Generation of Poisson distributed pseudo-random numbers

G. Palafox

September 29, 2020

Abstract

Two algorithms generating Poisson-distributed pseudo-random numbers are shown. A rigorous proof for one of them is provided, while the correctness of the other one is analyzed computationally. Lastly, an example of how the binomial distribution approaches the Poisson distribution is also given.

1 Introduction

In this work, two algorithms for generating Poisson distributed numbers are explained. The first one assumes access to an exponentially-distributed pseudo-random number generator, while the second one assumes access to a uniformly-distributed pseudo-random number generator. Graphics and elementary probability are used to support the validity of the algorithms. In the last section, an example of how the binomial distribution approaches the Poisson distribution is given. This study was performed with R version 4.0.0 [4] on a Jupyter [1] notebook[1].

2 Generating pseudo-random numbers

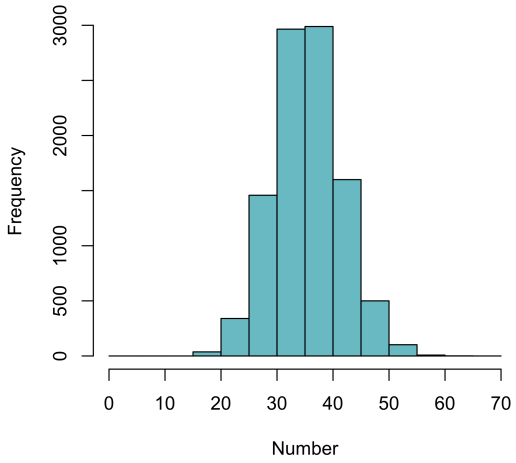
First, an algorithm which generates numbers with a Poisson distribution is shown in Algorithm 1. The idea is to generate numbers $x_i \sim \text{Exp}(\lambda)$ until $x_1 + \cdots + x_k$ first exceeds some number M. Then, we return $k-1$. This will generate numbers following a Poisson distribution with mean λM. A rigorous proof of this claim is not shown, however, the following intuitive explanation is given. In a Poisson process with rate λ, the number of events in an interval of length t is a Poisson distributed random variable with mean λt [6]. Additionally, the time between events in a Poisson process is distributed exponentially with mean $1/\lambda$. From these two facts we can see that the number of exponential variables generated with sum less than M must be the events occurring in a time interval of length M, so they must have distribution $\text{Pois}(\lambda M)$. A different, significantly more informal explanation, is that if each exponential random variable is $1/\lambda$ on average, $M\lambda$ of this variables will be needed on average for their sum to exceed M. Empirically, Figure 1 shows different comparisons between numbers generated by Algorithm 1 and numbers generated by R's `rpois` function with the mean we expected to see. Furthermore, R's function `MASS::fitdistr` supported our conclusions, as is seen in Table 1.

Algorithm 1 Poisson numbers from exponentials

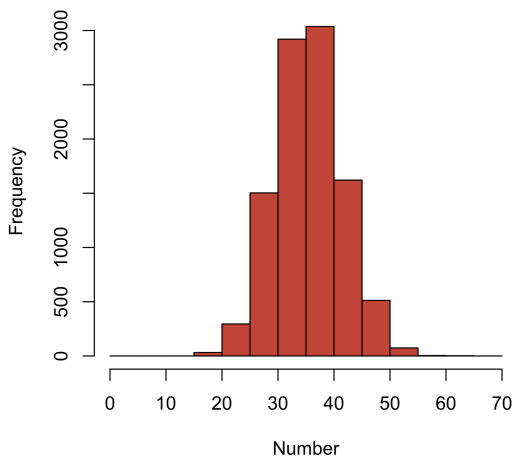
Input: Positive values `lambda, m`; positive integer `rep`.
Output: Array of `rep` numbers with Pois(`lambda` $*$ `m`) distribution.

1: Make empty numeric vector `ce`
2: **for** `i` $= 1, 2, \ldots,$ `rep` **do**
3: Make empty numeric vector `de`
4: **while** `sum(de)` $<$ `m` **do**
5: Generate pseudo-random `x` $\sim$ Exp(`lambda`)
6: Append `x` to array `de`
7: **end while**
8: Append `length(de)`-1 to array `ce`
9: **end for**
10: **return** `ce`

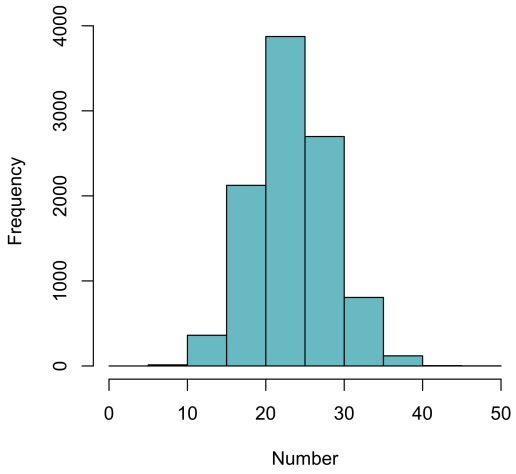
[1] The notebook with the code containing our analysis, as well as this report, can be found in the Github Repository: `https://github.com/palafox794/AppliedProbabilityModels/tree/master/Assignment4`



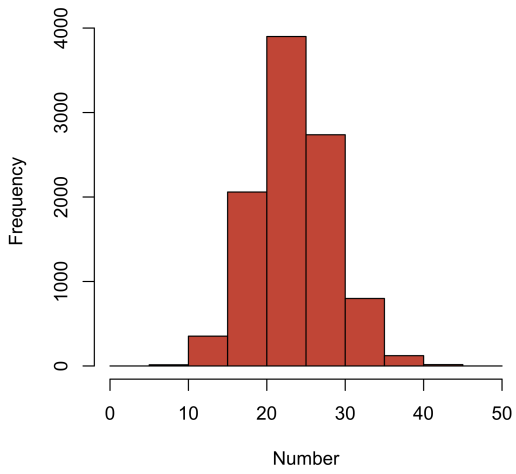
(a) Histogram of numbers generated by Algorithm 1 with `la` $= 12$, `me` $= 3$.



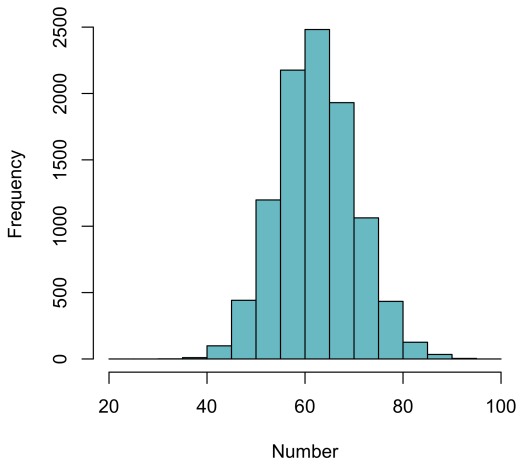
(b) Histogram of numbers generated by R with Pois(36) distribution.



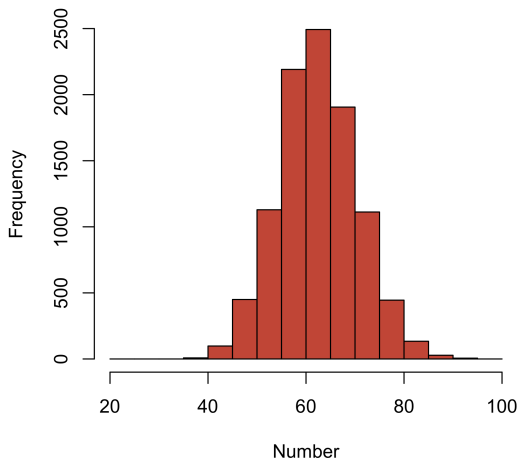
(c) Histogram of numbers generated by Algorithm 1 with `la` $= 4$, `me` $= 6$.



(d) Histogram of numbers generated by R with Pois(24) distribution.



(e) Histogram of numbers generated by Algorithm 1 with `la` $= 9$, `me` $= 7$.



(f) Histogram of numbers generated by R with Pois(63) distribution.

Figure 1: Comparison of Algorithm 1 vs R's generator.

Table 1: Proposed λ compared to λ given by `MASS::fitdistr`

Proposed λ	Estimated λ by `fitdistr`	Estimated standard error
36	35.954	0.059
24	23.895	0.048
63	62.853	0.079

2.1 Using uniform random variables

A different way of generating Poisson pseudo-random numbers is shown in Algorithm 2. The algorithm is attributed to Knuth [2], and it can be proven as a consequence of Algorithm 1.

Lemma 1. *If* $u \sim \text{Unif}(0,1)$ *and* $\lambda > 0$*, then* $\frac{-\log(u)}{\lambda} \sim \text{Exp}(\lambda)$.

Proof. First we find $\mathbb{P}\left(\frac{-\log(u)}{\lambda} \leq x\right)$. Observe that

$$\frac{-\log(u)}{\lambda} \leq x \Leftrightarrow -\log(u) \leq \lambda x \tag{1}$$

$$\Leftrightarrow \log(u) \geq -\lambda x \tag{2}$$

$$\Leftrightarrow u \geq \exp(-\lambda x) \tag{3}$$

Therefore,

$$\mathbb{P}\left(\frac{-\log(u)}{\lambda} \leq x\right) = \mathbb{P}\left(u \geq \exp(-\lambda x)\right) = 1 - \mathbb{P}\left(u \leq \exp(-\lambda x)\right) \tag{4}$$

From the cumulative distribution function for the uniform distribution, we see $\mathbb{P}\left(u \leq \exp(-\lambda x)\right) = \exp(-\lambda x)$, thus

$$\mathbb{P}\left(\frac{-\log(u)}{\lambda} \leq x\right) = 1 - \exp(-\lambda x) \tag{5}$$

This completes the proof. □

Proposition 1. *Knuth's Algorithm 2 generates pseudo-random numbers with a* $\text{Pois}(\lambda)$ *distribution when given a parameter* `lambda` $= \lambda$.

Proof. We know from Algorithm 1 that counting the number of exponential random variables with mean $1/\lambda$ such that $x_1 + \cdots + x_k < 1$ gives us a Poisson variable with mean λ. Using Lemma 1 we can rewrite this as $-\frac{\log(u_1)}{\lambda} - \cdots - \frac{\log(u_k)}{\lambda} < 1$, where $u_i \sim \text{Unif}(0,1)$. Then, observe that

$$-\frac{\log(u_1)}{\lambda} - \cdots - \frac{\log(u_k)}{\lambda} < 1 \Leftrightarrow -\log(u_1) - \cdots - \log(u_k) < \lambda \tag{6}$$

$$\Leftrightarrow -\log(u_1 u_2 \cdots u_k) < \lambda \tag{7}$$

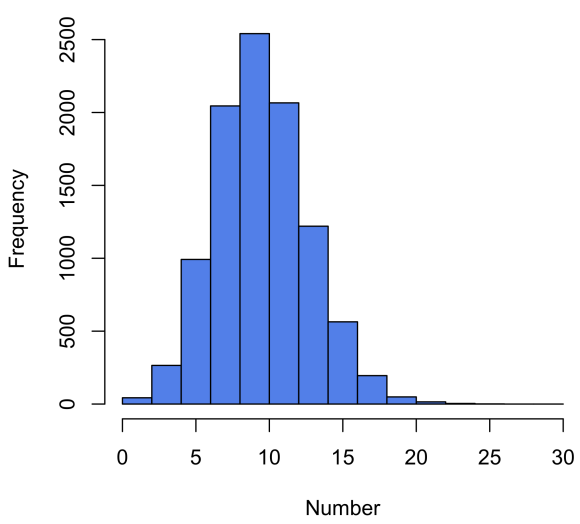
$$\Leftrightarrow u_1 u_2 \cdots u_k > \exp(-\lambda) \tag{8}$$

This shows that the number of exponential random variables generated before its sum exceeds one is the same as the number of uniform random variables generated before its product is less than $\exp(-\lambda)$ and so they must have the same distribution. □

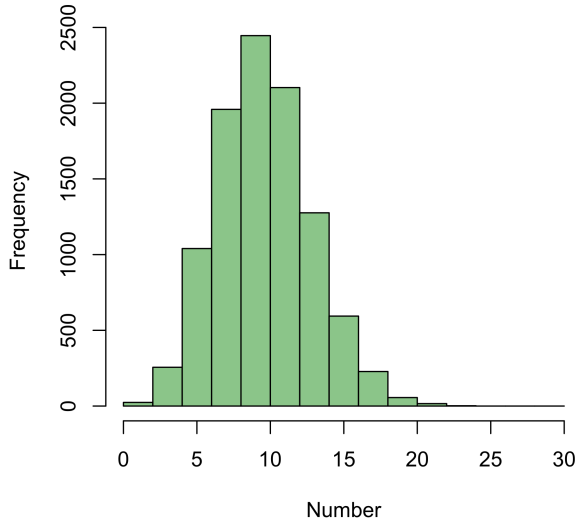
Supporting the formal proof, a comparison of numbers generated by Algorithm 2 and by R's `rpois` function is shown in Figure 2.

3 Binomial distribution tends to Poisson distribution

It is known [5] that for n large and small p, the binomial distribution $\text{Binom}(n,p)$ can be approximated with a Poisson distribution $\text{Pois}(np)$. This situation is exemplified with a random graph. An Erdős–Rényi random network [3], denoted as $G(n,p)$, is defined in the following way: n nodes are fixed, and an edge is placed between each distinct pair with independent probability p. It should be clear that the probability of any vertex having degree k is $\binom{n-1}{k}p^k(1-p)^{n-1-k}$. That is, the degree of vertices has a binomial distribution. A random network $G(10000, \frac{1}{10000})$ is created using R's library `igraph`. The degree distribution of this network is binomial, due to the justification given above. However, since n is sufficiently large and p is sufficiently small, the degree distribution can be approximated as a Poisson distribution. Figure 3a shows the degree distribution of the network, and Figure 3b shows an identical histogram for numbers having a Pois(1) distribution. Figure 3c shows a boxplot comparing these two sets of data.



(a) Histogram of numbers generated by Algorithm 2 with `la` = 10.



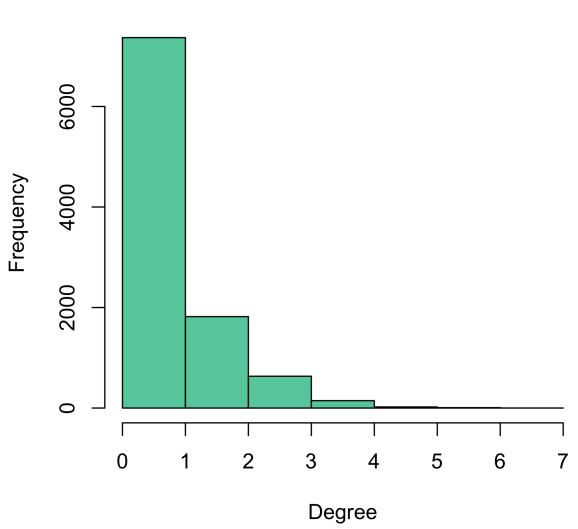
(b) Histogram of numbers generated by R with Pois(10) distribution.

Figure 2: Comparison of Algorithm 2 vs R's generator.

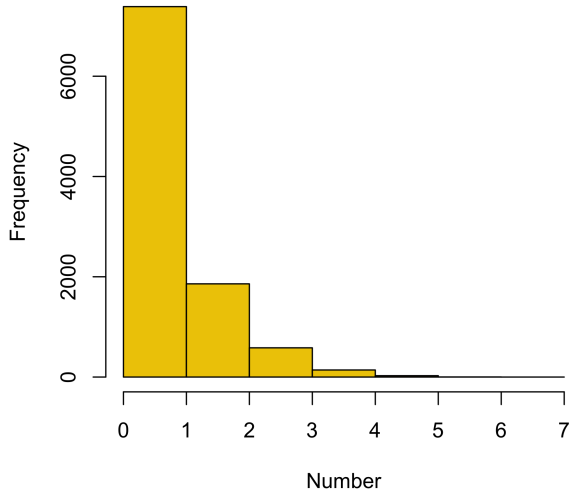
Algorithm 2 Knuth's algorithm

Input: Positive value `lambda`; positive integer `rep`.
Output: Array of `rep` numbers with Pois(`lambda`) distribution.

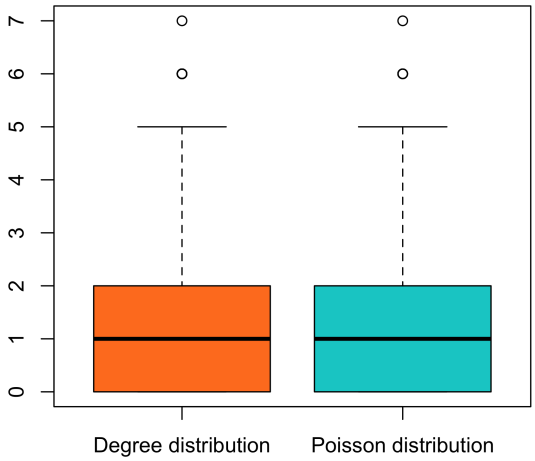
Make empty numeric vector `cu`
for `i` $= 1, 2, \ldots,$ `rep` **do**
    Make a numeric vector `du` containing only value 1
    while `prod(du)` $>$ `exp(-lambda)` **do**
        Generate pseudo-random `u` $\sim$ `Unif(0,1)`
        Append `u` to array `du`
    end while
    Append `length(du)-2` to array `cu`
end for
return `cu`



(a) Histogram of degrees in $G(10000, \frac{1}{10000})$ network.



(b) Histogram of 10 000 numbers generated by R with Pois(1) distribution.



(c) Boxplot comparing the data of Figures 3a and 3b.

Figure 3: Degree distribution of random network and Poisson-distributed numbers.

4 Conclusion

This work showed how some distributions (uniform, exponential, binomial) relate to the Poisson distribution. Further work may include the relation between normal and Poisson distributions. Additionally, it may be of interest to study how to use uniform values to generate pseudo-random numbers following distributions other than Poisson.

5 Acknowledgments

We thank Professor Elisa Schaeffer for providing an R code with Algorithms 1 and 2.

References

[1] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, et al., *Jupyter notebooks—a publishing format for reproducible computational workflows*, in Positioning and Power in Academic Publishing: Players, Agents and Agendas: Proceedings of the 20th International Conference on Electronic Publishing, IOS Press, 2016, p. 87.

[2] D. E. Knuth, *The art of computer programming*, vol. 2, Addison-Wesley, 3rd ed., 1997.

[3] M. Newman, *Networks*, vol. 1, Oxford University Press, Oct 2018.

[4] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2020.

[5] S. M. Ross, *A first course in probability*, Macmillan, 1976.

[6] ——, *Introduction to probability models*, Harcourt/Academic Press, 7th ed., 2000.