

# Practice 4: The Poisson distribution

Mayra Cristina Berrones Reyes 6291

September 29, 2020

## 1 Introduction

For this practice, we explore the properties of several R distributions, such as `rpois`, `runif`, and `rexp`, taking note in their difference as well as the similarities under certain circumstances. Before beginning with any experimentation, a brief description of each distribution will assist in the understanding of the problem.

### 1.1 Poisson distribution

A Poisson random variable can be used to model the number of times an event happened in a certain interval of time. It is described in terms of the rate in which this events happen, because an event can occur several times in a single interval. To be able to use this distribution we have to know the average number of events in an interval, which is designated by the sign of lambda ( $\lambda$ ) [2].

Lambda is also called the rate parameter. The probability of k events happening in an interval is represented in Equation 1:

$$P(k \text{ events in interval}) = e^{-\lambda} \frac{\lambda^k}{k!} \quad (1)$$

In R we can generate this distribution with the module `rpois`. The description for this module in R `help` says that is a “Density, distribution function, quantile function and random generation for the Poisson distribution with parameter  $\lambda$ ” [4]. The parameters it takes are stated in Equation 2:

$$\text{rpois}(n, \text{lambda}) \quad (2)$$

where

- **n** is the number of random values to return.
- **lambda** is the vector of non negative means.

### 1.2 Exponential distribution

The exponential distribution describes the time between events in a process in which they occur in a continuous and independent manner at a constant average rate. This distribution also

uses the parameter  $\lambda$  and it is called the rate parameter [1]. The equation for this distribution can be seen in Equation 3:

$$f(x; \lambda) = \begin{cases} \lambda \exp^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (3)$$

In **R** we can generate this distribution with the module **rexp**. The description for this module in **R help** says that is a “Density, distribution function, quantile function and random generation for the exponential distribution with rate (i.e., mean 1/rate)” [3]. This is a very important distinction, because we do not have to make the division in the variable ourselves, only add the *rate* value as it is [1].

The parameters it takes are stated in Equation 4:

$$\mathbf{rexp}(n, \text{rate} = 1) \quad (4)$$

where

- **n** is the number of observations. If  $\text{length}(n) > 1$ , the length is taken to be the number required.
- **rate** is the vector of rates.

### 1.3 Uniform distribution

The uniform distribution is one of the most simple and commonly used distribution. In the continuous uniform distribution we have the probability in Equation 5 as:

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b \\ 0 & \text{for } x < a \text{ or } x > b \end{cases} \quad (5)$$

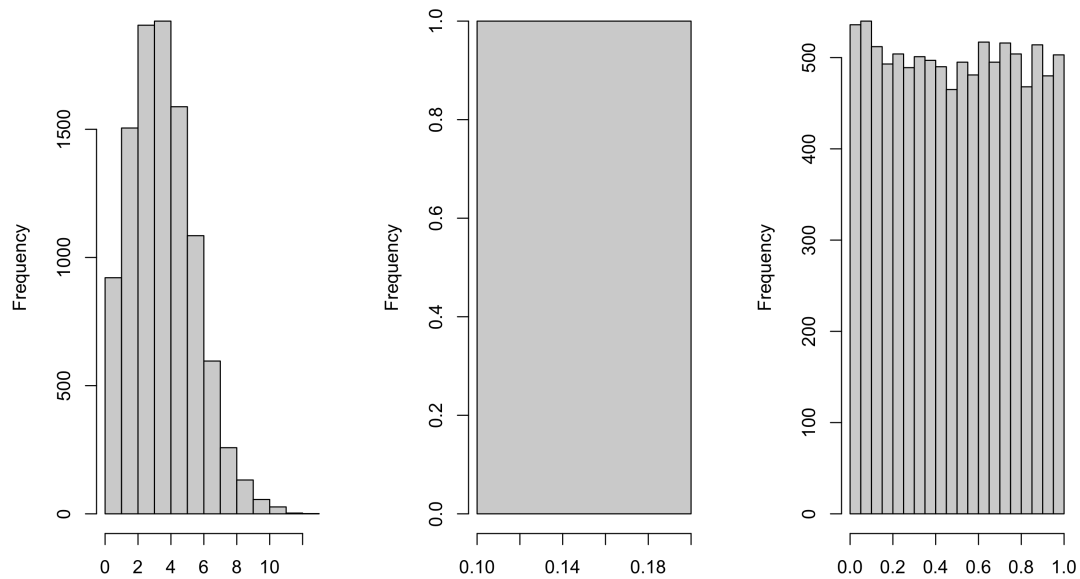
In this distribution, any interval of numbers of equal width have an equal probability of being observed. The curve describing the distribution is a rectangle with a constant height across interval and 0 height elsewhere. We can use in **R** a tool to see the this distribution, called **runif**. The description for this module in **R help** says that “These functions provide information about the uniform distribution on the interval from **min** to **max**. **runif** generates random deviates” [5].

The parameters it takes are stated in Equation 6:

$$\mathbf{runif}(n, \text{min} = 0, \text{max} = 1) \quad (6)$$

where

- **n** is the number of observations. If  $\text{length}(n) > 1$ , the length is taken to be the number required.
- **min**, **max** are the lower and upper limits of the distribution. Must be finite.



(a) Histogram of `rpois(10000, 4)`.      (b) Histogram of `rexp(1, 4)`      (c) Histogram of `runif(10000)`

Figure 1: Histograms showing the behaviours of each distribution (Poisson, Exponential and Uniform).

## 2 Comparisons between distributions

In Figure 1 we can appreciate how all of the described distributions behave. None of them look similar to one another. The goal of this practice is to show with which parameters we can make all of this distributions look similar, and explain why that is possible.

In order to search for ways to make them behave in a similar manner, we first needed an example of how the Poisson distribution works. Say we live near an airport, and we can see as an average, 4 planes in the span of one hour. Now as an experiment we record the number of planes we see every hour for 10,000 hours. In Subfigure 1a we can see such behaviour. As expected, the majority of the plane occurrences are 3, 4 or 5 inside the interval of an hour. There is a better chance of no planes showing than there is to have more than 7 in this interval. Understanding this, we begin now to translate this performance to the exponential and uniform distributions [7].

For the Poisson distribution it is possible to develop several generators that help us make a simply uniformly fast Poisson distribution. Such generators can be classified in several groups:

1. Generators that are based on the connection with homogeneous Poisson process. [6]. These type of generators are simple and run in expected time proportional to  $\lambda$ .
2. Inversion methods, that use sequential search starting at 0 and run in expected time

proportional to  $\lambda$ . If the sequential search starts here, the expected time is  $\mathcal{O}(\sqrt{\lambda})$  [8]

3. Generators that are based on recursive properties of the distribution.
4. Rejection methods.
5. The acceptance complement method with the normal distribution as a starting distribution.

In this case, one of the simple generators we have the exponential and uniform distribution. Beginning with the connection between the Poisson distribution and exponential arrival times in a homogeneous point process we have the Lemma 2.1 for the Poisson distribution [7].

**Lemma 2.1.** *If  $E_1, E_2, \dots$  are exponential random variables, and  $X$  is the smallest integer such that*

$$\sum_{i=1}^{X+1} E_i > \lambda$$

*then  $X$  is Poisson ( $\lambda$ ).*

As proof of this Lemma 2.1 we are presented with Proof 2:

*Proof.* Let  $f_k$  be the gamma ( $k$ ) density.

$$P(X \leq k) = P\left(\sum_{i=1}^{k+1} E_i > \lambda\right) = \int_{\lambda}^{\infty} f_{k+1}(y) dy$$

Thus, by partial integration,

$$\begin{aligned} P(X = k) &= P(X \leq k) - P(X \leq k-1) \\ &= \int_{\lambda}^{\infty} (f_{k+1}(y) - f_k(y)) dy \\ &= \int_{\lambda}^{\infty} (y - k) \frac{y^{k-1}}{k!} e^{-y} dy \\ &= \frac{1}{k!} \int_{\lambda}^{\infty} d(-y^k e^{-y}) \\ &= e^{-\lambda} \frac{\lambda^k}{k!}. \end{aligned}$$

□

In the end of this proof we see the same equation as we explained in the introduction in Equation 1 on the Poisson distribution. The algorithm that was based upon this proof is shown in Algorithm 1:

If we translate this to the example of the planes, we randomly generate 10,000 distributed random variables  $X$ , interpreting  $X$  as the waiting time from the passing of one plane to another measured by the hour. If we know the value of  $\lambda$  is 4 as the average of planes, the time that each event happens can be obtained from the cumulative sum of the waiting times from event to event. We can then aggregate the number of events that happen per unit time, and make an histogram

```

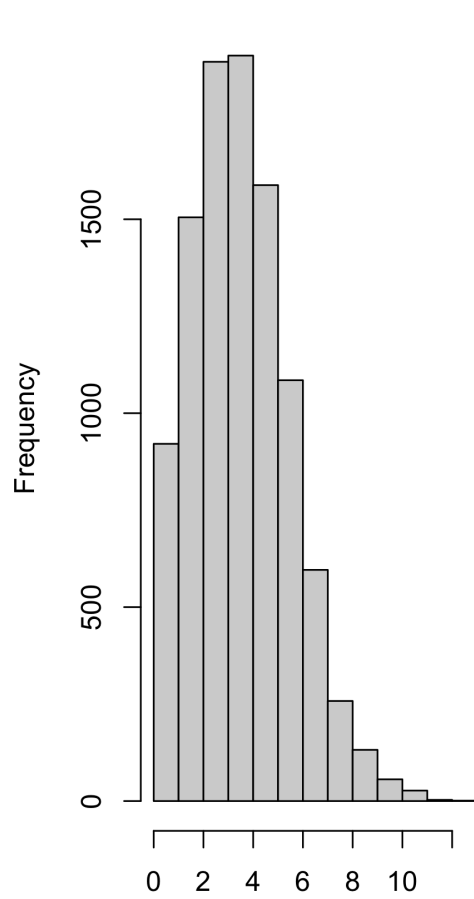
X ← 0;
Sum ← 0;
while TRUE do
    Generate an exponential random variate E;
    Sum ← Sum + E;
    if Sum <  $\lambda$  then
        | X ← X + 1;
    else
        | RETURN X;
    end
end

```

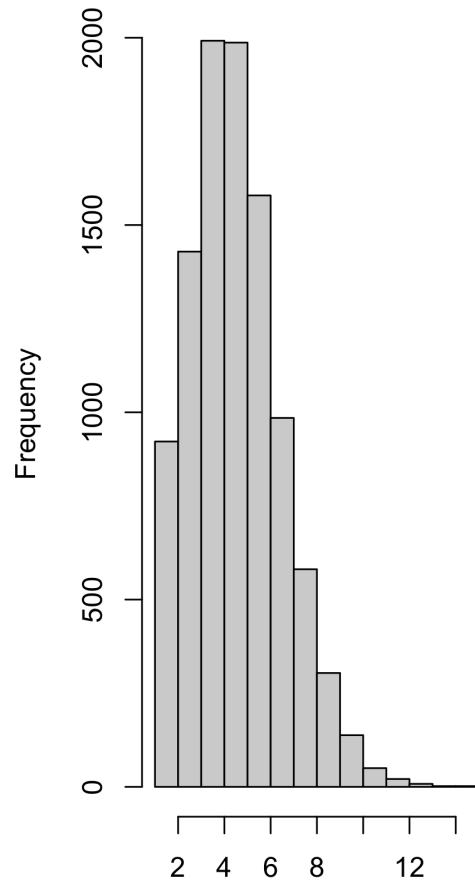
**Algorithm 1:** Poisson generator based upon exponential inter-arrival times

out of them.

Remembering the information of **R help** we put as a parameter the  $\lambda$  and not  $1/\lambda$ . Using the same parameters as the Poisson example we now have the results shown in Figure 2. Further analysis of both Subfigures 2a and 2b we can see that the behavior is very similar, in contrast with the first experimentation in Subfigure 1b



(a) Histogram of `rpois(10000, 4)`.



(b) Histogram of experimentation with cumulative sum of `rexp`

Figure 2: Histograms showing the behaviour of distribution Poisson and cumulative exponential experimentation.

For the uniform distribution we take on the fact that a uniform random variable is distributed as  $e^{-E}$ , we can equate the Lemma 2.1 to the next Lemma 2.2 [7].

**Lemma 2.2.** *Let  $U_1, U_2, \dots$  be uniform  $[0, 1]$  random variables, and  $X$  is the smallest integer such that*

$$\prod_{i=1}^{X+1} U_i < e^{-\lambda}$$

*then  $X$  is Poisson  $(\lambda)$ .*

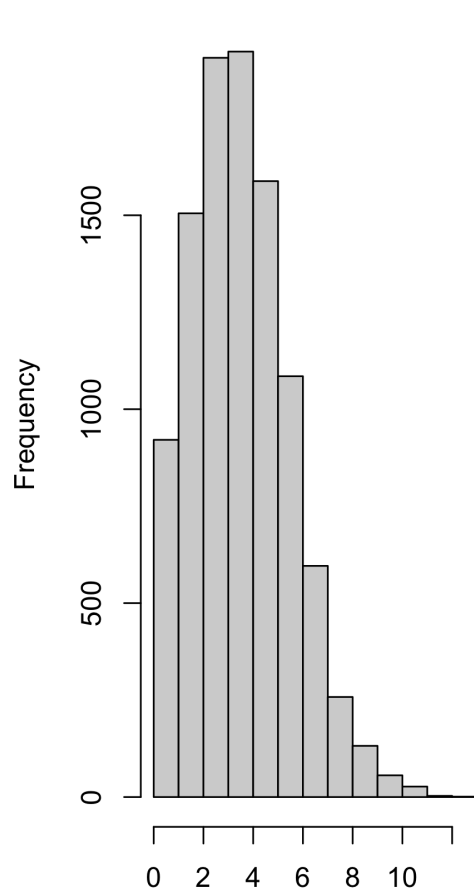
The resulting algorithm of this Lemma is equivalent as Algorithm 1

```

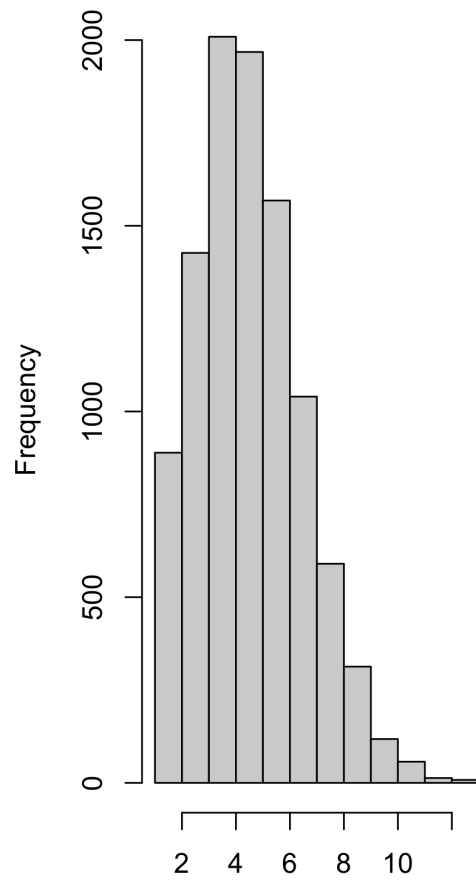
X ← 0;
Prod ← 1;
while TRUE do
    | Generate a uniform  $[0, 1]$  random variate U;
    | Prod ← Prod U;
    | if Prod >  $e^{-\lambda}$  (the constant should be computed only once) then
    | | X ← X + 1;
    | else
    | | RETURN X;
    | end
end

```

**Algorithm 2:** Poisson generator based upon multiplication of uniform random variates.



(a) Histogram of `rpois(10000, 4)`.



(b) Histogram of experimentation with multiplication of uniform random variates `runif`

Figure 3: Histograms showing the behaviours of distribution poisson and uniform random variates experimentation.

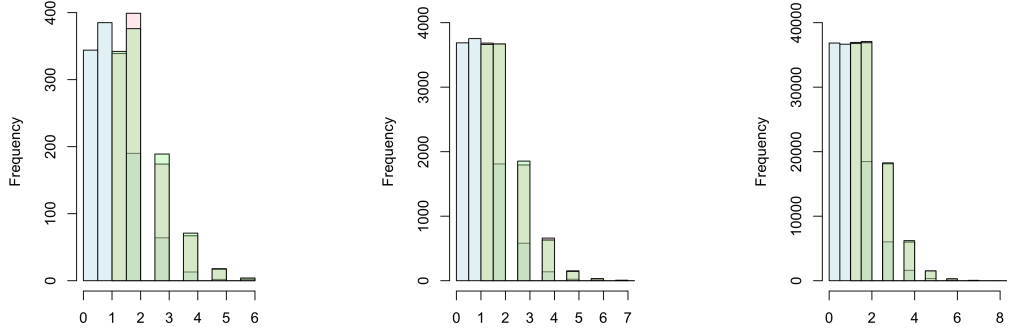


In this case, the experiment of the planes was not so clear in mind, because the involvement of the exponential part in the code did not make much sense. So we went to the R code to see what was the result of each of the experiments, exponential and uniform.

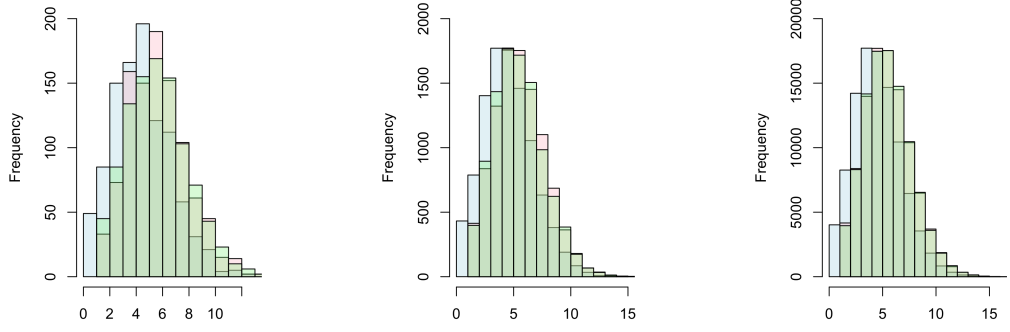
We printed the last iteration of each distribution, and in the case of the exponential, the cumulative sum was clear, as the results were 0.0614 and 1.4261. If the goal set is 1, those two events were enough to complete the interval. In the case of the uniform distribution the last iteration numbers were 0.2350, 0.8153, 0.4806, and 0.0748. At first this did not make sense, but after calculating the product of all these numbers and comparing them to the result of  $1 - \lambda$  it made sense why we had to use multiplication instead of cumulative sum.

### 3 Experimentation with different parameters

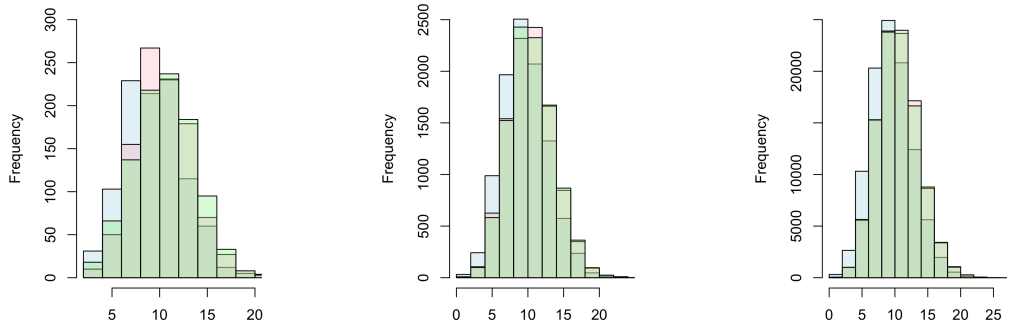
In Figure 4 there we can see all the distributions in different parameters. It is very curious that the lowest parameter on lambda in all three different iterations of Subfigures 4a, 4b and 4c, they all show a behaviour similar to the exponential that we saw in the introduction of this work. Subfigures 4d, 4e and 4f, all with  $\lambda = 5$ , resemble the Poisson distribution and Subfigures 4g, 4h and 4i with  $\lambda = 10$  to the normal distribution.



(a)  $\lambda = 1$ , and replica = 1,000    (b)  $\lambda = 1$ , and replica = 10,000    (c)  $\lambda = 1$ , and replica = 100,000



(d)  $\lambda = 5$ , and replica = 1,000    (e)  $\lambda = 5$ , and replica = 10,000    (f)  $\lambda = 5$ , and replica = 100,000



(g)  $\lambda = 10$ , and replica = 1,000    (h)  $\lambda = 10$ , and replica = 10,000    (i)  $\lambda = 10$ , and replica = 100,000

Figure 4: Histograms of all the distributions with different parameters. The blue represents the Poisson distribution, the pink is the exponential distribution, and the green the uniform distribution.

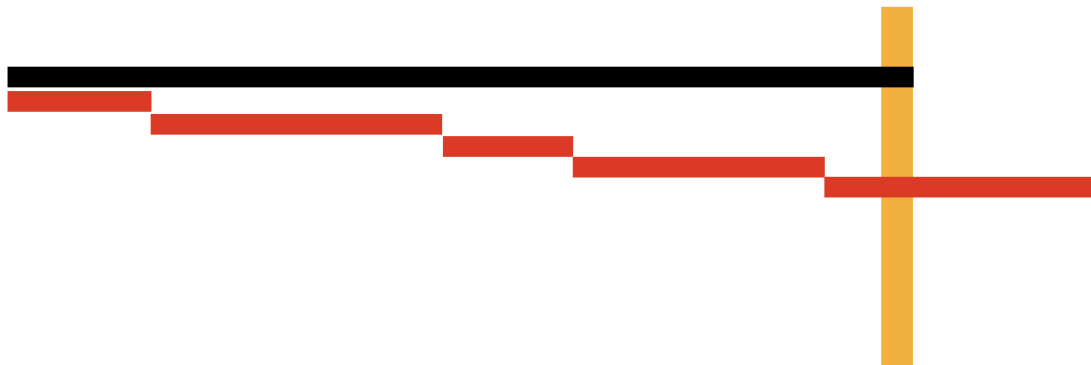


Figure 5: Example made by Dr. Elisa as a visual aid of how the Poisson distribution works.

## 4 Conclusions

The conclusions arrived in this practice were more clear in the exact order we mentioned them. Seeing the Poisson distribution first as an example made it easier to understand. Then to transform the exponential distribution to the behaviour of the Poisson one was easy because the concept of cumulative sum as depicted on the Figure 5 Dr. Elisa gave us in class was simple. After all of that, the uniform distribution using multiplication to lower the final result to fit inside the exponential condition was a bit harder to understand, but the experimentation in **R** made it easier to visualize and digest.

Figure 4 is also something interesting, how changing the parameters of lambda, we can change the behaviour of the distribution, making it seem like other distributions.

## References

- [1] Probability distributions in python. <https://www.datacamp.com/community/tutorials/probability-distributions-python>. Accessed: 2020-09-28.
- [2] The poisson distribution and poisson process explained. <https://towardsdatascience.com/the-poisson-distribution-and-poisson-process-explained-4e2cb17d459>. Accessed: 2020-09-28.
- [3] The exponential distribution. <https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/Exponential>. Accessed: 2020-09-28.
- [4] The poisson distribution. <https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/Poisson>. Accessed: 2020-09-28.
- [5] The uniform distribution. <https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/Uniform>. Accessed: 2020-09-28.

- [6] AC Atkinson. The computer generation of poisson random variables. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):29–35, 1979.
- [7] Luc Devroye. Random variate generators for poisson – poisson and related distributions. *Computational Statistics and Data Analysis*, (8):247–278, 1989.
- [8] CD Kemp. New algorithms for generating poisson variates. *Journal of computational and applied mathematics*, 31(1):133–137, 1990.