Simulating Poisson processes

University of Toronto, ACT 350

Prof. S. Pesenti

28/09/2021

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 already 0
for(i in 2:n){
  jump_times[i] <- jump_times[i - 1] + rexp(1, lambda)
}
jump_times
```

```
##  [1]  0.0000000  0.5283330  0.6892461  0.8721776  1.6622546  2.2599094
##  [7]  3.1880280  3.3157412  3.7241735  4.0765470  4.2761498  4.8747909
## [13]  5.0808755  5.1114063  5.1802213  5.7022319  6.0902246  6.2145804
## [19]  6.2878369  6.4854844  7.0689802  7.3378867  7.6447598  7.8465258
## [25]  8.7324164  9.1746606 10.4995833 12.5449682 12.9256397 13.0845084
## [31] 13.2027416 13.4091018 13.4797689 13.6025627 14.2873237 14.3352420
## [37] 15.5002634 15.5541356 15.5912774 15.9381172 16.3992362 17.5884432
## [43] 17.6611177 17.6652973 17.9113664 19.5077417 19.6198261 19.8558578
## [49] 19.9524658 20.2095705
```

A more efficient implementation, avoiding the for loop, is

```
jump_times_2 = c(0, 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]  0.00000000  0.04437553  0.29264023  0.41535901  0.73854297  1.69544716
##  [7]  3.29740583  3.65826386  5.60734062  5.94797994  7.06702419  8.39602880
## [13]  8.71882871 10.16757086 10.64986404 10.88384723 11.26915519 12.74946752
## [19] 12.80876763 14.01775780 14.28761950 14.45189968 14.78547787 14.86050622
## [25] 16.84001761 16.87562340 17.11827041 17.42740074 18.25976285 19.35008157
## [31] 19.37780297 19.54459554 19.59057005 19.68081168 20.08058313 20.16695809
## [37] 20.56266529 20.93641425 22.41415470 23.03350530 23.18904663 23.20929366
## [43] 23.93936112 25.11424508 25.50424153 26.04985940 26.12752702 26.21886897
## [49] 28.48605419 30.14667201
```

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.

```
# 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
```

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

```
## [1] 5
```

```
poisson_process(10)
```

```
## [1] 25
```

```
poisson_process(15)
```

```
## [1] 35
```

The first jump time is 0.528333. Let us check whether the Poisson process jumps at that time.

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

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

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

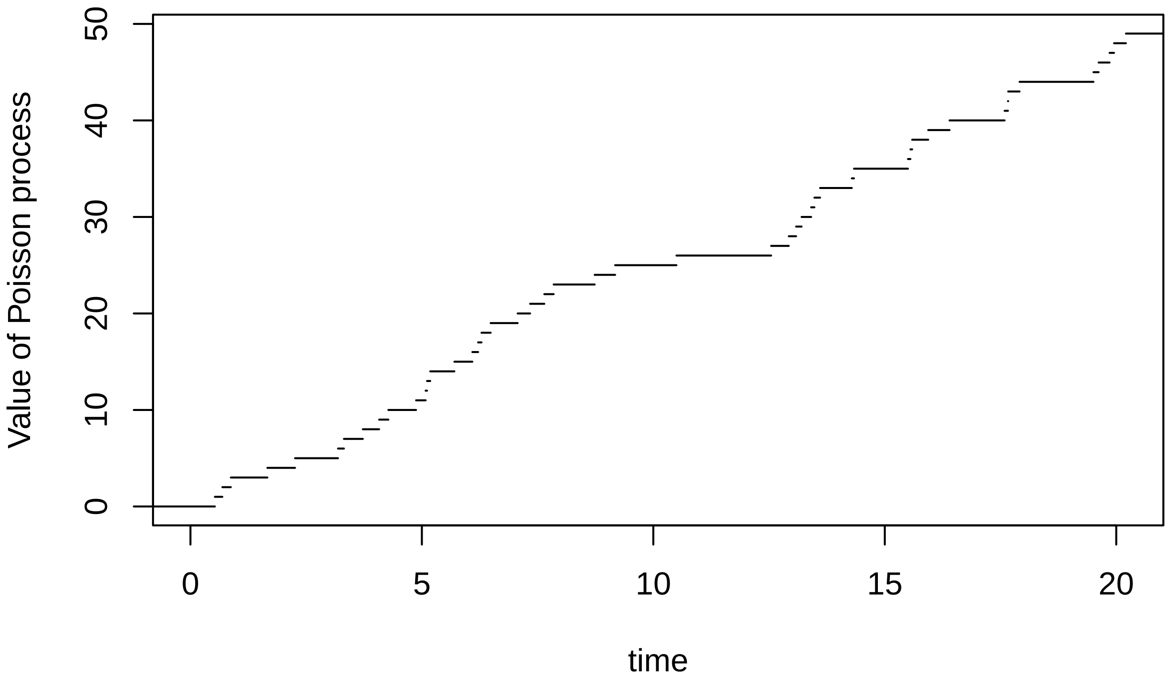
```
## [1] 1
```

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`.

```
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?

Tasks:

a) Plot multiple sample paths in one plot and provide an interpretation.

b) Plot a histogram of the Poisson process at a time t. Verify that it is indeed a Poisson random variable with parameter λ. E.g., calculate the sample mean, sample variance, compare the histograms.

b) Plot a histogram of an increment of a Poisson process and verify that it is Poisson distributed.

c) How can you show that disjoint increments are indeed independent?

Renewal Theory

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

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

and set

$$N(t) = \max\Big\{n \in \mathbb{N} \;\Big|\; \sum_{i=1}^{n} X_i \le t\Big\}, \quad \text{for all } t \ge 0.$$

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.

Tasks:

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.