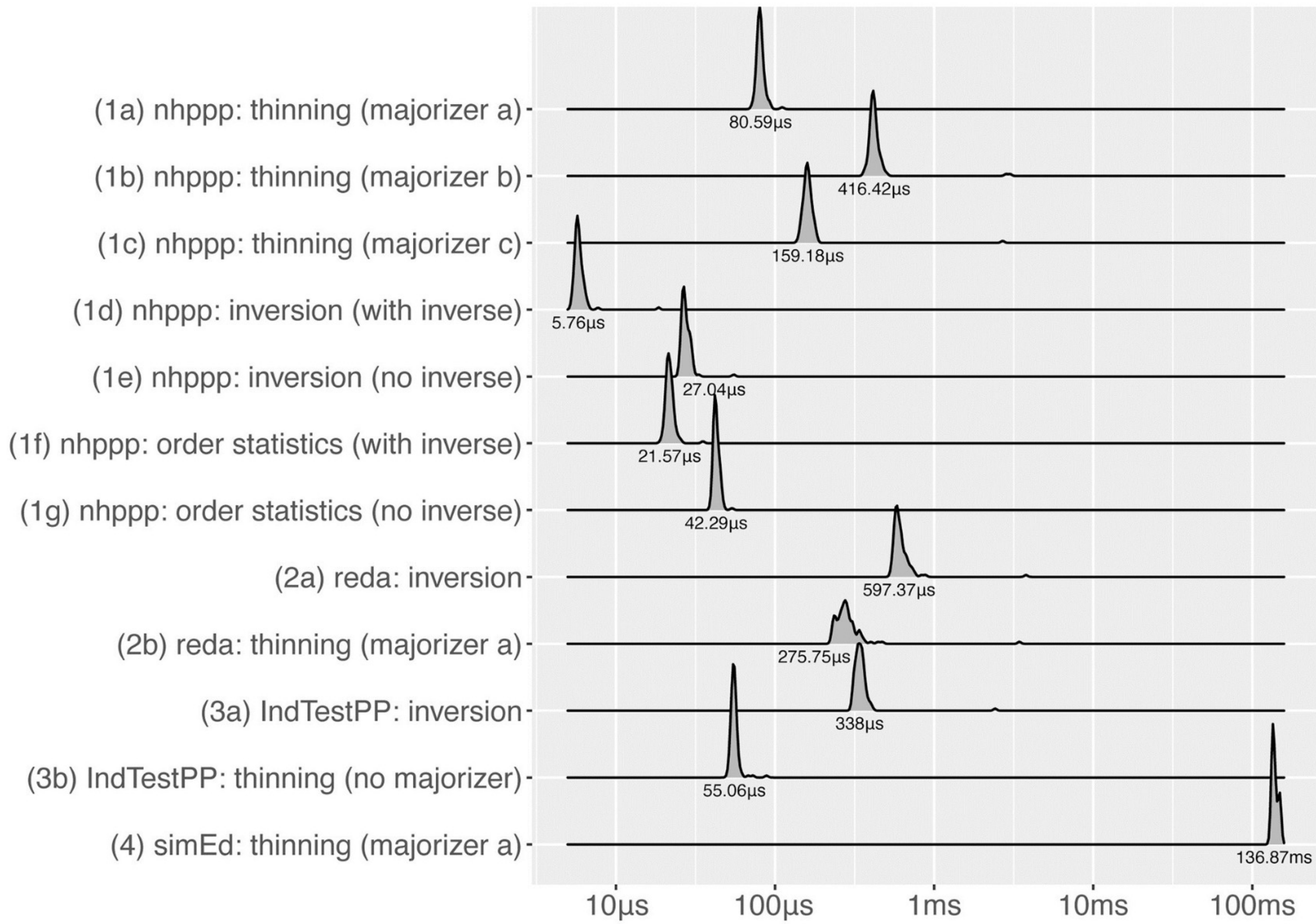
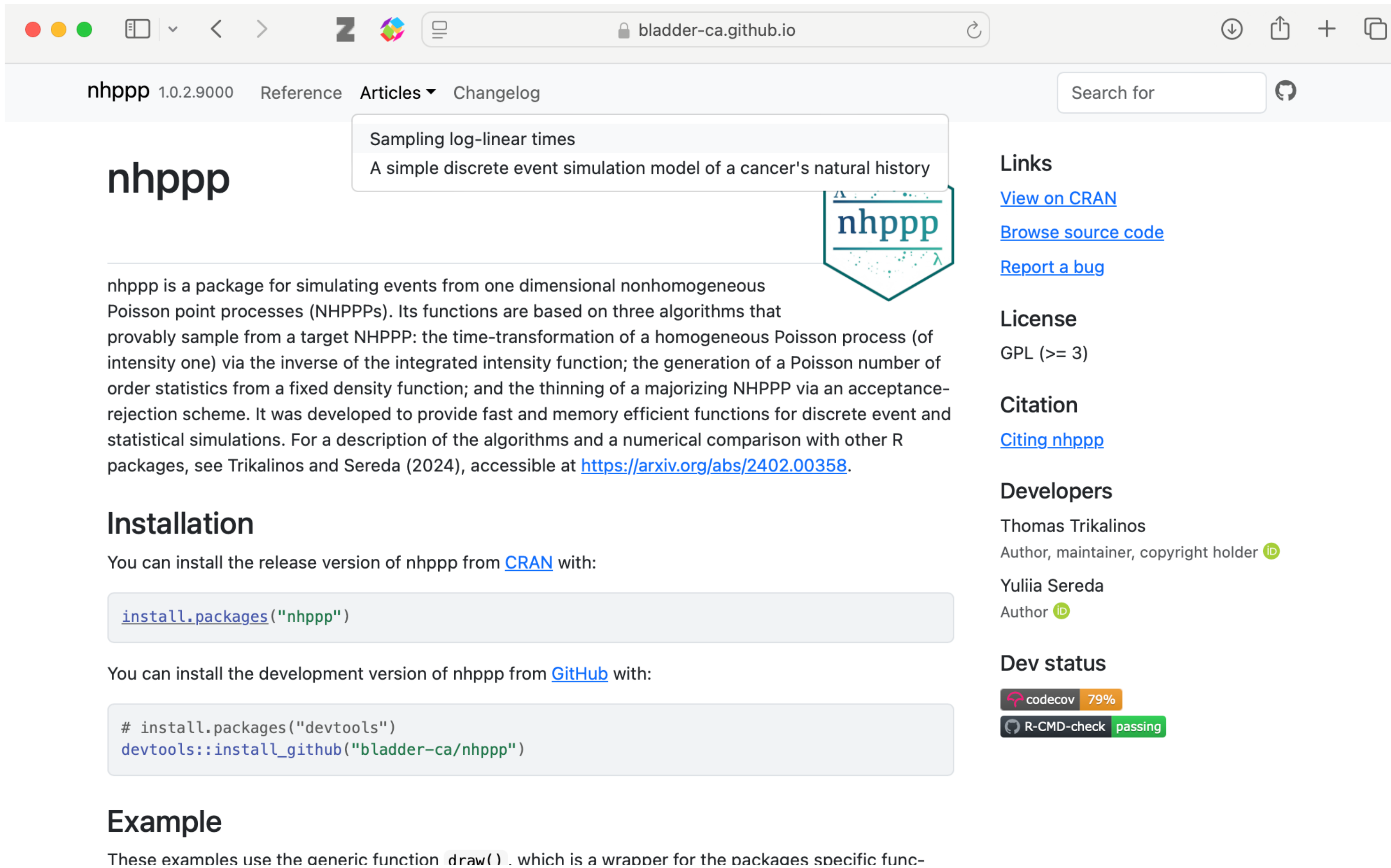


Fig 7. Computation times when drawing the first event in interval.

PLoS ONE 2024

See vignettes at package site



https://bladder-ca.github.io/nhppp/index.html

All materials on a public GitHub repository

All expository materials and example code are available at

`https://github.com/ttrikalin/des-R-course`

(choose the `2025_cisnet` release for today's talk)

Join us at the 47th Annual SMDM in Michigan

Meetings/Events   Networking   Education/Career Tools

UPCOMING MEETINGS / SMDM 47TH ANNUAL MEETING

SMDM 47th Annual Meeting

June 15 - June 18, 2025

University of Michigan, Ann Arbor, Michigan, USA

Meeting Co-Chairs:
David W. Hutton, PhD
Brian J. Zikmund-Fisher, PhD

Short Courses

AM Short Courses, In-Person

15 June 2025; 9:00 AM - 12:30 PM (local time)

Act the expert! Improvisational games to boost your scientific presentation skills
Brian Zikmund-Fisher, PhD
Brittany Batell, MPH, MSW, CHES
Daniel Matlock, MD, MPH

An Introduction to Research Prioritization and Study Design Using Value of Information Analysis
Fernando Alarid-Escudero, PhD
Jeremy Goldhaber-Fiebert, PhD
Hawre Jalal, PhD
Natalia Kunst, MD, PhD

Advanced Discrete - Event Simulations in R
Thomas A. Trikalinos, MD
Fernando Alarid-Escudero, PhD
Yuliia Sereda, PhD
Stavroula A. Chrysanthopoulou, PhD

Causal Diagrams, Target Trial Emulation and Causal Inference for Modeling and Medical Decision Making
Uwe Siebert, MD, MPH, MSc, ScD

Introduction to Reproducible Programming and Project Management
Jacob Jameson, MS
Madison Coots, MS