Trabajo de Simulación de Distribuciones de Probabilidad

Francisco Javier Jiménez Montes

Juan Marqués Garrido

Pablo Jesús Moreno Polo

Introducción

El propósito de está memoria es describir el algoritmo programado para realizar nuestra simulación de la distribución de Poisson de variable λ con muestra de tamaño N.

Generación de la Muestra

Para generar nuestra muestra de valores necesitamos generar valores $U_i \in (0,1)$. Necesitamos obtener los valores necesarios aplicando $V_i = -ln(U_i)$. Con estos V_i vamos a obtener los valores de nuestra variable aleatoria $X \sim P(\lambda)$, la cual es el numero de términos V_i que se han sumado antes de que la suma exceda λ.

La función que ha sido programada para generar los valores de la variable X con los parámetros $N\ y\ \lambda$ es:

```
gen_poisson_values <- function(lambda) {
  sum <- 0
  count <- 0
  while (sum < lambda) {
    sum <- sum - log(runif(1))
    count <- count + 1
  }
  return(count)
}
```

El parámetro *lambda* es nuestro λ.

Para generar nuestros N y ademas obtener el menor y el mayor valor obtenidos (los cuales nos serán útiles posteriormente para generar nuestros histogramas), tenemos estas tres lineas:

```
for (i in 1:size) {
  pois_vals <- append(pois_vals, gen_poisson_values(lambda))
}
min_val <- min(pois_vals)
max_val <- max(pois_vals)
```

Donde *size* es nuestra N y *pois_ vals* es un *array* en el cual almacenamos nuestros valores.

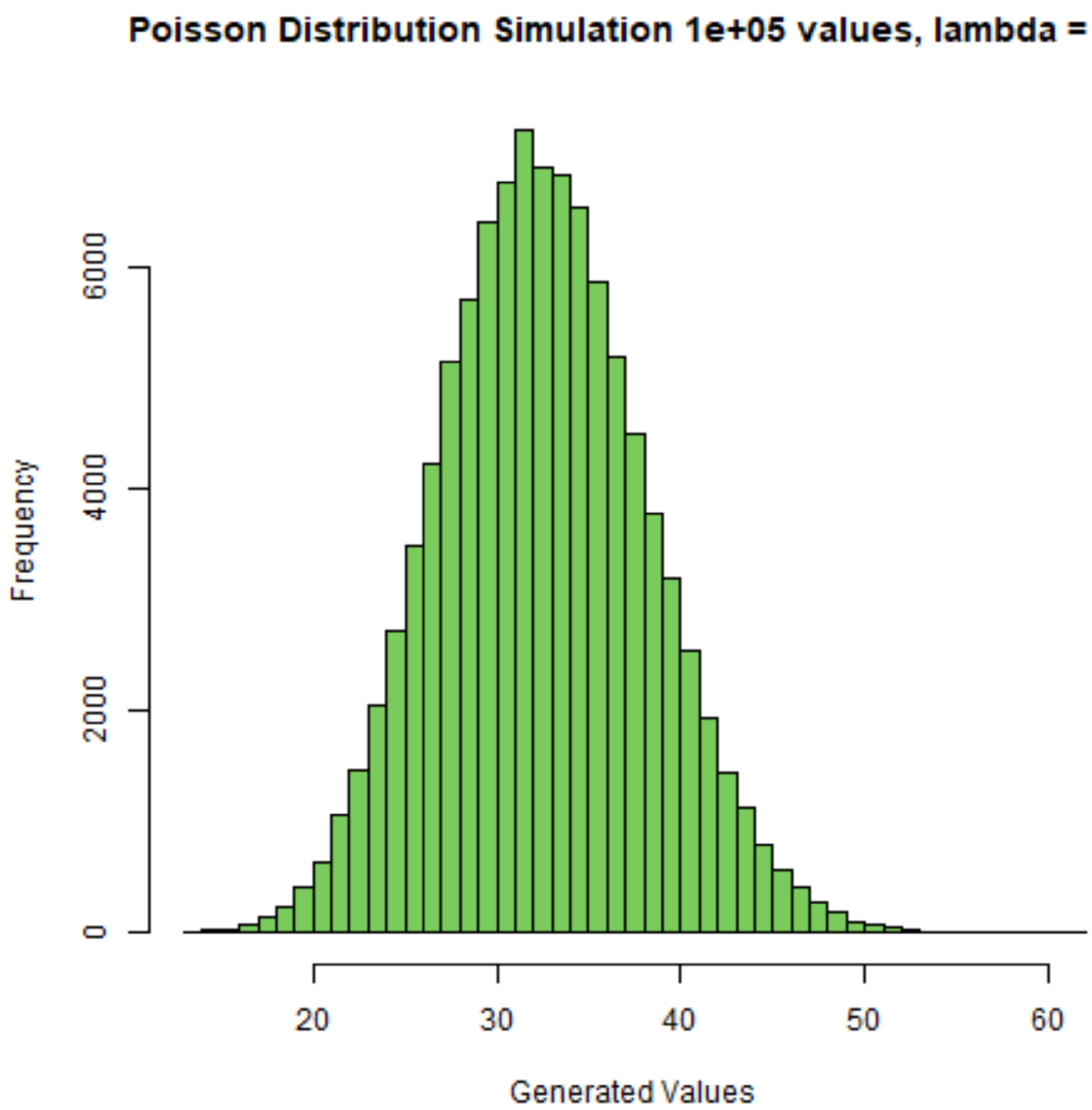
Representando nuestros datos con un histograma

Ahora tenemos nuestro vector con valores, pero visualizar estos datos es bastante complicado con un vector; es por eso que aquí entra en juego representar nuestros datos de una forma gráfica en un histograma.

Para generar un histograma usamos la función *hist()* con estos parámetros.

```
hist(pois_vals,
      main = paste("Poisson Distribution Simulation",
                   size, "values, lambda =", lambda
                 ),
      xlab = "Generated Values",
      xlim = c(min_val, max_val),
      breaks = max_val - min_val,
      col = "#79cc5b"
    )
```

Tras ejecutar está linea obtenemos la siguiente gráfica:



Estimación de probabilidad de intervalos de extremos finitos

Para estimar la probabilidad de todos los intervalos desde *min_ val* hasta *max_ val* hacemos uso de este bucle for sobre el cual iteramos hasta tener todos los errores de los intervalos, hacemos la suma de todos ellos y al final printeamos el error medio.

```
error_sum <- 0
for (i in min_val:max_val) {
  prob_dist <- ppois(i, lambda)
  prob_sample <- length(which(pois_vals < i)) / size

  error <- abs(prob_dist - prob_sample) / prob_dist
  error_sum <- error_sum + error
  print(sprintf(interval_format,
                prob_sample,
                prob_dist,
                error
                )
        )
}
sprintf("Mean error : %f", error_sum / size)
```

Estimaciones puntuales de la media y de la varianza muestral

La estimación de dichos estadísticos es sencilla de realizar, pues se puede hacer en tan solo 2 lineas de código.

```
pois_mean <- mean(pois_vals)
pois_var <- var(pois_vals)
```

Aplicación del Teorema Central del Límite

El Teorema Central del Límite, también llamado CLT (o *Central Limit Theorem)*, es aplicado sobre una muestra $\{x_1, x_2,..., x_n\}$ para generar una nueva muestra $\{y_1, y_2,..., y_n\}$. Cada valor y_i se calcula de la siguiente forma:

$$y_i = \frac{x_i - \bar{x}}{+\sqrt{S^2}}$$

La función CLT que vamos a necesitar se define de la siguiente forma:

```
clt_function <- function(x) {
  num_aux <- (x - pois_mean) / sqrt(pois_var)
  return(num_aux)
}
```

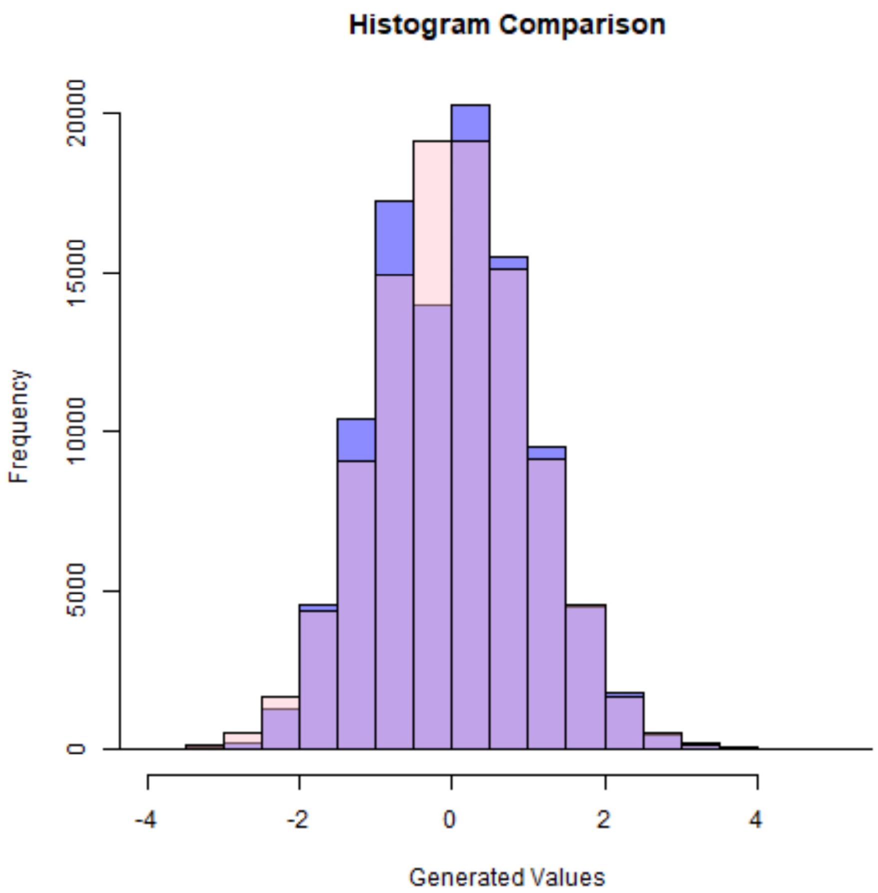
Posteriormente, aplicamos la función sobre nuestro vector de valores y obtenemos su media y desviación típica muestral.

```
clt_values <- sapply(pois_vals, clt_function)
clt_mean <- mean(clt_values)
clt_sd <- sd(clt_values)
```

Comprobando ambos valores observamos que coinciden con los resultados esperados.

```
> clt_mean
[1] 3.258299e-16
> clt_sd
[1] 1
```

Si superponemos un histograma con valores generados aleatoriamente de una distribución normal con un histograma de nuestros valores tras aplicarle la función CLT obtenemos la siguiente imagen.



Donde el histograma azul representa los valores después de aplicar la función CLT y el rosado representa los valores generados de la distribución normal.