Tutorial 3 – Giovanni Filomeno - 12315325

Exercise 14

a) Random walk

My code for the Week-5 is presented in Figure 1, while the generated path is shown in Figure 2.

```r
# Part A
RW <- function(N, p = 0.5) {
  # Generating all the steps
  steps <- sample(c(-1, 1), size = N, replace = TRUE, prob = c(1 - p, p))
  c(0, cumsum(steps))
}

# Exercise parameters
set.seed(2024)
N <- 8
path <- RW(N)

plot(0:N, path, type = "o", pch = 16, col = "steelblue",
     xlab = "n", ylab = expression(X[n]),
     ylim = range(path))
grid()
```

Figure 1: Week-5 Algorithm

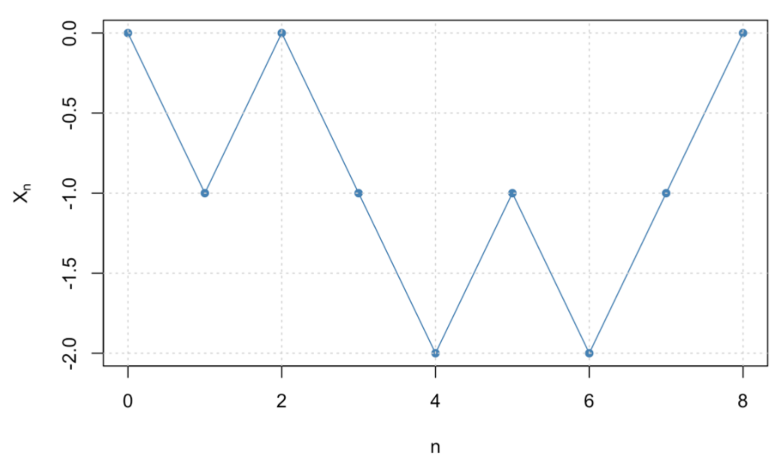

Figure 2: Generated path

b) Comparison with binomial distribution

Figure 3 shows the comparison between the distribution of X_n and a binomial distribution. The two distributions match perfectly, therefore X_n can be described with a binomial distribution.

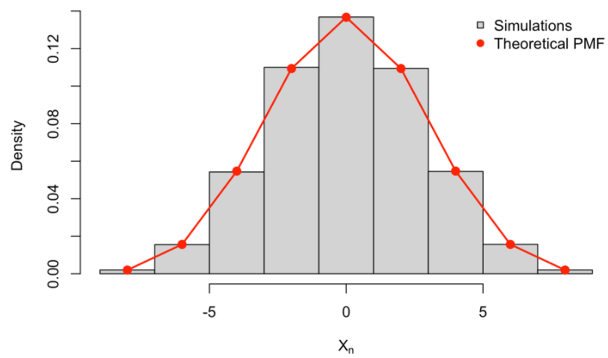

Figure 3: Comparison of the two distributions

Exercise 15

a) Transition matrix

I can build the transition matrix analyzing all the three states singularly:

- $P(1 \to 1) = P_{11} = 1/2$
- $P(1 \to 2) = P_{12} = 1/4$
- $P(1 \to 3) = P_{13} = 1/4$
- $P(2 \to 1) = P_{21} = 1/3$
- $P(2 \to 2) = P_{22} = 1/3$
- $P(2 \to 3) = P_{23} = 1/3$
- $P(3 \to 1) = P_{31} = 1/2$
- $P(3 \to 2) = P_{32} = 1/4$
- $P(3 \to 3) = P_{33} = 1/4$

Therefore, the transition matrix is:

$$P = \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ 1/3 & 1/3 & 1/3 \\ 1/2 & 1/4 & 1/4 \end{pmatrix}$$

b) Distribution of τ

Given the problem, the probability of leaving the state 1 at step k is given by $P(\tau = k) = P_{11}^{k-1}(1 - P_{11})$, because at each step it has the probability P_{11} of remaining in the state 1 and $(1 - P_{11})$ of leaving the state 1. According to the transition matrix P presented in 15.b, the equation can be simplified in $P(\tau = k) = \left(\frac{1}{2}\right)^{k-1} \cdot \frac{1}{2} = \left(\frac{1}{2}\right)^{k}$. This formula leads to:

- $P(\tau = 1) = \frac{1}{2}$
- $P(\tau = 2) = \frac{1}{4}$
- $P(\tau = 3) = \frac{1}{8}$

The founded formula describes a geometric distribution with parameter $p = 1/2$ (link: https://en.wikipedia.org/wiki/Geometric_distribution).

Exercise 16

a) Transition Matrix

Exactly as before, I can build the matrix starting from the chain in the picture:

- $P(1 \to 3) = P_{13} = 1$
- $P(2 \to 1) = P_{21} = 1$
- $P(3 \to 2) = P_{32} = 1/2$
- $P(3 \to 4) = P_{34} = 1/2$
- $P(4 \to 3) = P_{43} = 1/2$
- $P(4 \to 5) = P_{45} = 1/2$
- $P(5 \to 4) = P_{54} = 1$

Which leads to the following transition matrix:

$$P = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Since the exercise states that the initial value is $X_0 = 1$ then a proper initial vector would be $\alpha = [1,0,0,0,0]$.

b) Algorithm and Path

Figure 4 shows my approach for the exercise, while Figure 5 presents the resulting path.

```r
# Markov Chain
MC <- function(N, P, x0) {
  S<- nrow(P)
  path <- integer(N + 1)
  path[1] <- x0
  for (n in 1:N)
    path[n + 1] <- sample(1:S, 1, prob = P[path[n], ])
  path
}

# Transition matrix
P <- matrix(c(0,0,1,0,0,
              1,0,0,0,0,
              0,0.5,0,0.5,0,
              0,0,0.5,0,0.5,
              0,0,0,1,0),
            nrow = 5, byrow = TRUE)

set.seed(2024)
N<- 20
path <- MC(N, P, 1)

plot(0:N, path, type = "o", pch = 16, col = "steelblue",
     ylim = c(1, 5), yaxt = "n", xlab = "n", ylab = expression(X[n]))
axis(2, at = 1:5)
grid(nx = NA, ny = NULL, lty = 2)
```

Figure 4: Markov chain algorithm

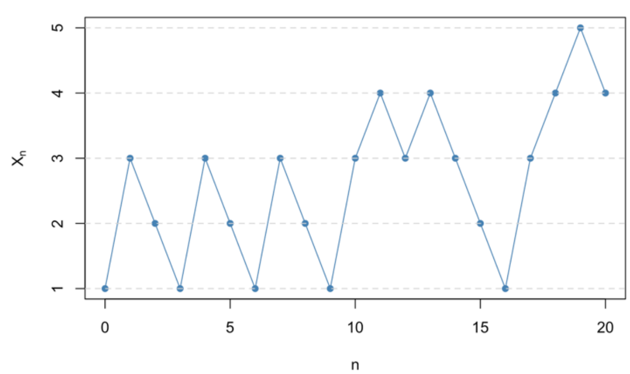

Figure 5: Path of Markov chain

Exercise 17

a) Compute the probability

Figure 6 presents my code, which gives me a probability of 0.1875.

```r
library(matrixcalc)
# Transition matrix
transMat <- matrix(c(0,0,1,0,0,
                     1,0,0,0,0,
                     0,.5,0,.5,0,
                     0,0,.5,0,.5,
                     0,0,0,1,0),
                   byrow = TRUE, ncol = 5)

MC5stateProbability <- function(m, j, i, transMat){
  Pm <- matrix.power(transMat, m)   # matrix.power
  Pm[i, j]
}

MC5stateProbability(4, 3, 3, transMat)
```

Figure 6: Probability algorithm

b) Proof of stationary distribution

To solve this exercise, I have to define the following vector $\boldsymbol{\pi} = \{\pi_1, \pi_2, \pi_3, \pi_4, \pi_5\}$ with the following condition $\sum_i \pi_i = 1$. I use this vector to impose the equation $\pi_j = \sum_i \pi_i P_{ij}$.

Substituting the element of the matrix P, I obtain:

- $\pi_1 = \pi_2$
- $\pi_2 = \frac{1}{2}\pi_3$
- $\pi_3 = \pi_1 + \frac{1}{2}\pi_4$
- $\pi_4 = \pi_5 + \frac{1}{2}\pi_3$
- $\pi_5 = \frac{1}{2}\pi_4$

Which leads to the final vector $\boldsymbol{\pi} = \left\{\frac{1}{7}, \frac{1}{7}, \frac{2}{7}, \frac{2}{7}, \frac{1}{7}\right\}$. Therefore, since it is possible to solve the equation and finding the vector, it exists a stationary distribution (checked empirically with R).

Exercise 18

a) Implement the algorithm

In Figure 7 the algorithm to generate a path of the homogeneous Poisson is presented.

```r
simulate_poisson_times <- function(lambda, T) {
  times <- c()
  t <- 0
  while (TRUE) {
    t <- t + rexp(1, rate = lambda)
    if (t > T) break
    times <- c(times, t)
  }
  return(times)
}

poisson_path <- function(lambda, T, dt = 0.01) {
  arrivals <- simulate_poisson_times(lambda, T)
  grid     <- seq(0, T, by = dt)
  counts   <- findInterval(grid, arrivals)
  data.frame(t = grid, Nt = counts)
}

```

Figure 7: Homogeneous Poisson path algorithm

b) Plot a path for $\lambda = 3$ and $T = 10$

Figure 8 shows the result of the algorithm with the given parameters.

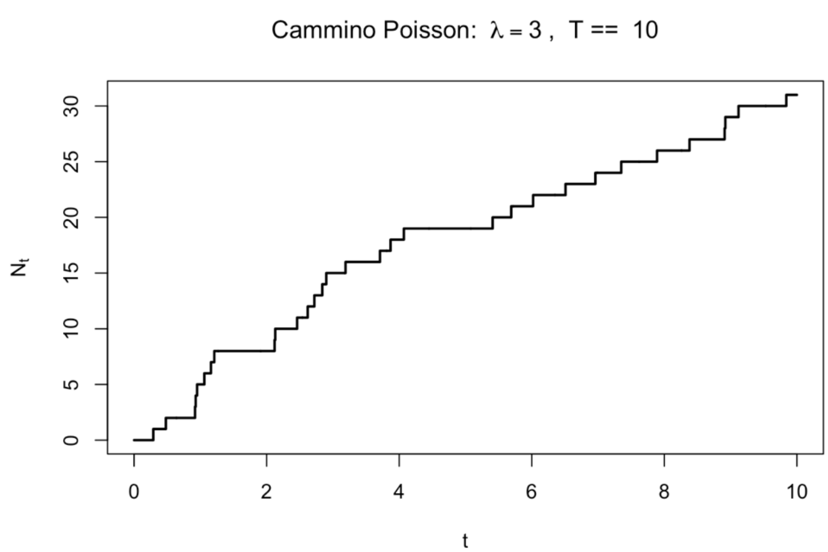

Figure 8: Homogeneous Poisson path given the exercise parameters

Exercise 19

a) Algorithm Type 1

Figure 9 shows the difference between the two pmf. To better address the differences I also evaluated the mean and variance between the theoretical and empirical pmf. The mean is 5.9911 and 6 for the empirical and theoretical respectively, while the variance is 5.907812 for the empirical and 6 for the theoretical.

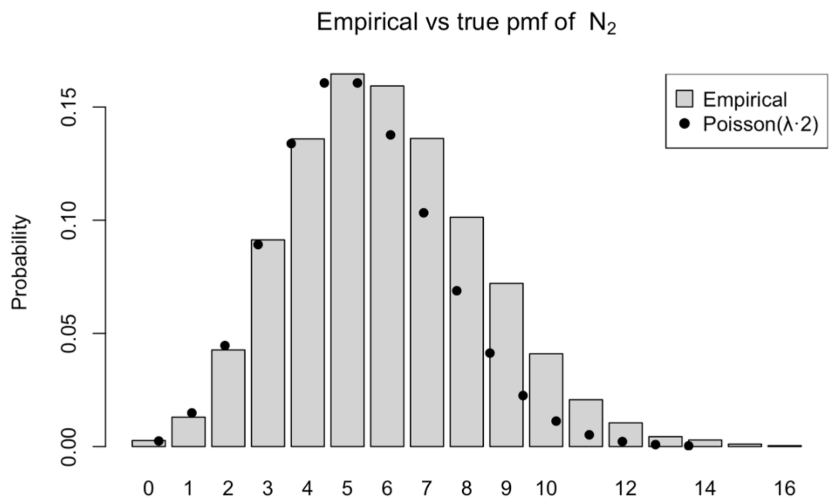

Figure 9: Empirical vs Theoretical pmf of N_2

b) Algorithm Type 2

Similarly to part a), Figure 10 presents the differences. The theoretical mean is 1.333 and the empirical mean is 1.325948. Regarding the variance, it is 0.4444 for the theoretical and 0.4531797 for the empirical.

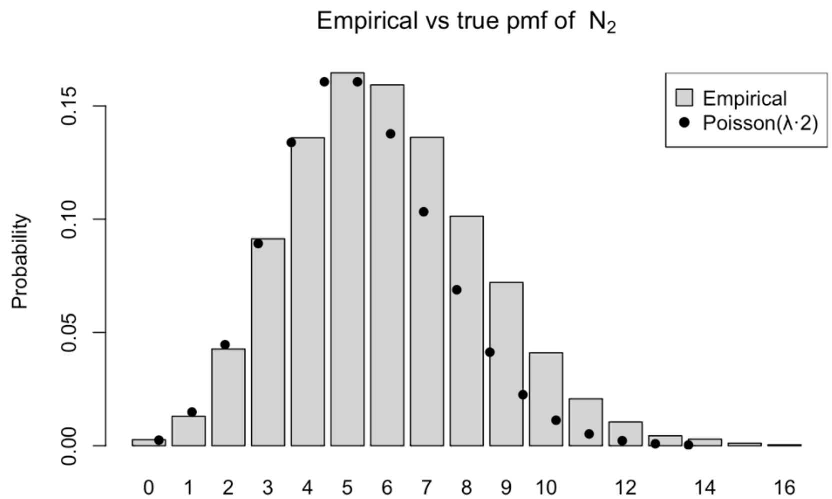

Figure 10: Empirical vs Theoretical pmf of T_4

Exercise 20

Before resolving the exercise, I want to clarify that $\lambda(t) = 5\sin^2\left(\frac{t}{2}\right) = 2.5(1 - \cos t)$ and that $\Lambda(t) = \int_0^t \lambda(s)ds = 2.5(t - \sin t)$, therefore $N_t \sim \text{Poisson}\big(\Lambda(t)\big)$.

a) Implement the algorithm and b) plot a path with $T = 30$

Figure 11 shows the implemented algorithm and Figure 12 presents the results of a run with $T = 30$.

```
rm(list = ls())
set.seed(123)
lambda_fun  <- function(t) 5 * sin(t / 2)^2
lambda_max  <- 5 # lambda sup
Lambda_fun  <- function(t) 2.5 * (t - sin(t)) # exaplained in the pdf

simulate_nhpp_times <- function(lambda_fun, lambda_max, T) {
  times <- numeric(0)
  t     <- 0
  repeat {
    t <- t + rexp(1, rate = lambda_max)   # candidate
    if (t > T) break
    if (runif(1) <= lambda_fun(t) / lambda_max)
      times <- c(times, t) # point accepted
  }
  times
}

## restituisce N_t (conteggio) per un tempo singolo
simulate_Nt_nhpp <- function(lambda_fun, lambda_max, t) {
  length(simulate_nhpp_times(lambda_fun, lambda_max, t))
}

```

Figure 11: Non-homogeneous Poisson Process algorithm

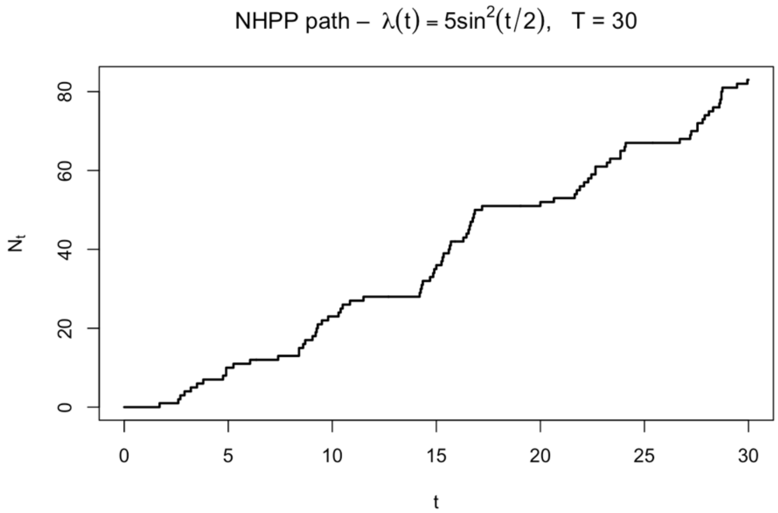

Figure 12: Path with $T = 30$

b) Theoretical vs Empirical pmf

Similar to the previous exercises, Figure 13 compares the theoretical and empirical pmf. In the case of a Poisson distribution, mean and variance are the same.

The theoretical mean/variance is 14.89731, while the empirical mean is 14.9142 and the empirical variance is 14.93173.

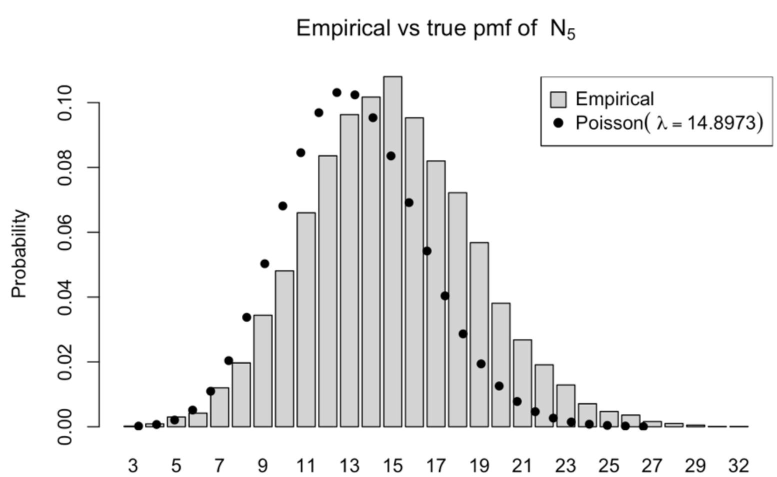

Figure 13: Empirical vs Theoretical pmf