Simulating Poisson processes

University of Toronto, ACT 350

Prof. S. Pesenti

08/09/2020

How to simulate a Poisson process

Recall that a Poisson process with rate $\lambda > 0$ is a counting process $\{N(t),\ t \geq 0\}$ that has the following properties

1. *starts at 0*: $N(0) = 0$ with probability 1
2. *is increasing*: $N(s) \leq N(t)$ for all $s \leq t$
3. *has Poisson distributed increments*: $P(N(s+t) - N(s) = n) \sim \text{Pois}(\lambda t)$ for all $s, t \geq 0$.

However, this definition does not help if we want to simulate a Poisson process. Thus, to simulate a Poisson process we use its alternative definition. That is, for independent random variables $X_1, X_2, \dots$ that are Exponentially distributed with rate $\lambda > 0$, a Poisson process with rate $\lambda > 0$ can be written as

$$N(t) = \max\left\{n \in \mathbb{N} \;\middle|\; \sum_{i=1}^{n} X_i \leq t\right\}, \quad \text{for all } t \geq 0.$$

Thus, for simulating a Poisson process we first need to simulate exponential random variables, add them up and check whether they are smaller or equal to t. However, we will run into problems when we want to store the value of $N(t)$ for every $0 \leq t < \infty$. The way out is to implement $N(t)$ as a function of t in the following way. First generate the jump points of the Poisson process. Second, we know that a Poisson process can only jump one unit at a time. Thus, we can define the Poisson process as a step function that jumps exactly one unit at every jump point.

1. Generating jump times

This is an algorithm for generating the jump times of a Poisson process

1. Set $\tau_0 = 0$.
2. For $k \geq 1$ do
   a) generate $X \sim$Exp(λ)
   b) set $\tau_k = \tau_{k-1} + X$

The $\tau_k,\ k = 0, 1, \ldots$ are the time points, where the Poisson process jumps.

Numerical implementation: R code

```
# set the seed for generating random variables (for reproducability)
# set.seed(2019)

# set the rate of the Poisson process/Exponential distribution
lambda <- 2
# number of jumps
n <- 50
# initialise the jump times
jump_times <- rep(0, lenthought = n)

# note that the first value of the jump_times is aleady 0
for(i in 2:n){
  jump_times[i] <- jump_times[i - 1] + rexp(1, lambda)
}
jump_times
```

```
##  [1]  0.00000000  0.08392964  0.40169155  0.69439969  0.86486407  0.95194932
##  [7]  1.28498569  1.45903169  2.05518462  2.38224677  2.92635016  3.06556691
## [13]  4.61114863  5.18812490  6.01274892  6.79459443  8.80242283 10.00229765
## [19] 10.62725970 10.85193395 11.12956236 12.29444269 12.35255619 12.83112242
## [25] 13.52349274 14.05565723 14.83834796 15.12041067 15.20725859 15.61810617
## [31] 15.87047088 15.91764105 16.13303110 16.17630992 16.44527679 16.99007003
## [37] 17.50494681 18.24211092 18.88526868 18.97270443 19.05522498 19.72987822
## [43] 19.85945608 21.41060795 22.31358469 22.86029328 22.91795597 23.34722470
## [49] 23.59448295 24.00388081
```

A more efficient implementation, avoiding the for loop, is

```
jump_times_2 = cumsum(rexp(n-1, lambda))
# note that the two generated sample_paths are different,
# since the Exponential random variables are different.
jump_times_2
```

```
##  [1]  0.1458827  0.4775990  0.6205391  3.5627986  3.9839791  4.2751437
##  [7]  4.8299963  4.8538748  5.4361382  5.8027378  7.2884842  7.4042789
## [13]  7.6468467  7.7344969  8.0282881  8.8381585  9.0054665  9.0717844
## [19]  9.8012940 10.0508849 10.5103700 10.5135410 11.3223250 11.3289122
## [25] 11.3613650 11.4691510 11.5477209 12.6287205 12.9800112 13.1558413
## [31] 14.4728279 15.3039947 15.4729084 15.9731209 16.0664084 17.0613587
## [37] 17.2257266 17.5341449 18.4087557 19.2998072 19.6211126 20.9096770
## [43] 24.3958251 24.7768882 25.2472920 26.3649781 27.2638825 27.5002652
## [49] 27.6224412
```

2. Defining the Poisson process as a function

Next, we can define the Poisson process as a step function that jumps exactly one unit at every value in `jump_times`. Note that the implemented Poisson process is now a *function* of the time t.

```r
# define the Poisson process as a function
poisson_process <- stepfun(jump_times[2 : 50], seq(0, n - 1, by = 1))
# the first argument of `stepfun` is where the step function jumps
# the second argument is the hight of the step function

# Let us check some values.
# The starting value is 0.
poisson_process(0)
```

```
## [1] 0
```

```r
# What is the value of N(3)? (Note that we use jump_times and not jump_times_2)
poisson_process(3)
```

```
## [1] 10
```

```r
poisson_process(10)
```

```
## [1] 16
```

```r
poisson_process(15)
```

```
## [1] 26
```

The first jump time is 0.0839296. Let us check whether the Poisson process jumps at that time.

```r
# The first jump of the Poisson process is
first_jump <- jump_times[2]
poisson_process(first_jump - 0.001)
```

```
## [1] 0
```

```r
poisson_process(first_jump + 0.001)
```

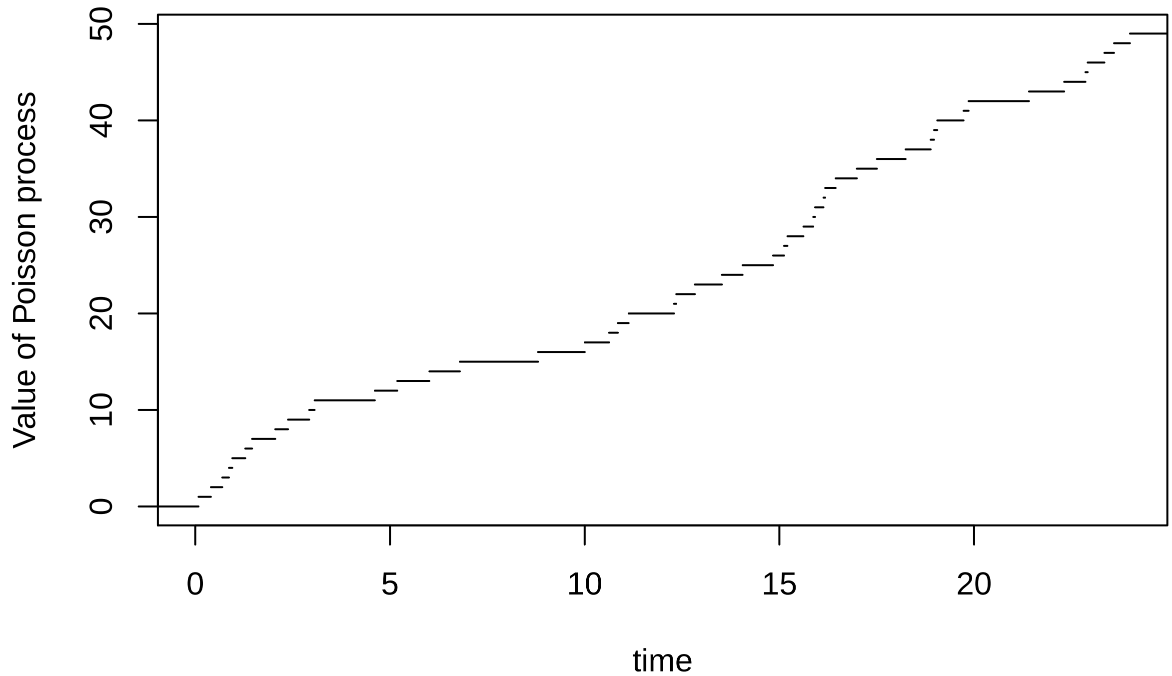
```
## [1] 1
```

Note you can evaluate the Poisson process for any time point t, however, keep in mind that we only simulated 50 jumps of the Poisson process. Thus, evaluating `N(t)` for t large does not make sense.

3. Plotting a sample path

To plot a sample path we have to plot the implemented step function `poisson_process`.

```r
plot(poisson_process, xlab = "time", ylab = "Value of Poisson process", main = NULL,
     verticals = FALSE, do.points = FALSE, xlim = c(0,jump_times[n]))
```



Question: Rerun the code, what do you observe?
Question: Rerun the code with a different λ or change the number of jumps. How does the sample path change?

Renewal Theory

A renewal process is defined as follows: Consider a sequence $T_1, T_2, \ldots$ of independent identically distributed (i.i.d.) non-negative random variables. Define the stochastic process

$$X_0 = 0\,, \quad X_n = T_1 + \cdots + T_n\,, \quad n = 1, 2, \ldots\,,$$

and set

$$N(t) = \max\left\{n \in \mathbb{N} \;\middle|\; \sum_{i=1}^{n} X_i \leq t\right\}, \quad \text{for all } t \geq 0.$$

Then the process $\{N(t) \,|\, t \geq 0\}$ is called a renewal process.

If the $T_i\,, i = 1, 2, \ldots$ are i.i.d. Exponential with parameter $\lambda > 0$, then the renewal process $\{N(t) \,|\, t \geq 0\}$ is a Poisson process.

Task:

a) Generate a code to simulate from a renewal process, where the interarrival times $T_i\,, i = 1, 2, \ldots$ are i.i.d. LogNormal distributed with mean 2 and standard deviation 0.5. *Hint: use the function `rlnorm`.*

b) Plot a sample path of the renewal process define in **a)**.

c) What do you observe compared to the Poisson process?

d) Repeat tasks **a)** to **b)** with interarrival times being Gamma distributed with the same mean and standard deviation as in **a)**. *Hint: use the function* `rgamma`.

e) Compare the Poisson process with the renewal process from **a)** and the renewal process from **d)** and give interpretations.