
MÉTODOS DE MONTE CARLO Y SUS APLICACIONES.

TRABAJO FIN DE GRADO

Autor:

Lorena Ramos Segura

Tutor:

Antonio Salmerón Cerdán

GRADO EN MATEMÁTICAS



MAYO, 2021
Universidad de Almería

Índice general

1	Introducción	1
1.1.	Antecedentes	1
1.2.	Objetivos	1
2	Métodos de Monte Carlo	3
2.1.	Números aleatorios	3
	Generación de números aleatorios, 3.— Medidas estadísticas de la calidad de un generador de números aleatorios, 8.	
2.2.	Generación de muestras aleatorias	11
	Método de la transformada inversa, 11.— Método de composición, 12.— Convolución, 12.— Método de aceptación-rechazo, 13.— Casos particulares, 14.	
2.3.	Estimación por Monte Carlo	16
3	Teoría de colas	19
3.1.	Conceptos básicos	19
	Notación y terminología, 20.— Fórmulas de Little, 22.	
3.2.	Modelo M/M/1	22
3.3.	Modelos con distribución arbitraria	25
	Simulación, 27.	
4	Resultados	29
	Primera fase, 29.— Segunda fase, 31.— Tercera fase, 34.	
5	Conclusión	43
	Bibliografía	45

Abstract in English

This final degree project deals with Monte Carlo methods.

The first part focuses on randomness. The generation of random numbers and random samples are part of this section. These aspects are necessary for the application of these methods. The last section is the Monte Carlo estimate.

The next chapter covers queuing theory. All the definitions and basic concepts about this theory are included in the first section. These notations will be used to model the problem and carry out the convenient simulation. Likewise, the two most important queuing models are exposed: the M/M/1 model and the model with arbitrary distribution.

Lastly, we present the results obtained when applying the Monte Carlo methods to the problem of determining the mean wait of a patient in a Hospital Emergency Unit.

Resumen en español

El presente trabajo de fin de grado trata acerca de los métodos de Monte Carlo.

La primera parte se centra en la aleatoriedad, esto es, en la generación de números aleatorios y muestras aleatorias, necesarias para la aplicación de estos métodos. Como último apartado se expone la estimación por Monte Carlo.

El siguiente capítulo abarca la teoría de colas. En él se encuentran todas las definiciones y conceptos básicos sobre dicha teoría. Estas notaciones servirán para poder modelar el problema y realizar la conveniente simulación. Así mismo, se exponen los dos modelos de colas más importantes: el modelo $M/M/1$ y el modelo con distribución arbitraria.

En último lugar se presentan los resultados obtenidos al aplicar los métodos de Monte Carlo al problema de determinar la espera media de un paciente en una Unidad de Urgencias Hospitalaria.

Introducción

1.1 Antecedentes

El origen de estos métodos se remonta a finales de los 40 cuando fueron desarrollados por Stan Ulam y John Von Neumann [7] en su investigación sobre el movimiento aleatorio de los neutrones.

Una primera consecuencia de la aleatoriedad presente en estos métodos es su propio nombre, acuñado así por el principado de Mónaco conocido por ser "la capital del juego de azar".

La definición formal de los métodos de Monte Carlo es la siguiente:

Definición 1.1. *Un método de Monte Carlo es una técnica numérica para resolver un problema de carácter determinista mediante la introducción de un componente aleatorio para simplificarlo.*

En relación a estos métodos podemos encontrar la teoría de colas, la cual permite modelar sistemas dinámicos complejos y que permiten tomar decisiones óptimas sobre la asignación de determinados recursos.

1.2 Objetivos

En este trabajo nos planteamos la resolución de un problema práctico real consistente en la determinación de los recursos en un servicio de urgencias hospitalarias.

Este objetivo general lo abordamos mediante los siguientes objetivos específicos:

1. Estudio teórico de los fundamentos de los métodos de Monte Carlo.
2. Estudio teórico de los fundamentos de la teoría de colas.
3. Modelado del problema de las urgencias hospitalarias mediante colas.
4. Resolución del problema mediante Simulación Monte Carlo.

Métodos de Monte Carlo

2.1 Números aleatorios

Los métodos de Monte Carlo se apoyan en la obtención de una muestra para aquellas variables aleatorias del problema en cuestión que no disponen de alguna. Por esta razón, se recurre a la generación artificial de muestras.

Los métodos para obtener muestras aleatorias parten de los números aleatorios. La definición formal de número aleatorio sería la siguiente:

Definición 2.1. *Un número aleatorio es una realización de una variable aleatoria que tiene asociada una ley de probabilidades F , en un espacio de probabilidades (Ω, \mathcal{R}, P) .*

Sin embargo, es preferible tratar con secuencias de números aleatorios:

Definición 2.2. *Una secuencia de números aleatorios es una sucesión de variables aleatorias X_1, X_2, \dots, X_n donde:*

- Cada variable X_i sigue una distribución uniforme en el intervalo $[0,1)$.
- Las variables son independientes.

De esta forma, una vez hemos definido estos elementos iniciales pasemos a estudiar cómo obtenerlos.

Generación de números aleatorios

Algunas de las primeras formas de obtener números aleatorios eran tirar unos dados, sortear bolas en una urna o lanzar una moneda. Sin embargo, estos métodos han ido quedando obsoletos a medida que la tecnología ha avanzado.

Los procedimientos para la generación de dichos números se clasifican en:

1. Métodos manuales.
2. Mecanismos analógicos y tablas.
3. Métodos digitales.

De entre todas estas posibilidades el método idóneo para las aplicaciones del tipo Monte Carlo es, normalmente, el uso de algoritmos determinísticos, formalmente, **generadores de números pseudoaleatorios**. Este matiz en la aleatoriedad de los números se debe al hecho de que todos los elementos de la sucesión generada son predecibles a partir del primero, denominado semilla.

Ciertas propiedades exigibles a un generador de números pseudoaleatorios son las que aparecen a continuación:

- Producir muestras según una distribución uniforme en el intervalo $[0,1)$.
- Pasar los contrastes de aleatoriedad e independencia más usuales.
- Presentar una longitud de ciclo tan grande como se requiera.

Mientras que algunas características deseables son:

- Velocidad.
- Ocupar poca memoria.

Veamos a continuación algunos ejemplos de generadores de números pseudoaleatorios.

Método de los cuadrados medios

Es el primer método de generación de números aleatorios en ordenador y fue propuesto por John Von Neumann [5] en 1946.

El procedimiento es el siguiente:

1. Se parte de un número entero, x_0 , la semilla, de $2n$ cifras.
2. Se eleva al cuadrado, obteniendo un número de $4n$ cifras. En caso de que no se alcance dicho número de cifras se completaría con ceros a la izquierda.
3. Se considera x_1 el número entero constituido por las $2n$ cifras centrales.
4. Se eleva al cuadrado x_1 y se repite el mecanismo anterior tantas veces como sea preciso.
5. En último lugar se toman los números $u_i = \frac{x_i}{10^{2n}}$, contenidos en el intervalo $(0,1)$.

Algunos inconvenientes que presenta este método son la fuerte tendencia a degenerar a cero rápidamente y la repetición cíclica de los números después de una secuencia corta.

Ejemplo 1.1.

$$x_0 = 5129 \Rightarrow x_0^2 = 26|3066|41 \Rightarrow x_1 = 3066 \Rightarrow u_1 = 0,3066$$

$$x_1 = 3066 \Rightarrow x_1^2 = 9|4003|56 \Rightarrow x_2 = 4003 \Rightarrow u_2 = 0,4003$$

$$x_2 = 4003 \Rightarrow x_2^2 = 16|0240|09 \Rightarrow x_3 = 240 \Rightarrow u_3 = 0,0240$$

$$x_3 = 240 \Rightarrow x_3^2 = 00|0576|00 \Rightarrow x_4 = 576 \Rightarrow u_4 = 0,0576$$

$$x_4 = 576 \Rightarrow x_4^2 = 00|3317|76 \Rightarrow x_5 = 3317 \Rightarrow u_5 = 0,3317$$

$$x_5 = 3317 \Rightarrow x_5^2 = 11|0024|89 \Rightarrow x_6 = 24 \Rightarrow u_6 = 0,0024$$

$$x_6 = 24 \Rightarrow x_6^2 = 00|0005|76 \Rightarrow x_7 = 5 \Rightarrow u_7 = 0,0005$$

$$x_7 = 5 \Rightarrow x_7^2 = 00|0000|25 \Rightarrow x_8 = 0 \Rightarrow u_8 = 0$$

Métodos congruenciales

Actualmente, los principales generadores de números pseudoaleatorios son los generadores congruenciales lineales. Estos métodos fueron introducidos por Lehmer [6] en 1951. Estos procesos siguen el siguiente esquema:

$$x_n = ax_{n-1} + c, \quad \text{mód } m. \quad (2.1)$$

Los elementos a , m y c son números enteros positivos, y reciben el nombre de multiplicador, módulo e incremento, respectivamente.

En caso de que $c = 0$ en (2.1) el generador se denomina multiplicativo. En caso contrario, se trata de un generador mixto.

Finalmente, la sucesión viene dada por $u_i = \frac{x_i}{m}$.

Para asegurar la aleatoriedad de la sucesión, la semejanza a una distribución uniforme en el intervalo $(0,1)$ y que la longitud del ciclo sea la mayor posible es necesario realizar una conveniente elección de los parámetros a , c y m .

En 1962, Hull y Dobell [2] ofrecieron una condición suficiente para garantizar que el generador sea de ciclo completo, es decir, que la longitud del ciclo coincida con m .

Teorema 2.1. *Un generador congruencial tiene periodo completo si se cumplen las siguientes condiciones:*

- m y c son primos relativos.
- Si q es un número primo que divide a m , entonces q divide a $a - 1$.
- Si 4 divide a m , entonces 4 divide a $a - 1$.

Demostración:

Si $a = 1$ y c es primo relativo con m , el periodo es claramente m .

Consideremos $a \neq 1$. A partir de la expresión $x_i = ax_{i-1} + c$ con $i = 1, 2, \dots, n-1$ podemos obtener por inducción:

$$x_n \equiv a^n x_0 + \frac{a^n - 1}{a - 1} c \quad \text{mód } m.$$

Estamos interesados en el valor más pequeño de n tal que $x_n = x_0$, es decir, aquel para el que:

$$\frac{(a^n - 1)(x_0(a - 1) + c)}{a - 1} \equiv 0 \quad \text{mód } m. \quad (2.2)$$

Por las condiciones del teorema, $x_0(a - 1) + c$ es primo relativo con m . Por esta razón, la congruencia (2.2) sería equivalente a encontrar el menor n tal que:

$$\frac{a^n - 1}{a - 1} \equiv 0 \quad \text{mód } m. \quad (2.3)$$

El siguiente paso es mostrar que el menor n es m , cuando a satisface las condiciones del teorema. Vamos a distinguir casos:

- $m = p^\alpha$, $\alpha \in \mathbb{Z}^+$, p primo impar.

Para $\alpha = 1$, por la condición (ii) este caso se reduce al caso trivial $a = 1$.

Tomemos a partir de ahora $\alpha \geq 2$.

Como a satisface las condiciones del teorema y $a \neq 1$, podemos afirmar:

$$a = 1 + kp^\beta. \quad (2.4)$$

donde k y p son primos relativos, $k \neq 0$ y $\beta \in \mathbb{Z}^+$. Ahora, para ver que $n = p^\alpha$ satisface (2.3), vamos a sustituir este valor de n en dicha expresión teniendo en cuenta la igualdad (2.4):

$$\frac{a^n - 1}{a - 1} = p^\alpha + \frac{p^\alpha(p^\alpha - 1)}{1 \cdot 2}kp^\beta + \frac{p^\alpha(p^\alpha - 1)(p^\alpha - 2)}{1 \cdot 2 \cdot 3}(kp^\beta)^2 + \dots + (kp^\beta)^{p^\alpha - 1}. \quad (2.5)$$

A continuación, probaremos que esta expresión es divisible por p^α . Esto implica que cada término de la expresión es divisible por p^α . Para ello, vamos a reescribir el término j -ésimo como sigue:

$$\frac{p^\alpha}{j} \left[\frac{(p^\alpha - 1)(p^\alpha - 2) \dots (p^\alpha - j + 1)}{1 \cdot 2 \cdot \dots \cdot (j - 1)} \right] k^{j-1} p^{(j-1)\beta}. \quad (j > 1)$$

La parte de este término que precede al factor k^{j-1} es un coeficiente binomial, por lo que es un número entero. Esto último significa que el denominador de esta parte debe dividir al numerador. Además, el cociente entre paréntesis es de nuevo un coeficiente binomial y así, un número entero. De esta forma, j es el único factor en el denominador que quizás "necesite" algo del factor p^α del numerador. No obstante, el número de veces que el factor p puede aparecer en j es menor que:

$$\frac{j}{p} + \frac{j}{p^2} + \frac{j}{p^3} + \dots = \frac{j}{p-1}, \quad (2.6)$$

lo cual es menor que $j - 1$. Pero el factor p aparece como mínimo $j - 1$ veces en el elemento $p^{(j-1)\beta}$ ya que $\beta \geq 1$. Luego el factor p^α no es necesario para que j divida al numerador.

Así, hemos visto que efectivamente cada término del lado derecho de la igualdad (2.5) es divisible por p^α . Esto implica que la identidad (2.3) es satisfecha por $n = p^\alpha$, al menos bajo las condiciones establecidas.

Para acabar este caso vamos a probar que ningún valor de n más pequeño que p^α va a satisfacer (2.3).

En primer lugar, es fácil demostrar que un valor de n va a satisfacer (2.3) si y sólo si es un múltiplo del menor valor de n .

Ahora, sabiendo que $n = p^\alpha$ satisface (2.3), consideraremos sólo aquellos valores de n que sean potencias de p . En particular, para demostrar que ningún valor de n más pequeño que p^α va a satisfacer (2.3) basta mostrar que para $n = p^{\alpha-1}$ no se verifica.

Comenzaremos sustituyendo $n = p^{\alpha-1}$ en (2.3) y considerando la expresión de a en (2.4):

$$\frac{a^n - 1}{a - 1} = p^{\alpha-1} + \frac{p^{\alpha-1}(p^{\alpha-1} - 1)}{1 \cdot 2}kp^\beta + \frac{p^{\alpha-1}(p^{\alpha-1} - 1)(p^{\alpha-1} - 2)}{1 \cdot 2 \cdot 3}(kp^\beta)^2 + \dots + (kp^\beta)^{p^{\alpha-1} - 1}. \quad (2.7)$$

Vamos a demostrar que la sumatoria de la derecha de la igualdad no es divisible por p y por tanto no se verifica (2.3).

El primer término, $p^{\alpha-1}$, es obviamente no divisible por p^α . Así, para probar la no divisibilidad de la sumatoria basta con probar que el resto de sumandos sí son divisibles por p^α . El argumento que se sigue es el mismo dado a través del término j -ésimo en (2.5), excepto por un matiz. En este caso tenemos $p^{\alpha-1}$ en lugar de p^α en el coeficiente binomial. Por ello, necesitamos otro factor de p . Este elemento se encuentra en $p^{(j-1)\beta}$ con la condición de que hagamos uso de la premisa de que p es impar, lo cual implica que (2.6) es menor o igual que $j-2$. Así queda completada la prueba para $m = p^\alpha$ siendo p un primo impar.

- $m = 2^\alpha$.

Para $\alpha = 1$ resulta de nuevo trivial.

El caso donde $\alpha \geq 2$ se demuestra de forma análoga al caso anterior donde $m = p^\alpha$ salvo por una pequeña diferencia. Esta discrepancia reside en que el elemento β de la igualdad (2.4) debe cumplir que $\beta > 1$. Esta restricción es necesaria a la hora de probar que 2^α es el valor más pequeño que satisface (2.3).

Así, el teorema queda establecido cuando m es una potencia de un primo.

- m compuesto.

Entonces podemos expresar m de la forma:

$$m = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_s^{\alpha_s}.$$

Y así, por las condiciones del teorema, a viene dado por:

$$a = 1 + k p_1^{\beta_1} p_2^{\beta_2} \dots p_s^{\beta_j},$$

con p_i primo, $\alpha_i \in \mathbb{Z}^+$, $k \neq 0$ primo relativo con m y $\beta_i \geq 1$ o si $p_i = 2$ y $\alpha_i \geq 2$, $\beta_i \geq 2$.

El argumento para probar este caso se sigue de la misma forma vista en los casos anteriores.

Por otra parte, es obvio que se requiere que el tamaño de m sea grande para así lograr un periodo largo y una gran densidad en $(0,1)$. No obstante, la operación de dividir por m y determinar el resto es relativamente lenta. Una elección acertada para m computacionalmente es $m = 2^k$, siendo k -bits el tamaño de palabra del microprocesador¹. Esta operación permite aprovechar el desbordamiento de datos para no tener que realizar en sí la operación dada por el generador. Esto es, si disponemos de una máquina de k -bits, el mayor entero que puede ser representado es $2^k - 1$. Si se quisiera representar T un entero mayor, ocupando $t > k$ dígitos binarios, se perderían los $t - k$ dígitos binarios más a la izquierda, y permanecerían k dígitos los cuales coinciden con T módulo 2^k .

Ejemplo 1.2.

¹En el ámbito de la informática, una palabra es una cadena finita de bits. De aquí, el tamaño de la palabra designa el número de bits de la misma.

Consideremos el generador congruencial $x_n = 5x_{n-1} + 3$ módulo 16, con $x_9 = 14$. Entonces $5x_9 + 3 = 73$, que en binario se expresa como 1001001. Dado que la capacidad es de 4 bits (ya que $16 = 2^4$), el desbordamiento de datos provoca la pérdida de los tres primeros dígitos, quedando 1001 que es la representación binaria de $x_{10} = 9$.

Medidas estadísticas de la calidad de un generador de números aleatorios

Esta sección está dedicada a la presentación de contrastes estadísticos útiles a la hora de comprobar ciertas discrepancias de las muestras generadas con respecto a las hipótesis de aleatoriedad: independencia y ajuste a una distribución $U(0,1)$.

Comprobación de la uniformidad

■ **Contraste chi-cuadrado**

Para la realización de este método de validación es necesario seguir los siguientes pasos:

1. Se divide el intervalo $(0,1)$ en k clases disjuntas de la misma amplitud, $\frac{1}{k}$. Para cada clase C_j , se cuenta el número de elementos O_j que pertenecen a dicha clase.
2. Se considera el estadístico:

$$T = \sum_{j=1}^k \frac{(O_j - \frac{n}{k})^2}{\frac{n}{k}}$$

siendo n el número de elementos de la sucesión de números pseudoaleatorios. El estadístico T se distribuye asintóticamente según una distribución chi cuadrado con $k - 1$ grados de libertad.

3. Para un nivel de significación α , la región crítica viene dada por:

$$\{\chi^2 \leq \chi_{k-1, \frac{\alpha}{2}}^2\} \cup \{\chi^2 \geq \chi_{k-1, 1-\frac{\alpha}{2}}^2\}.$$

Por tanto, se aceptará la uniformidad si T no pertenece a la región crítica.

Para poder aplicar este test es indispensable que $\frac{n}{k} > 5$.

■ **Contraste de Kolmogorov-Smirnov**

El test chi-cuadrado está limitado cuando el tamaño muestral es pequeño. Como alternativa a dicho contraste se puede aplicar el test de Kolmogorov-Smirnov, el cual no precisa del agrupamiento de los datos y es útil para muestras pequeñas.

Este test consiste en:

1. Ordenar los valores de la muestra en orden creciente, $X_{(1)} \leq X_{(2)} \leq \dots \leq X_{(n)}$.
2. Calcular la función de distribución empírica de la muestra, $S_n(x)$:

$$S_n(x) = \frac{1}{n} \sum_{i=1}^n I_{(-\infty, x)}(X_i).$$

siendo I la función indicadora.

3. Calcular la discrepancia máxima entre la distribución empírica y la teórica a través del estadístico:

$$D_n = \sup_{x \in \mathbb{R}} |S_n(x) - F_0(x)|.$$

donde F_0 es la función de distribución de una variable aleatoria uniforme en el intervalo $(0, 1)$. Concretando, este estadístico puede expresarse como:

$$D_n = \max_{1 \leq i \leq n} \left\{ \max \left\{ \left| \frac{i}{n} - X_{(i)} \right|, \left| \frac{i-1}{n} - X_{(i)} \right| \right\} \right\}.$$

4. La distribución de D_n está tabulada, de forma que la región crítica es $[D_{n,1-\alpha}, \infty)$.

■ Contraste de los pares consecutivos no solapados

A partir de la muestra u_1, u_2, \dots, u_n , para n par, la realización de este test consta de seis fases:

1. Se divide el intervalo $(0,1)$ en k clases disjuntas de la misma amplitud.
2. Se discretiza la muestra inicial asociando a cada u_i el índice de la clase a la que pertenece. Sea y_1, \dots, y_n la muestra discretizada.
3. Se agrupan los elementos de la muestra discretizados en pares consecutivos no solapados, es decir, $(y_1, y_2), (y_3, y_4), \dots, (y_{n-1}, y_n)$.
4. Se cuenta el número de veces que aparece el par (i, j) , lo cual vendrá denotado por O_{ij} . Si la hipótesis de uniformidad es cierta, entonces O_{ij} sigue una distribución binomial de parámetros $\frac{n}{2}$ y $p = \frac{1}{k^2}$. Así, $E[O_{ij}] = E_{ij} = \frac{n}{2k^2}$.
5. Se calcula el estadístico:

$$T = \sum_{i=1}^k \sum_{j=1}^k \frac{(O_{ij} - E_{ij})^2}{E_{ij}}.$$

6. Se rechaza la hipótesis de uniformidad si $T > \chi_{k-1,1-\alpha}^2$.

Debido a que se mide la uniformidad en la región $(0, 1) \times (0, 1)$, de forma indirecta, se comprueba la independencia ya que este test descubriría patrones como por ejemplo que números próximos al cero estén seguidos por otros números también cercanos al cero.

Comprobación de la aleatoriedad

Los contrastes de aleatoriedad sirven para determinar si los elementos de una determinada secuencia pueden considerarse independientes entre sí.

■ Test de rachas

A partir de la muestra u_1, \dots, u_n se construye una sucesión constituida por 0 y 1 teniendo en cuenta lo siguiente:

- En la posición i se coloca un 0 si $u_{i+1} < u_i$.
- En la posición i se coloca un 1 si $u_{i+1} > u_i$.

Cada secuencia de ceros o unos genera una racha.

Para n suficientemente grande, el número de rachas R sigue una distribución normal de parámetros:

$$\mu = \frac{2n-1}{3}, \quad \sigma^2 = \frac{16n-29}{90}.$$

El estadístico de contraste vendría dado por:

$$Z = \frac{R - \mu}{\sigma}.$$

Por tanto, para un nivel de significación α se rechaza la hipótesis de aleatoriedad si $|Z| > Z_{\frac{\alpha}{2}}$.

■ Test de huecos

En primer lugar, consideremos un intervalo de extremos a, b con $0 \leq a < b \leq 1$. Asimismo, se denominará hueco a cualquier serie de números en la secuencia inicial que:

- No están dentro del intervalo $[a, b)$.
- Están comprendidos entre dos números que sí pertenecen a $[a, b)$.

El número de elementos en el hueco se llamará longitud del hueco. Los distintos valores que podrá tomar dicha longitud estarán acotados por 0 y t^* , siendo este último un valor fijo establecido a nuestra conveniencia. En caso de que este valor sea superado, la longitud del hueco se considerará como t^* .

Una vez hemos determinado los huecos, se cuentan las frecuencias observadas O_i , $i = 0, \dots, t^*$ y las frecuencias esperadas $E_i = n \cdot p_i$, para $i = 0, \dots, t^*$. La probabilidad de obtener un hueco de longitud i , esto es, p_i se determina mediante $p_i = p(1-p)^i$ para $i = 1, 2, \dots, t^*$ donde $p = b-a$ es la probabilidad de que un número al azar esté en $[a, b)$.

En último lugar, se calcula el estadístico:

$$\chi^2 = \sum_{i=0}^{t^*} \frac{(O_i - E_i)^2}{E_i}.$$

A partir de este valor, rechazaremos la independencia de la muestra si se tiene que $\chi^2 \in \{\chi^2 \leq \chi_{t^*, \frac{\alpha}{2}}^2\} \cup \{\chi^2 \geq \chi_{t^*, 1-\frac{\alpha}{2}}^2\}$.

2.2 Generación de muestras aleatorias

En los problemas de simulación surgen habitualmente modelos que pueden involucrar variables aleatorias que sigan distribuciones genéricas, por lo que es necesario definir métodos que permitan simular (es decir, obtener muestras artificiales) variables sea cual sea su distribución. No obstante, en el caso de las distribuciones más habituales, existen métodos específicos que hacen uso de algunas de sus propiedades.

Algunos de los métodos para la obtención de variables aleatorias más conocidos son el método de la transformada inversa, el método de composición, la convolución y el método de aceptación-rechazo.

Método de la transformada inversa

Sea X una variable aleatoria con función de distribución F^2 . Dado que F es una función no decreciente podemos definir su inversa como:

$$F^{-1}(y) = \inf\{x : F(x) \geq y\}, \quad 0 \leq y \leq 1.$$

Consideremos $U \sim U(0,1)$. Entonces la función de distribución de la transformada inversa $F^{-1}(U)$ viene dada por:

$$P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x).$$

De esta forma, si nuestro objetivo es obtener una variable aleatoria X con función de distribución F a través del método de la transformada inversa los pasos a seguir son:

1. Considerar $U \sim U(0,1)$.
2. Configurar $X = F^{-1}(U)$.

Este método requiere que la inversa de la función de distribución F pueda determinarse analíticamente o mediante algún algoritmo. Algunas distribuciones en las que podemos aplicar este método son las distribuciones Exponencial, Uniforme y Cauchy. Sin embargo, en muchas otras distribuciones de probabilidad resulta muy complejo o incluso imposible encontrar dicha inversa, como por ejemplo la Normal.

Una ventaja de esta técnica es su posible aplicación en distribuciones absolutamente continuas y discretas. El procedimiento de la transformada inversa para distribuciones discretas requiere de las siguientes fases:

1. Generar $U \sim U(0,1)$.
2. Encontrar el entero positivo más pequeño que verifique $F(x_k) \geq U$ y devolver $X = x_k$.

²La referencia bibliográfica utilizada ha sido [3]

Método de composición

Esta técnica suele aplicarse cuando la función de distribución F a partir de la que se desea generar la muestra se puede expresar como una combinación convexa de otras funciones de distribución F_1, F_2, \dots . Concretamente, asumiremos que para todo x , $F(x)$ puede escribirse como:

$$F(x) = \sum_{j=1}^{\infty} p_j F_j,$$

donde $p_j \geq 0$, $\sum_{j=1}^{\infty} p_j = 1$ y cada F_j es una función de distribución. Equivalentemente, si X tiene función de densidad f , puede expresarse de la siguiente manera:

$$f(x) = \sum_{j=1}^{\infty} p_j f_j(x),$$

siendo cada f_j función de densidad. Para la variante discreta, el procedimiento es análogo.

El algoritmo de este método consiste en:

1. Generar un entero positivo aleatorio J tal que $P(J = j) = p_j$ para $j = 1, 2, \dots$
2. Devolver X con función de distribución F_j .

El primer paso consiste en elegir la función de distribución F_j con probabilidad p_j , lo cual puede lograrse mediante el método de la transformada inversa discreto. Dado $J = j$, obtener X en el segundo paso es independiente de la elección de J . Así, condicionando sobre el valor de J que hemos generado en el paso 1, obtenemos que X tiene función de distribución F :

$$P(X \leq x) = \sum_{j=1}^{\infty} P(X \leq x | J = j) P(J = j) = \sum_{j=1}^{\infty} F_j(x) p_j = F(x).$$

Convolución

Este método asume que existen Y_1, \dots, Y_m , para $m \in \mathbb{N}$ fijo, variables aleatorias independientes e idénticamente distribuidas tales que $Y_1 + Y_2 + \dots + Y_m$ tiene la misma distribución que X . De esta forma, podemos esquematizar este procedimiento de la siguiente forma:

1. Generar Y_1, Y_2, \dots, Y_m iid con función de distribución G .
2. Devolver $X = Y_1 + \dots + Y_m$.

El método de convolución es bastante simple, siempre y cuando podamos generar las pertinentes variables aleatorias Y_j fácilmente. Además, dependiendo de los parámetros de la distribución de X puede no resultar la técnica más idónea o eficiente.

Método de aceptación-rechazo

Los dos métodos anteriores podríamos considerarlos como técnicas directas, en el sentido de que tratan directamente con la distribución o variable aleatoria deseada. El proceso de aceptación-rechazo no es tan directo y resulta de gran utilidad cuando los métodos previos no se pueden aplicar.

El fundamento teórico de este método se resume en el siguiente teorema:

Teorema 2.2. Sea X una variable aleatoria con función de densidad $f_X(x)$, $x \in I \subset \mathbb{R}$ tal que puede factorizarse como

$$f_X(x) = Cg(x)h(x), \quad (2.8)$$

con $C \in \mathbb{R}$, $C \geq 1$, $0 \leq g(x) \leq 1$ y $h(x)$ función de densidad en I . Sea U una variable con distribución $\mathcal{U}(0,1)$ e Y una variable con densidad $h(y)$ en I . Entonces,

$$f_Y(x|U \leq g(Y)) = f_X(x).$$

Demostración:

La densidad condicionada viene dada por:

$$f_Y(x|U \leq g(Y)) = \frac{P\{U \leq g(Y)|Y = x\}h(x)}{P\{U \leq g(Y)\}}.$$

De aquí,

$$P\{U \leq g(Y)|Y = x\} = P\{U \leq g(x)\} = g(x),$$

puesto que U es uniforme en $(0,1)$. Además,

$$\begin{aligned} P\{U \leq g(Y)\} &= \int_I P\{U \leq g(Y)|Y = x\}h(x)dx \\ &= \int_I g(x)h(x)dx \\ &= \int_I \frac{f_X(x)}{C}dx = \frac{1}{C}, \end{aligned}$$

a partir de la fórmula (2.8). Por lo tanto, tenemos que

$$f_Y(x|U \leq g(Y)) = \frac{g(x)h(x)}{1/C} = Cg(x)h(x) = f_X(x).$$

■

De esta forma, las distintas fases de este método son:

1. Generar valores de acuerdo con la distribución $h(x)$.
2. Generar $U \sim U(0,1)$.
3. Si $U \leq g(Y)$, se devuelve $X = Y$. En caso contrario, volver al paso 1 y repetir la secuencia de pasos.

La eficiencia de la técnica de aceptación-rechazo se define como la probabilidad de aceptación, es decir,

$$P\{U \leq g(Y)\} = \frac{1}{C}.$$

Esta propiedad decrece rápidamente a medida que el número de dimensiones aumenta.

En último lugar, veremos cómo obtener muestras que sigan distribuciones concretas como la Exponencial, la Gamma, la Normal o la Poisson [4].

Casos particulares

Distribución Exponencial $E(\beta)$

Obtener una muestra de esta distribución es posible a través de la siguiente variante del método de la transformada inversa:

1. Generar $U \sim U(0, 1)$.
2. Devolver $X = -\beta \ln U$.

Aplicar el método de la inversa implicaría usar $1 - U$ en el paso 2. No obstante, dado que $1 - U$ y U tienen la misma distribución $U(0, 1)$ es posible realizar la modificación.

Distribución Gamma $\Gamma(\alpha, \beta)$

Dada una variable aleatoria $X \sim \Gamma(\alpha, \beta)$, su densidad es:

$$f(x) = \frac{1}{\Gamma(\alpha, \beta)} x^{\alpha-1} e^{-\beta x} \quad x > 0,$$

siendo $\Gamma(\alpha, \beta)$ la función gamma.

El método de inversión no es aplicable directamente ya que generalmente no podemos obtener la expresión analítica de la inversa de la función de distribución. Como solución a este problema se recurre a que si $X_1, \dots, X_k \sim E(\beta)$, entonces:

$$Y = \sum_{i=1}^k X_i \sim \Gamma\left(k, \frac{1}{\beta}\right).$$

Por otro lado, dada $X \sim \Gamma(\alpha, 1)$, podemos obtener para cualquier $\beta > 0$ una variable aleatoria $X' \sim \Gamma(\alpha, \beta)$ considerando $X' = \beta X$. Como consecuencia, sólo debemos restringirnos a generar elementos de una $\Gamma(\alpha, 1)$. Puesto que la configuración $\Gamma(1, 1)$ es una exponencial con media 1 (podemos aplicar el método anterior), sólo debemos estudiar los casos en los que $0 < \alpha < 1$ y $\alpha > 1$. Aplicando el método de aceptación-rechazo, el algoritmo a seguir en cada caso es el siguiente:

- $0 < \alpha < 1$.
 1. Generar $U_1 \sim U(0, 1)$ y tomar $P = bU_1$, siendo $b = \frac{e+\alpha}{e}$. Si $P > 1$, pasar al paso 3. En caso contrario, continuar con el paso 2.

2. Considerar $Y = P^{\frac{1}{\alpha}}$ y generar $U_2 \sim U(0, 1)$. Si $U_2 \leq e^{-Y}$, devolver $X = Y$. Si no se tiene lo anterior, volver al paso 1.
3. Sea $Y = -\ln[\frac{b-P}{\alpha}]$ y generar $U_2 \sim U(0, 1)$. Si $U_2 \leq Y^{\alpha-1}$, devolver $X = Y$. De lo contrario, tornar al paso 1.

- $\alpha > 1$.

En este caso, es preciso conocer las constantes:

$$a = \frac{1}{\sqrt{2\alpha - 1}}, \quad b = \alpha - \ln 4, \quad q = \alpha + \frac{1}{a},$$

$$\theta = 4,5, \quad d = 1 + \ln \theta.$$

Así, los pasos son:

1. Generar U_1 y U_2 iid $U(0, 1)$.
2. Denotar $V = a \ln[U_1/(1 - U_1)]$, $Y = \alpha e^V$, $Z = U_1^2 U_2$ y $W = b + qV - Y$.
3. Si $W + d - \theta Z \geq 0$, devolver $X = Y$. En caso contrario, continuar con el paso 4.
4. Si $W \geq \log Z$, devolver $X = Y$. De lo contrario, repetir el paso 1.

El paso 3 de este último algoritmo (si es satisfecho) tiene como objetivo evitar calcular el logaritmo del paso 4 del método de aceptación-rechazo usual.

Distribución Normal $\mathcal{N}(0, 1)$

Existen diferentes formas de generar muestras que sigan una distribución normal. Una primera estrategia sería recurrir al teorema central del límite. Así, si consideramos U_1, \dots, U_n variables aleatorias independientes con distribución $U(0, 1)$, entonces:

$$Z = \sum_{i=1}^n U_i,$$

se aproxima a una distribución normal a medida que n crece.

La segunda técnica que se puede utilizar es definiendo a partir de U_1 y U_2 dos variables independientes con distribución $U(0, 1)$, las variables:

$$Z_1 = \sqrt{-2 \log U_1} \cos(2\pi U_2),$$

$$Z_2 = \sqrt{-2 \log U_1} \sin(2\pi U_2).$$

Estas variables son independientes y siguen una distribución normal $\mathcal{N}(0, 1)$. Además, tal y como podemos observar, se trata de un método exacto.

Una mejora de este método que elimina los cálculos trigonométricos es el **método polar**:

1. Generar U_1 y U_2 iid $U(0, 1)$. Considerar $V_i = 2U_i - 1$ para $i = 1, 2$ y sea $W = V_1^2 + V_2^2$.
2. Si $W > 1$, volver al paso anterior. De lo contrario, considerar $Y = \sqrt{(-2 \ln W)/W}$, $X_1 = V_1 Y$ y $X_2 = V_2 Y$. De aquí, X_1 y X_2 son iid $\mathcal{N}(0, 1)$.

Distribución de Poisson $\mathcal{P}(\lambda)$

La función masa de probabilidad de una distribución de Poisson es:

$$P(X = x) = \frac{e^{-\lambda} \lambda^x}{x!} \quad x = 0, 1, \dots$$

A medida que λ crece,

$$Z = \frac{X - \lambda + 0,5}{\sqrt{\lambda}} \sim \mathcal{N}(0, 1).$$

De esta forma,

$$X = \text{máx}\{0, [\lambda + Z\sqrt{\lambda} - 0,5]\} \sim \mathcal{P}(\lambda).$$

Otra forma de obtener una muestra de una distribución $\mathcal{P}(\lambda)$ es a través de la siguiente técnica:

1. Sea $a = e^{-\lambda}$, $b = 1$ y $i = 0$.
2. Generar $U_{i+1} \sim U(0, 1)$ y actualizar $b = bU_{i+1}$. Si $b < a$, devolver $X = i$. En caso contrario, continuar con el paso 3.
3. Reemplazar i por $i + 1$ y volver al paso 2.

2.3 Estimación por Monte Carlo

En la presente sección vamos a tratar una técnica para la resolución de integrales. Esta estrategia nos permitirá calcular esperanzas y varianzas.

Nuestro problema consiste en resolver la integral de una función continua $g(X)$ dada por:

$$\mathcal{I} = \int_0^1 g(X) dX.$$

Esta integral puede interpretarse como el cálculo de la esperanza de la función $g(X)$ para $X \sim U(0, 1)$. Ahora, consideramos X_1, X_2, \dots, X_n una muestra de variables aleatorias independientes y con una distribución uniforme en el intervalo $[0, 1]$ obtenida por medio de alguno de los métodos vistos anteriormente. Así, en base a la Ley de los Grandes Números podemos aproximar la anterior integral de la siguiente forma:

$$\mathcal{I} = E[g(X)] \simeq \frac{1}{n} \sum_{i=1}^n g(X_i).$$

En aquellos casos en los que la integral en cuestión tenga como intervalo de integración $[a, b]$, podemos trasladar esta situación al intervalo unidad mediante el cambio de variable:

$$z = \frac{x - a}{b - a}.$$

Asimismo, podemos estudiar integrales impropias de la forma:

$$\int_0^{\infty} g(x) dx.$$

Realizando el cambio de variable:

$$z = \frac{1}{x+1}.$$

Tal y como mencionamos al inicio de la sección esta herramienta nos permitirá calcular esperanzas y varianzas teniendo en cuenta que estas últimas verifican la siguiente igualdad:

$$\text{Var}[X] = E[X^2] - E[X]^2.$$

Este método de integración es conveniente para el cálculo de integrales multidimensionales donde otras técnicas como la regla de Simpson o la regla del trapecio dejan de ser eficientes.

Teoría de colas

Históricamente, el inicio de la teoría de colas está marcado por los trabajos del matemático-ingeniero Erlang [1], quien en 1909 publicó *La teoría de probabilidades y las conversaciones telefónicas*. Este estudio ofrecía una aplicación de técnicas existentes en probabilidad al problema de determinar el número óptimo de líneas telefónicas en una centralita. Más tarde, en 1927 y 1928, destacan los trabajos realizados por Molina y Thornton.

La **teoría de colas** es una disciplina que tiene por objeto el estudio y análisis de situaciones en las que existen entidades que solicitan un determinado servicio, de tal forma que este servicio no puede ser satisfecho inmediatamente, por lo que se provocan esperas o colas.

Tal y como queda reflejado en la definición anterior, el área de aplicación de la teoría de colas es enorme: esperas para ser atendidos en establecimientos comerciales, esperas para facturar en un aeropuerto, esperas para ser atendidos en hospitales, entre muchas otras.

Desde el punto de vista matemático, esta teoría ofrece conceptos y expresiones que han sido deducidas y que permiten calcular los parámetros característicos del sistema en estudio. Sin embargo, estas relaciones consideran que el número de llegadas y de servicios sigue la distribución de Poisson. Cuando las distribuciones de probabilidad de los tiempos de servicio son diferentes (Normal, Gamma, Beta, Chi-cuadrado, T de Student, etc.), las fórmulas analíticas deducidas no funcionan, por lo que habrá que recurrir a la **simulación de Monte Carlo**.

Con todas estas herramientas podremos dar solución a problemas como determinar el número de camas en una UCI, calcular el número de surtidores en una gasolinera o bien decidir el número de cajas registradoras a instalar en un supermercado.

3.1 Conceptos básicos

Comencemos definiendo los diferentes términos que nos podemos encontrar a la hora de trabajar con la teoría de colas.

- Fuente de entrada o población potencial.

Representa el conjunto de individuos que pueden llegar a demandar el servicio en cuestión. Puede ser finita o infinita. A pesar de que la posibilidad infinita no se ajusta a la realidad, permite resolver de una manera más sencilla aquellas situaciones en las que la población es finita, pero de un tamaño considerable.

- Cliente.

Es cualquier entidad de la fuente de entrada que solicita el servicio. Suponiendo que los tiempos de llegada consecutivos son $0 < t_1 < t_2 < \dots$, conocer el patrón de probabilidad por el cual la fuente de entrada genera clientes será importante. Normalmente, se toma como referencia los tiempos entre las llegadas de dos clientes consecutivos: $\tau_k = t_k - t_{k-1}$, estableciendo su distribución de probabilidad. Esta distribución de probabilidad de los τ_k no depende del número de clientes que estén en espera de completar su servicio cuando la fuente de entrada es

infinita. En cambio, si es finita, la distribución de los τ_k variará en función del número de clientes en proceso de ser atendidos.

▪ Capacidad de la cola.

Es el número máximo de clientes que pueden estar esperando a ser atendidos. Al igual que en el caso de la fuente de entrada, esta cantidad puede ser finita o infinita. En caso de ser finita, el sistema rechazará aquellas solicitudes que no quepan, bien negando la nueva petición, o bien eliminando alguna que ya existiese. En algunas ocasiones, para una mayor simplicidad en los cálculos, se considerará infinita. No obstante, aunque en la mayoría de los sucesos reales la capacidad de cola es finita, suponerla infinita no resulta un impedimento si es extremadamente improbable que alcanzado ese número límite en la cola no puedan entrar clientes a la cola.

▪ Disciplina de la cola.

Es la forma en la que los clientes son escogidos para ser atendidos. Las siguientes disciplinas son las más comunes:

- Disciplina FIFO o FCFS, se atiende al cliente según el orden de llegada, esto es, al cliente que haya llegado antes.
- Disciplina LIFO o LCFS, consiste en atender primero al cliente que ha llegado el último.
- Disciplina RSS o SIRO, se seleccionan los clientes de forma aleatoria.
- Disciplina RR, según esta disciplina se otorga un pequeño intervalo de tiempo de servicio a cada cliente de forma secuencial.

▪ Mecanismo de servicio.

Es el procedimiento a seguir para dar servicio a los clientes que lo solicitan. Para determinarlo debemos conocer el número de servidores y la distribución de probabilidad del tiempo que necesita cada servidor para dar un servicio.

▪ Cola.

Es el conjunto de clientes que han solicitado el servicio pero que aún no han pasado al mecanismo de servicio.

▪ Sistema de cola.

Es el conjunto constituido por el mecanismo de servicio, la cola y la disciplina de la misma.

Notación y terminología

A continuación, determinaremos la notación con la que nos vamos a referir a las distintas variables que aparecen en un modelo de la teoría de colas.

$N(t)$: Denota el número de clientes en el sistema en el instante t .

$N_q(t)$: Representa el número de clientes en la cola en el instante t .

$P_n(t)$: Es la probabilidad de que se encuentren, en el instante t , n clientes en el sistema.

s : Denota el número de servidores del mecanismo de servicio.

λ_n : Simboliza el número medio de llegadas al sistema por unidad de tiempo cuando ya se encuentran n clientes en él. Cuando esta cantidad no depende de n suele denotarse por λ dicho valor constante.

μ_n : Denota el número medio de clientes a los que se les finaliza el servicio, por unidad de tiempo, habiendo n clientes en el sistema. Otro nombre que recibe este término es tasas de compleción de servicio. Si todos los servidores cuentan con la misma distribución del tiempo de servicio, se denota por μ al número medio de clientes que puede ser atendido por cada servidor por unidad de tiempo. Consecuentemente, se tiene que $\mu_n = n\mu$ para $n = 1, 2, \dots, s$ y $\mu_n = s\mu$ para $n \geq s$.

ρ : Su nombre es constante de utilización del sistema o intensidad de tráfico. Esta cantidad puede calcularse como:

$$\rho = \frac{\lambda}{s\mu},$$

donde $s\mu$ designa el número medio de clientes que pueden ser atendidos los s servidores cuando todos están ocupados. De esta forma, ρ representa la fracción de recursos del sistema que es consumida por los clientes. Así, de forma intuitiva, resulta esencial que, en estos casos, $\rho < 1$, y cuanto más próximo esté a 1, mayor tráfico soporta el sistema.

Los modelos de colas que expondremos en el posterior capítulo son estacionarios. En estos modelos las distribuciones de probabilidad marginales de los procesos estocásticos $N(t)$ y $N_q(t)$ no varían con el tiempo. En dichas condiciones podemos definir las siguientes nociones:

N : Representa la variable aleatoria que numera el número de clientes en el sistema.

N_q : Denota la variable aleatoria que representa el número de clientes en la cola.

p_n : Designa la probabilidad de que se haya n clientes en el sistema.

L : Denota la esperanza de la variable aleatoria N por lo que representa el número medio de clientes en el sistema.

L_q : Es el número medio de clientes en la cola, matemáticamente $L_q = E[N_q]$.

\mathcal{W} : Se trata de una variable aleatoria que simboliza el tiempo que un cliente pasa en el sistema.

\mathcal{W}_q : Representa el tiempo de espera de un cliente en la cola.

W : Es la esperanza de la variable aleatoria \mathcal{W} , lo cual significa que representa el tiempo medio que un cliente está en el sistema.

W_q : Denota el tiempo medio de espera en la cola para un cliente genérico, es decir, $W_q = E[\mathcal{W}_q]$.

Para clasificar los diferentes tipos de sistemas de colas debemos detallar las características de los elementos que lo componen. Así, Kendall introdujo en 1953 la notación $A/B/s$ donde cada una de estas letras tiene el siguiente significado:

A , representa la distribución del tiempo entre llegadas. Algunas de las abreviaturas más frecuentes son: M (Exponencial), D (Determinística), E_k (Erlang con segundo parámetro k), U (Uniforme), Γ (Gamma) o G (distribución genérica), entre otras.

B , representa la distribución del tiempo de servicio. Las abreviaturas empleadas son las mismas que las destacadas para A .

s , constituye el número de servidores que dispone el sistema. Este elemento puede ser un número entero positivo o bien infinito.

No obstante, es posible extender esta notación mediante la expresión $A/B/s/K/H/Z$ donde las nuevas entradas denotan:

K , es la capacidad de la cola, o, en otras palabras, la longitud máxima de la misma.

H , es el tamaño de la población potencial. Puede tomarse finito o infinito.

Cuando cualquiera de estos dos elementos es omitido en la notación, se consideran infinitos.

Z , es la disciplina en la cola. En caso de omisión, la disciplina que se sigue es FIFO.

Fórmulas de Little

En los modelos en los que la distribución del tiempo entre llegadas y la distribución del servicio es exponencial podemos recurrir a ciertas fórmulas que relacionan algunos de los conceptos vistos anteriormente. Cuando las tasas de llegada son constantes, es decir, $\lambda_n = \lambda$ para cada $n = 0, 1, \dots$ podemos expresar el número medio de clientes en el sistema como:

$$L = \lambda \cdot W.$$

Asimismo, la cantidad L_q cumple:

$$L_q = \lambda \cdot W_q.$$

Estas dos expresiones se conocen como **fórmulas de Little**. Estas igualdades dejan de ser válidas cuando las λ_n no son constantes. Sin embargo, pueden generalizarse de manera que puedan emplearse en dicha situación:

$$L = \bar{\lambda} \cdot W,$$

$$L_q = \bar{\lambda} \cdot W_q,$$

siendo $\bar{\lambda} = \sum_{n=0}^{\infty} \lambda_n p_n$.

Una última relación que resulta fundamental es la dada por:

$$W = W_q + \frac{1}{\mu}.$$

3.2 Modelo M/M/1

Siguiendo la notación vista anteriormente, este modelo define una situación en la que tanto el tiempo entre dos llegadas consecutivas de clientes como el tiempo de servicio siguen una distribución exponencial. Además, el sistema cuenta con un único servidor. El resto de parámetros, al no tener ninguna indicación, se toman por defecto. Esto último significa que no hay restricción respecto al número de clientes que pueden estar en la cola, la disciplina de la cola es FIFO y la población potencial es infinita.

De esta forma, si consideramos que el tiempo entre dos llegadas consecutivas de clientes sigue una exponencial de parámetro λ y el tiempo de servicio una exponencial μ , las tasas de llegada y de servicio vendrán dadas por:

$$\lambda_n = \lambda \quad n = 0, 1, \dots$$

$$\mu_n = \mu \quad n = 0, 1, \dots$$

Por tanto, se tiene que

$$c_n = \frac{\lambda_{n-1} \lambda_{n-2} \cdots \lambda_0}{\mu_n \mu_{n-1} \cdots \mu_1} = \frac{\lambda^n}{\mu^n} = \left(\frac{\lambda}{\mu}\right)^n = \rho^n \quad n = 1, 2, \dots$$

Dado que $\sum_{n=1}^{\infty} c_n = \sum_{n=1}^{\infty} \rho^n$ es una serie geométrica, será convergente si y sólo si $|\rho| < 1$. No obstante, como $\rho > 0$, la convergencia está asegurada si y sólo si $\rho < 1$, lo cual equivale a que el modelo sea estacionario. Otra forma de interpretar esta desigualdad sería mediante la expresión $\lambda < \mu$. Esto significaría que el número medio de clientes que entran en el sistema por unidad de tiempo es menor que el número medio de clientes que podrían ser atendidos por el servidor por unidad de tiempo.

En los próximos desarrollos tendremos en cuenta que $\rho < 1$.

Comencemos calculando la suma de la serie de las c_n :

$$\sum_{n=1}^{\infty} c_n = \sum_{n=1}^{\infty} \rho^n = \frac{\rho}{1-\rho}.$$

Como consecuencia:

$$p_0 = \frac{1}{1 + \sum_{n=1}^{\infty} c_n} = \frac{1}{1 + \frac{\rho}{1-\rho}} = 1 - \rho.$$

Asimismo, se tiene que $p_n = c_n \cdot p_0 = (1 - \rho)\rho^n$ para todo $n \geq 0$. A partir de esto, podemos afirmar que la masa de probabilidad de la variable "número de clientes en el sistema" viene dada por:

$$P(N = n) = p_n = (1 - \rho)\rho^n \quad n = 0, 1, \dots$$

Por lo que $N \sim \mathcal{G}(1 - \rho)$ ¹. De aquí, sabiendo que la media de una variable aleatoria $X \sim \mathcal{G}(\phi)$ es:

$$E[X] = \frac{1 - \phi}{\phi}.$$

Tenemos que:

$$L = E[N] = \frac{1 - (1 - \rho)}{1 - \rho} = \frac{\rho}{1 - \rho}.$$

A continuación, a partir de L calcularemos L_q :

$$\begin{aligned} L_q &= E[N_q] = \sum_{n=0}^{\infty} nP(N_q = n) = \sum_{n=1}^{\infty} nP(N_q = n) \\ &= \sum_{n=1}^{\infty} nP(N = n + 1) = \sum_{n=1}^{\infty} np_{n+1} = \sum_{n=0}^{\infty} np_{n+1} \\ &= \sum_{m=1}^{\infty} (m - 1)p_m = \sum_{m=1}^{\infty} mp_m - \sum_{m=1}^{\infty} p_m \\ &= \sum_{m=1}^{\infty} mp_m - (1 - p_0) = L - (1 - p_0), \end{aligned}$$

¹Recordemos que la distribución Geométrica representa la distribución del número de fracasos antes del primer éxito y que el parámetro $1 - \rho$ indica la probabilidad de éxito.

y sustituyendo las cantidades de L y p_0 :

$$L_q = L - (1 - p_0) = \frac{\rho}{1 - \rho} - (1 - (1 - \rho)) = \frac{\rho^2}{1 - \rho}.$$

Así, una vez hemos calculado L y L_q , aplicando las fórmulas de Little podemos obtener W y W_q :

$$W = \frac{L}{\lambda} = \frac{1}{\mu - \lambda},$$

$$W_q = \frac{L_q}{\lambda} = \frac{\lambda}{\mu(\mu - \lambda)}.$$

Ahora, para obtener más información acerca de la espera de clientes en la cola o en el sistema, debemos calcular la distribución de probabilidad de las variables \mathcal{W} y \mathcal{W}_q . Estas distribuciones van a permitir calcular la probabilidad de cualquier suceso relacionado con el tiempo de estancia en la cola o en el sistema.

Comenzaremos calculando la función de distribución de la variable \mathcal{W} , la cual denotaremos por $W(t)$. Esta distribución la obtendremos a partir de la regla de las probabilidades totales, condicionando a N , y considerando que $\mathcal{W}|_{N=n} \stackrel{d}{=} \Gamma(n+1, \mu)$. Matemáticamente:

$$\begin{aligned} W(t) &= P(\mathcal{W} \leq t) = \sum_{n=0}^{\infty} P(\mathcal{W}|_{N=n})P(N = n) \\ &= \sum_{n=0}^{\infty} \left(\int_0^t \frac{\mu^{n+1}}{n!} x^n e^{-\mu x} dx \right) p_n \\ &= \sum_{n=0}^{\infty} \left(\int_0^t \frac{\mu^{n+1}}{n!} x^n e^{-\mu x} dx \right) \left(1 - \frac{\lambda}{\mu}\right) \left(\frac{\lambda}{\mu}\right)^n \\ &= \int_0^t \mu \left(1 - \frac{\lambda}{\mu}\right) e^{-\mu x} \left(\sum_{n=0}^{\infty} \frac{(\lambda x)^n}{n!} \right) dx \\ &= \int_0^t \mu \left(1 - \frac{\lambda}{\mu}\right) e^{-\mu x} e^{\lambda x} dx = \int_0^t (\mu - \lambda) e^{-(\mu - \lambda)x} dx \\ &= \left[-e^{-(\mu - \lambda)x} \right]_{x=0}^{x=t} = 1 - e^{-(\mu - \lambda)t}. \end{aligned}$$

De esta forma, $\mathcal{W} \stackrel{d}{=} \mathcal{E}(\mu - \lambda)$.

Por otro lado, para obtener la función de distribución de \mathcal{W}_q basta descomponerla de la forma:

$$W_q(t) = P(\mathcal{W}_q \leq t) = P(\mathcal{W}_q = 0) + P(0 < \mathcal{W}_q \leq t).$$

En el segundo término podríamos aplicar la misma estrategia que para el cálculo de la función de distribución de \mathcal{W} , y así tendríamos lo siguiente:

$$W_q(t) = \begin{cases} 1 - \frac{\lambda}{\mu} e^{-(\mu - \lambda)t} & \text{si } t \geq 0 \\ 0 & \text{si } t < 0 \end{cases}$$

A partir de esta expresión, podemos concluir que \mathcal{W}_q es una variable mixta, es decir, una mezcla de continua y discreta. Esta variable toma el valor nulo con probabilidad p_0 y para $t > 0$ tiene una componente continua con función de subdensidad definida por:

$$\frac{\lambda(\mu - \lambda)}{\mu} e^{-(\mu - \lambda)t}$$

3.3 Modelos con distribución arbitraria

Nuestro estudio se centrará en el modelo $M/G/1$ [1] considerando varias colas de este tipo. Cada cola contará con un único servicio, tiempos entre llegadas de clientes consecutivos exponenciales y tiempos de servicio con distribución general, G . En este marco, el parámetro λ seguirá rigiendo los tiempos entre dos llegadas de clientes consecutivos. De esta forma, el número de clientes que llegan al sistema por unidad de tiempo será λ . Debido a que G es una distribución arbitraria, el elemento μ vendrá dado por el inverso del tiempo medio de servicio, esto es:

$$\mu = \frac{1}{E[G]}.$$

El siguiente paso será deducir las fórmulas que nos permitan obtener los valores de los parámetros en cuestión en un sistema como el que estamos tratando. Comenzaremos denotando por X_n a la variable aleatoria que contabiliza el número de clientes en el sistema cuando el n -ésimo cliente lo abandona. Por otro lado, expresaremos por A_n el número de clientes que llegan durante el tiempo de servicio del n -ésimo cliente. Estos dos conceptos podemos relacionarlos mediante la igualdad:

$$X_{n+1} = X_n - U(X_n) + A_{n+1}, \quad (3.1)$$

donde

$$U(X_n) = \begin{cases} 1 & \text{si } X_n > 0 \\ 0 & \text{si } X_n = 0 \end{cases}$$

Si tomamos esperanzas en la expresión anterior, y suponiendo un modelo estacionario, es decir, $L = E[X_n] = E[X_{n+1}]$, se tiene que:

$$L = L - E[U(X_n)] + E[A_{n+1}],$$

con lo cual:

$$E[U(X_n)] = E[A_{n+1}] = P(X_n > 0).$$

Esta cantidad se puede calcular usando la distribución de S_{n+1} , la variable aleatoria que simboliza el tiempo de servicio del $(n+1)$ -ésimo cliente. En el modelo que estamos estudiando esta variable aleatoria tiene distribución general, la cual consideraremos continua y con densidad g :

$$\begin{aligned} E[A_{n+1}] &= E\left[E\left(A_{n+1} | S_{n+1}=t\right)\right] = \int_0^\infty E\left[A_{n+1} | S_{n+1}=t\right] g(t) dt \\ &= \int_0^\infty \lambda t g(t) dt = \lambda E[S_{n+1}] = \frac{\lambda}{\mu} =: \rho \end{aligned}$$

Este mismo resultado se obtendría si la distribución de S_{n+1} fuese discreta o incluso mixta con parte continua y parte discreta. Ahora bien, si elevamos al cuadrado cada miembro de la igualdad (3.1) llegamos a:

$$\begin{aligned} X_{n+1}^2 &= X_n^2 + U(X_n)^2 + A_{n+1}^2 - 2X_n U(X_n) + 2X_n A_{n+1} - 2U(X_n)A_{n+1} \\ &= X_n^2 + U(X_n) + A_{n+1}^2 - 2X_n + 2X_n A_{n+1} - 2U(X_n)A_{n+1}. \end{aligned}$$

Volviendo a tomar esperanzas y dada la independencia entre X_n y A_{n+1} se sigue:

$$\begin{aligned} E[X_{n+1}^2] &= E[X_n^2] + E[U(X_n)] + E[A_{n+1}^2] - 2E[X_n] + 2E[X_n]E[A_{n+1}] - 2E[U(X_n)]E[A_{n+1}] \\ &= E[X_n^2] + \rho + E[A_{n+1}^2] - 2L + 2L\rho - 2\rho^2. \end{aligned}$$

Aplicando la estacionariedad de nuevo:

$$2L - 2L\rho = \rho - 2\rho^2 + E[A_{n+1}^2],$$

$$2L(1 - \rho) = \rho - 2\rho^2 + E[A_{n+1}^2].$$

Por lo que despejando:

$$L = \frac{\rho - 2\rho^2 + E[A_{n+1}^2]}{2(1 - \rho)}.$$

Como podemos observar, para determinar una expresión cerrada de L debemos expresar $E[A_{n+1}^2]$ mediante cantidades conocidas:

$$E[A_{n+1}^2] = \text{Var}[A_{n+1}] + E[A_{n+1}]^2 = \text{Var}[A_{n+1}] + \rho^2,$$

de donde:

$$\begin{aligned} \text{Var}[A_{n+1}] &= E[\text{Var}(A_{n+1}|S_{n+1})] + \text{Var}[E(A_{n+1}|S_{n+1})] = E[\lambda S_{n+1}] + \text{Var}[\lambda S_{n+1}] \\ &= E[\lambda S_{n+1}] + \text{Var}[\lambda S_{n+1}] = \lambda \frac{1}{\mu} + \lambda^2 \sigma_S^2 = \rho + \lambda^2 \sigma_S^2, \end{aligned}$$

siendo σ_S^2 la varianza del tiempo de servicio. Por tanto:

$$E[A_{n+1}^2] = \rho + \lambda^2 \sigma_S^2 + \rho^2.$$

Así, sustituyendo este valor en la expresión de L calculada anteriormente podemos calcularla como:

$$L = \rho + \frac{\rho + \lambda^2 \sigma_S^2}{2(1 - \rho)}.$$

Esta igualdad se conoce como **fórmula de Pollaczek-Khintchine** y a partir de la misma pueden calcularse W , W_q y L_q , puesto que siguen verificándose las fórmulas de Little y la relación entre tiempos medios en el sistema y en la cola.

En último lugar, daremos una idea general de cómo calcular las expresiones de las p_n . Las probabilidades de los distintos estados (p_n) son solución del sistema de infinitas ecuaciones:

$$p_n = p_0 k_n + \sum_{m=1}^{n+1} p_m k_{n-m+1} \quad \text{para } n = 0, 1, 2, \dots$$

para

$$k_n = P(A_m = n) = \begin{cases} \int_0^\infty \frac{e^{-\lambda t} (\lambda t)^n}{n!} g(t) dt, \\ \sum_i \frac{e^{-\lambda s_i} (\lambda s_i)^n}{n!} q_i. \end{cases}$$

donde la integral se calcula para el caso continuo, y la sumatoria cuando S es discreta con masa de probabilidad q_i .

Normalmente resulta difícil resolver este sistema por lo que se suele acudir a las funciones características para determinar la solución. La idea consiste en definir:

$$P(z) = \sum_{n=0}^{\infty} p_n z^n, \quad K(z) = \sum_{n=0}^{\infty} k_n z^n.$$

Estas dos funciones se encuentran relacionadas mediante la igualdad:

$$P(z) = \frac{(1-\rho)(1-z)K(z)}{K(z)-z}.$$

Las constantes del desarrollo en serie de las funciones características y sus derivadas sucesivas en 0 verifican:

$$P^{(n)}(0) = n!p_n, \quad K^{(n)}(0) = n!k_n.$$

donde las constantes k_n son calculables a partir de la distribución del tiempo de servicio. A partir de esta situación, podemos encontrar las fórmulas que nos permitan calcular las p_n .

Simulación

Como hemos podido observar en el modelo $M/G/1$ su resolución analítica se complica debido a que la distribución del tiempo de servicio no es exponencial. Así, para solventar aquellas complicaciones derivadas del hecho de que la distribución del tiempo entre llegadas o la del tiempo de servicio no es exponencial podemos recurrir a la simulación. Esta herramienta permite estimar los parámetros de interés en estos modelos.

Previamente a la resolución del problema a través de la simulación debemos recurrir a la modelización del mismo, es decir, especificar las variables relevantes que parecen regir el sistema y las relaciones existentes entre ellas. De esta forma, recurriremos a la teoría de colas puesto que se trata de una disciplina que tiene como objetivo modelizar experiencias reales en las que existe espera para dar un servicio. Así, supondremos que el problema ya lo tenemos en términos de un modelo de colas, especificando las distribuciones del tiempo entre llegadas y del tiempo de servicio. Esto último precisa de una experimentación real anterior en la cual se observen los tiempos para una serie de clientes de ambas variables.

Para entender mejor todo lo explicado previamente tratemos un modelo de colas sencillo que resolveremos mediante simulación. Este modelo vendrá dado por una cola con un único servidor, capacidad máxima de tres clientes en la cola y fuente de entrada ilimitada.

Tras la experimentación previa, hemos observado que la distribución Weibull, $W(\beta, \alpha)$ se ajusta adecuadamente a los tiempos entre llegadas. La función de densidad es:

$$f(x) = \alpha \beta^\alpha x^{\alpha-1} e^{-(\beta x)^\alpha} \quad \forall x \geq 0.$$

Supongamos que los tiempos entre llegadas, en minutos, siguen una $W(0.1, 0.25)$. De forma análoga, hemos determinado que la duración del tiempo de servicio en minutos se ajusta a una $U(2, 8)$. Así, expresando el modelo según la notación vista anteriormente podemos representarlo como $W/U/1/3$.

Pasemos a continuación a calcular los valores de las cantidades que nos interesan en este modelo. Para ello, podemos seguir el siguiente procedimiento:

1. Comenzaremos suponiendo que no hay ningún cliente en el sistema inicialmente ($N = 0$).
2. El siguiente paso consiste en simular artificialmente el tiempo que tardará en llegar el primer cliente, de acuerdo a una distribución $W(0.1, 0.25)$.
3. Una vez ha transcurrido el tiempo calculado en el paso anterior, debemos actualizar el número de clientes, $N = 1$, y calcular mediante el algoritmo de generación de números aleatorios según una $U(2, 8)$, el tiempo que se tardaría en darle servicio a este cliente. Además, habría que calcular el tiempo necesario para que se produzca la llegada del segundo cliente.
4. Comparar los dos tiempos del paso previo y determinar cuál sucederá antes. Así, lo próximo es dejar transcurrir el tiempo del suceso anterior.
5. Si se produce la llegada del nuevo cliente antes, entonces se tomaría $N = 2$, y se contabilizaría el tiempo durante el cuál ha habido un cliente en el sistema. Igualmente, deberíamos calcular el tiempo de llegada del tercer cliente y continuar el proceso.
6. Si por el contrario el menor tiempo del paso cuatro fuese el de servicio, tomaríamos $N = 0$ y teniendo en cuenta el tiempo durante el cuál ha habido un cliente en el sistema. Dejaríamos transcurrir lo que resta de tiempo hasta que se produce la llegada del segundo cliente y proseguiríamos con el método.
7. En general, este esquema simularía las llegadas de nuevos clientes y los tiempos de servicio, considerando la restricción de que si el instante de llegada de un cliente se produce cuando ya hay tres en cola entonces la limitación de la cola provocaría que ese cliente no pueda entrar, reanudando el cálculo del tiempo de la próxima llegada.

Como podemos observar, en el esquema anterior utilizaremos los métodos vistos sobre números aleatorios y muestras de variables aleatorias.

Resultados

En esta sección proponemos una solución mediante métodos de Monte Carlo al problema de decidir el número óptimo de puestos destinados a consulta, exploración y observación en una Unidad de Urgencias Hospitalaria (UUH).

Nuestro estudio se va a centrar en el funcionamiento de una UUH durante 24 horas y que sigue el siguiente esquema representado en la imagen 4.1.

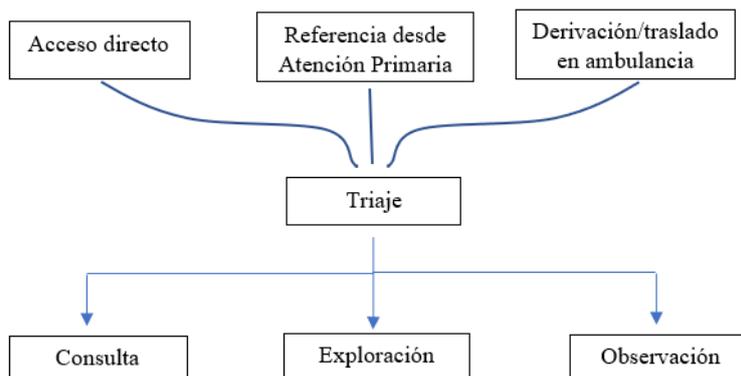


Figura 4.1: Funcionamiento de una UUH.

Nuestro objetivo es calcular la esperanza de la variable aleatoria que determina el tiempo de espera de un paciente en UUH. Denotaremos esta variable por \mathcal{T} . Así, si consideramos $f_{\mathcal{T}}$ su función de densidad la esperanza de \mathcal{T} vendría dada por:

$$E[\mathcal{T}] = \int x f_{\mathcal{T}}(x) dx.$$

Sin embargo, la primera dificultad que encontramos a la hora de resolver esta integral reside en la expresión de $f_{\mathcal{T}}$. Ofrecer una expresión de esta función de densidad resulta arduo ya que tal y como podemos observar en 4.1 existen diferentes variables aleatorias y con distintas distribuciones que intervienen en el proceso de espera. Por esta razón, el método idóneo para la resolución de problemas de este tipo es la simulación de Monte Carlo.

Una vez visto el planteamiento del problema pasaremos a la simulación la cual se encuentra dividida en 3 fases.

Primera fase

Este primer paso se basa en calcular el número de pacientes que llegan a Urgencias así como el tiempo que transcurre entre la llegada de dos pacientes.

El proceso por el cual llega un paciente a Urgencias sigue una distribución de Poisson. De esta forma, tenemos una variable aleatoria con distribución de Poisson por cada una de las vías iniciales:

- Acceso directo, $\mathcal{X} \sim \mathcal{P}(\lambda_1)$.
- Atención primaria, $\mathcal{Y} \sim \mathcal{P}(\lambda_2)$.

- Ambulancia, $\mathcal{Z} \sim \mathcal{P}(\lambda_3)$.

Los parámetros escogidos para esta distribución han sido elegidos según nuestro criterio. Así, las distintas variables aleatorias correspondientes a la llegada de pacientes tienen como esperanza:

- Acceso directo, $\lambda_1 = 6$.
- Atención primaria, $\lambda_2 = 3$.
- Ambulancia, $\lambda_3 = 0,5$.

En una situación real, en la que tendríamos a nuestro alcance las distintas listas de pacientes y sus horas de llegada, deberíamos estimarlos.

Por otro lado, la variable aleatoria que describe el tiempo que transcurre entre la llegada de dos pacientes sigue una distribución exponencial¹. De nuevo, tendríamos una variable aleatoria de este tipo para cada uno de los principales accesos:

- Tiempo entre llegadas en acceso directo, $\mathcal{T}_{AD} \sim \mathcal{E}(\alpha_1)$.
- Tiempo entre llegadas en atención primaria, $\mathcal{T}_{AP} \sim \mathcal{E}(\alpha_2)$.
- Tiempo entre llegadas en ambulancia, $\mathcal{T}_A \sim \mathcal{E}(\alpha_3)$.

Como valores de los parámetros presentes tenemos:

$$\alpha_1 = 10, \quad \alpha_2 = 20, \quad \alpha_3 = 120.$$

Una vez hemos establecido las distribuciones de las variables aleatorias que intervienen en esta primera parte y los distintos parámetros que las caracterizan, podemos pasar a obtener las diferentes muestras de pacientes y tiempos entre llegadas. Para ello, haremos uso del programa R.

Comenzaremos calculando el número de pacientes que van a llegar a lo largo de un día. A partir del comando `rpois` y los parámetros escogidos obtendremos el número de pacientes que llegan por cada vía. Como entradas a esta función introduciremos 24, por ser el número de horas de un día y la media de la distribución de Poisson en estudio.

Ahora, para calcular el tiempo entre dos llegadas consecutivas realizaremos un cálculo análogo con el comando `rexp`. Como parámetros de esta función deberemos recurrir a la suma de todos los pacientes obtenidos en cada vía y el parámetro que sigue la exponencial en dicho caso.

¹Tal y como habíamos citado anteriormente, estos parámetros se podrían obtener mediante la estimación en aquel caso en el que dispusiéramos de los datos del centro hospitalario correspondiente.

Segunda fase

A partir de los datos obtenidos en la anterior parte vamos a establecer la hora concreta del día a la que cada paciente llega a Urgencias.

Los tiempos que hemos obtenido a través de la exponencial son números decimales por lo que previamente tendremos que hacerles un tratamiento para obtener los minutos y segundos de llegada. Así, si hemos obtenido un valor de 32.883175 minutos nuestro procedimiento consistiría en aproximar este dato a 32,88 y de aquí, calcular con cuántos minutos y segundos se corresponde.

Después de ajustar los datos al formato anterior, procedemos a calcular las distintas horas de llegada. Para esta cuestión, hemos creado la función *calculo_hora* la cual a partir de dos vectores, uno de ellos con los minutos de llegada y el restante con los segundos, devuelve la hora de llegada de cada paciente. El código de esta función es el siguiente:

```
calculo_hora <- function(minutos , segundos){
  hora = c(0,0,0)
  horas=c(replicate(length(minutos),0))
  mins=c(replicate(length(minutos),0))
  segs=c(replicate(length(minutos),0))
  for (i in c(1:length(minutos))){
    hor = hora[1]
    min = hora[2] + minutos[i]
    seg = hora[3] + segundos[i]
    if (seg >= 60){
      min = min + trunc(seg/60)
      seg = seg%%60
    }
    if (min >=60){
      hor = hor + trunc(min/60)
      min = min%%60
    }
    hora = c(hor,min,seg)
    print(paste("Hora llegada paciente", i, ":", hora[1],"horas",hora[2],"
      minutos",hora[3],"segundos."))
    horas[i] = hor
    mins[i] = min
    segs[i] = seg
  }
  return(list(horas,mins,segs))
}
```

En esta segunda fase realizaremos el triaje de cada paciente. Este proceso consiste en la clasificación del paciente en consulta, observación o exploración.

Las distintas probabilidades utilizadas para dicha agrupación son:

- En acceso directo:
 - Probabilidad de ir a consulta: 0.4.
 - Probabilidad de ir a observación: 0.3.
 - Probabilidad de ir a exploración: 0.3.
- En atención primaria:
 - Probabilidad de ir a consulta: 0.5.

4. RESULTADOS

- Probabilidad de ir a observación: 0.3.
- Probabilidad de ir a exploración: 0.2.
- En ambulancia:
 - Probabilidad de ir a consulta: 0.2.
 - Probabilidad de ir a observación: 0.5.
 - Probabilidad de ir a exploración: 0.3.

Una vez hemos establecido las probabilidades anteriores vamos a determinar cuántos pacientes de cada vía es clasificado en consulta, observación o exploración. Para ello hemos creado las siguientes funciones:

```
triaje_AD <- function(valores){
  triaje = c(replicate(length(valores), 0))
  for (i in c(1:length(valores))){
    if (valores[i]<= 0.4) {
      triaje[i] = "Consulta"
    } else if (0.4 < valores[i] && valores[i]>= 0.7){
      triaje[i] = "Observacion"
    } else {
      triaje[i] = "Exploracion"
    }
    print(triaje[i])
  }
}
```

Esta primera función realiza el triaje a aquellos pacientes que provienen de acceso directo.

```
triaje_AP <- function(valores){
  triaje = c(replicate(length(valores), 0))
  for (i in c(1:length(valores))){
    if (valores[i]<= 0.5) {
      triaje[i] = "Consulta"
    } else if (0.5 < valores[i] && valores[i]>= 0.8){
      triaje[i] = "Observacion"
    } else {
      triaje[i] = "Exploracion"
    }
    print(triaje[i])
  }
}
```

Tal y como se puede apreciar en el nombre de la función está destinada a realizar la clasificación de las personas cuya entrada al sistema ha sido a través de la atención primaria.

```
triaje_Am <- function(valores){
  triaje = c(replicate(length(valores), 0))
  for (i in c(1:length(valores))){
    if (valores[i]<= 0.5) {
      triaje[i] = "Observacion"
    } else if (0.5 < valores[i] && valores[i]>= 0.8){
      triaje[i] = "Exploracion"
    } else {
      triaje[i] = "Consulta"
    }
  }
}
```

```

    }
    print(triaje[i])
  }
}

```

Análogamente, este código realiza la clasificación a los pacientes de ambulancia.

El parámetro de entrada de estas funciones será un vector con números aleatorios de una distribución $\mathcal{U}(0,1)$. El número de elementos del vector será el número de pacientes que haya entrada a través de acceso directo, atención primaria y ambulancia, respectivamente.

Al final de esta fase tendremos la hora de llegada de cada paciente, así como su destino.

En la siguiente etapa será preciso conocer cuántos pacientes tenemos de consulta, exploración y observación, así como sus horas de llegada. Para determinar esta cantidad lo que haremos será definir 3 listas (consulta, exploración y observación) que recojan dicha información.

Mostraremos el proceso seguido para el caso de entrar mediante acceso directo ya que para las otras dos entradas el mecanismo es análogo.

Creamos las listas:

```

ExploAD1 = list(replicate(length(tiempoAD1),c(0,0,0)))
ObsAD1 = list(replicate(length(tiempoAD1),c(0,0,0)))
ConsAD1 = list(replicate(length(tiempoAD1),c(0,0,0)))

```

En ellas guardaremos aquellos pacientes de acceso directo que hayan sido derivados a consulta, exploración u observación, respectivamente.

A continuación, recorreremos el vector con el triaje de los pacientes de acceso directo:

```

for (i in c(1:length(tiempoAD1))) {
  if (triajeAD1[i] == "Exploracion"){
    ExploAD1[[i]] = c(horaAD1[i], minutoAD1[i], segundoAD1[i])
    ObsAD1[[i]] = c(0,0,0)
    ConsAD1[[i]] = c(0,0,0)
  } else if (triajeAD1[i] == "Observacion"){
    ObsAD1[[i]] = c(horaAD1[i], minutoAD1[i], segundoAD1[i])
    ExploAD1[[i]] = c(0,0,0)
    ConsAD1[[i]] = c(0,0,0)
  } else if (triajeAD1[i] == "Consulta") {
    ConsAD1[[i]] = c(horaAD1[i], minutoAD1[i], segundoAD1[i])
    ExploAD1[[i]] = c(0,0,0)
    ObsAD1[[i]] = c(0,0,0)
  } else{
    ConsAD1[[i]] = c(0,0,0)
    ExploAD1[[i]] = c(0,0,0)
    ObsAD1[[i]] = c(0,0,0)
  }
}
}

```

A continuación, repetiríamos el proceso con el resto de entradas (atención primaria y ambulancia). De esta forma, obtendríamos 3 listas para consulta, 3 para exploración y otras 3 para observación. Uniríamos estas 3 listas de forma que sólo obtengamos una con todos los pacientes que se dirigirán a consulta, exploración u observación y sus horas de entrada.

4. RESULTADOS

No obstante, debemos modificar estas listas ya que contienen la hora 00:00:00, y además deberíamos ordenar las distintas horas. Por ello, creamos las funciones:

```
eliminar_Ceros <- function(lista){
  listanueva <- list()
  for (i in c(1:length(lista))) {
    if (lista[[i]][1] != 0 || lista[[i]][2] != 0 || lista[[i]][3] != 0){
      listanueva[[length(listanueva) + 1]] <- c(lista[[i]][1], lista[[i]][2], lista[[i]][3])
    }
  }
  return(listanueva)
}
```

```
ordenarHoras <- function(lista){
  listanueva <- list()
  for (k in c(1:length(lista))) {
    hora = c(replicate(length(lista), 1000))
    minuto = c(replicate(length(lista), 1000))
    segundo = c(replicate(length(lista), 1000))
    for (j in c(1:length(lista))) {
      hora[j] = lista[[j]][1]
    }
    h = min(hora)
    for (j in c(1:length(lista))) {
      if (lista[[j]][1] == h){
        minuto[j] = lista[[j]][2]
      }
    }
    m = min(minuto)
    for (j in c(1:length(lista))) {
      if (lista[[j]][1] == h && lista[[j]][2] == m){
        segundo[j] = lista[[j]][3]
      }
    }
    s = min(segundo)
    listanueva[[length(listanueva) + 1]] <- c(h,m,s)
    for (j in c(1:length(lista))) {
      if (lista[[j]][1] == h && lista[[j]][2] == m && lista[[j]][3] == s)
      {
        lista[[j]] = c(1000, 1000, 1000)
      }
    }
  }
  return(listanueva)
}
```

Una vez hemos aplicado estas dos funciones ya obtendríamos 3 listas que contarían con el número de pacientes y el horario de llegada de los mismos.

Tercera fase

El siguiente paso se basa en calcular el tiempo de servicio de cada paciente. Calcularemos el tiempo de estancia en consulta, observación y exploración. Una vez conozcamos estos tiempos podremos operar y conocer cuánto tiempo espera cada paciente a ser atendido.

El tiempo de servicio de un paciente (en minutos) sigue una distribución Normal. Los parámetros que hemos escogido para cada tiempo de servicio son:

- En consulta: $\mu = 30$ y $\sigma = 5$.
- En observación: $\mu = 180$ y $\sigma = 60$.
- En exploración: $\mu = 30$ y $\sigma = 5$.

Recogiendo estos datos y recurriendo a la hora de llegada de cada paciente podemos calcular cuánto tiempo espera cada usuario. Para esta operación hemos programado la siguiente función:

```
esperaN <- function(N, minutosServ, horasServ, segsServ, minEntrada,
  segEntrada, horaEntrada){
  esperas = c(replicate(length(minutosServ),0))
  relojesConsultas = list(replicate(N, c(0,0,0)))
  #Establecer las horas de salida de las N consultas.
  for (k in c(1:N)) {
    relojesConsultas[[k]] = c(horaEntrada[k]+horasServ[k],minEntrada[k] +
      minutosServ[k],segEntrada[k] + segsServ[k])
  }
  #Ajustar los segundos, minutos y horas.
  for (k in c(1:N)) {
    if (relojesConsultas[[k]][3] >= 60){
      relojesConsultas[[k]][2] = relojesConsultas[[k]][2] + trunc(
        relojesConsultas[[k]][3]/60)
      relojesConsultas[[k]][3] = relojesConsultas[[k]][3]%%60
    }
    if (relojesConsultas[[k]][2] >= 60){
      relojesConsultas[[k]][1] = relojesConsultas[[k]][1] + trunc(
        relojesConsultas[[k]][2]/60)
      relojesConsultas[[k]][2] = relojesConsultas[[k]][2]%%60
    }
  }
  print(paste("Horas iniciales."))
  for (k in c(1:N)){
    print(paste("Hora salida paciente",k,":", relojesConsultas[[k]][1], "
      horas",relojesConsultas[[k]][2], "minutos",relojesConsultas[[k]][3], "
      segundos."))
  }

  horComparacion = c(replicate(N,0))
  minComparacion = c(replicate(N,60))
  segComparacion = c(replicate(N,60))

  #Llegada de pacientes:
  for (i in c((N+1):length(minutosServ))) {

    #Ajustar los segundos, minutos y horas.
    for (k in c(1:N)) {
      if (relojesConsultas[[k]][3] >= 60){
        relojesConsultas[[k]][2] = relojesConsultas[[k]][2] + trunc(
          relojesConsultas[[k]][3]/60)
        relojesConsultas[[k]][3] = relojesConsultas[[k]][3]%%60
      }
      if (relojesConsultas[[k]][2] >= 60){
```

```

    relojesConsultas[[k]][1] = relojesConsultas[[k]][1] + trunc(
        relojesConsultas[[k]][2]/60)
    relojesConsultas[[k]][2] = relojesConsultas[[k]][2]%%60
}
}
cat("\n Relojes Consultas\n")
print(relojesConsultas)
cat("\n FIN Relojes Consultas\n")
#Buscar la menor hora.
for (k in c(1:N)) {
    horComparacion[k] = relojesConsultas[[k]][1]
}
a = min(horComparacion)
for (k in c(1:N)) {
    if (relojesConsultas[[k]][1] == a){
        minComparacion[k] = relojesConsultas[[k]][2]
    } else {
        minComparacion[k] = 100
    }
}
b = min(minComparacion)
for (k in c(1:N)) {
    if (relojesConsultas[[k]][1] == a && relojesConsultas[[k]][2] == b){
        segComparacion[k] = relojesConsultas[[k]][3]
    } else {
        segComparacion[k] = 100
    }
}

}
cat("segcomparacion\n")
print(segComparacion)
ce = min(segComparacion)
for (k in c(1:N)) {
    if (relojesConsultas[[k]][1] == a && relojesConsultas[[k]][2] == b &&
        relojesConsultas[[k]][3] == ce){
        d = k
    }
}
}
cat("a=",a,"b=",b,"ce=",ce,"\n")
cat("\nMinimo: ",d,"\n")
print(paste("El paciente", i, "entra en la consulta", d))

#Calcular la espera.
if ((horaEntrada[i] < relojesConsultas[[d]][1]) || (horaEntrada[i] ==
    relojesConsultas[[d]][1] && minEntrada[i] < relojesConsultas[[d]
]][2]) || (horaEntrada[i] == relojesConsultas[[d]][1] && minEntrada[
i] == relojesConsultas[[d]][2] && segEntrada[i] < relojesConsultas[[
d]][3])){
    hora_llegada = c(horaEntrada[i], minEntrada[i],segEntrada[i])
    print(paste("Hora salida paciente",d,":", relojesConsultas[[d]][1],"
        horas", relojesConsultas[[d]][2],"minutos", relojesConsultas[[d]
]][3],"segundos."))
    esperas[i] = calculo_espera(hora_llegada, relojesConsultas[[d]])
    relojesConsultas[[d]][1] = relojesConsultas[[d]][1] + horasServ[i]
    relojesConsultas[[d]][2] = relojesConsultas[[d]][2] + minutosServ[i]
    relojesConsultas[[d]][3] = relojesConsultas[[d]][3] + segsServ[i]
} else if ((horaEntrada[i] > relojesConsultas[[d]][1]) || (horaEntrada[i]

```

```

    ] == relojesConsultas[[d]][1] && minEntrada[i] > relojesConsultas[[d]]
    ][2]) || (horaEntrada[i] == relojesConsultas[[d]][1] && minEntrada[
    i] == relojesConsultas[[d]][2] && segEntrada[i] > relojesConsultas[[
    d]][3]))){
    relojesConsultas[[d]][1] = horaEntrada[i]+ horasServ[i]
    relojesConsultas[[d]][2] = minEntrada[i]+ minutosServ[i]
    relojesConsultas[[d]][3] = segEntrada[i]+ segsServ[i]
    esperas[i] = 0
  }
}
return(esperas)
}

```

Los parámetros de entrada son el número de consultas, los vectores con las horas, los minutos y los segundos de servicio respectivamente así como los vectores con las horas, los minutos y los segundos de llegada.

Esta función devuelve un vector con los segundos que espera cada paciente. Como podemos observar en el código anterior se hace uso de otra función llamada *calculo_espera*, la cual a partir de dos horas es capaz de calcular el tiempo transcurrido entre ambas expresado en segundos. El código de esta última función es:

```

calculo_espera <- function(hora_llegada, reloj){
  #Paso todo a segundos y realizo la operacion.
  h = hora_llegada[1]*60*60
  m = hora_llegada[2]*60
  hora1 = h + m + hora_llegada[3]
  hre = reloj[1]*60*60
  mre = reloj[2]*60
  horar = hre + mre + reloj[3]
  return(horar - hora1)
  #Devuelve en segundos la espera
}

```

Por tanto, el último paso de esta fase será representar de forma gráfica la evolución de la espera a medida que el número de consultas aumenta desde 1 a 10 consultas.

Para resolver esta cuestión haremos uso del comando `boxplot` de R, el cual a partir de un vector con los datos representa un diagrama de cajas. En estos gráficos la línea horizontal que atraviesa a las diferentes cajas representa la mediana. La base inferior y superior del rectángulo coinciden con el primer y tercer cuartil, respectivamente. En último lugar, las líneas discontinuas que surgen de la caja indican el valor mínimo y máximo de los valores utilizados para representarla.

Las gráficas obtenidas en este primer proceso de simulación han sido:

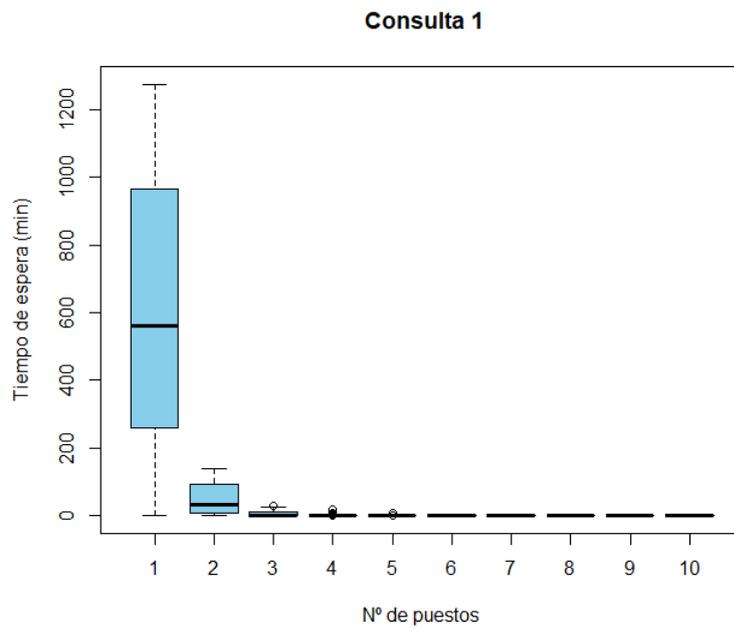


Figura 4.2: Evolución de la espera en consulta.

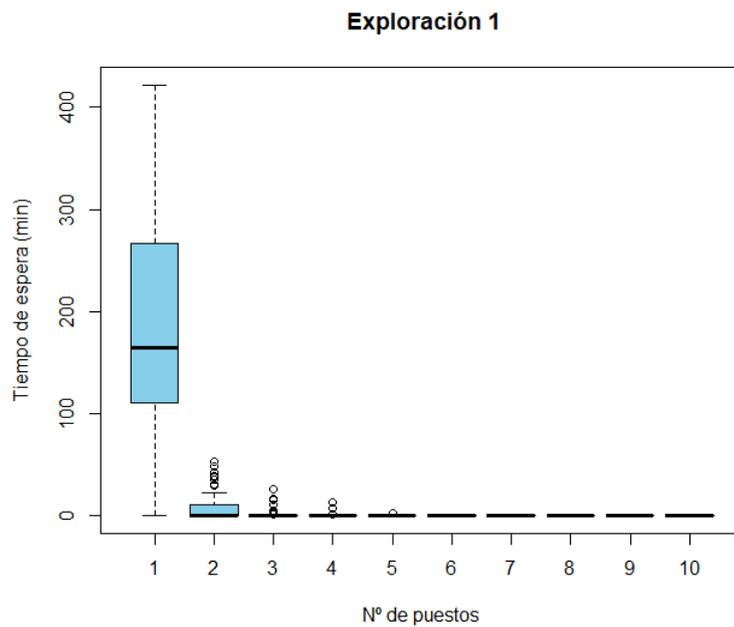


Figura 4.3: Evolución de la espera en exploración.

En estas primeras dos gráficas (Figuras 4.2 y 4.3) podemos observar como la espera se reduce en gran medida cuando se pasa de un sólo puesto de servicio a dos. Además, otra conclusión que se puede extraer de estas representaciones es que la espera no se ve afectada a partir de la tercera consulta.

A diferencia de las anteriores gráficas, el tiempo de espera viene expresado en horas

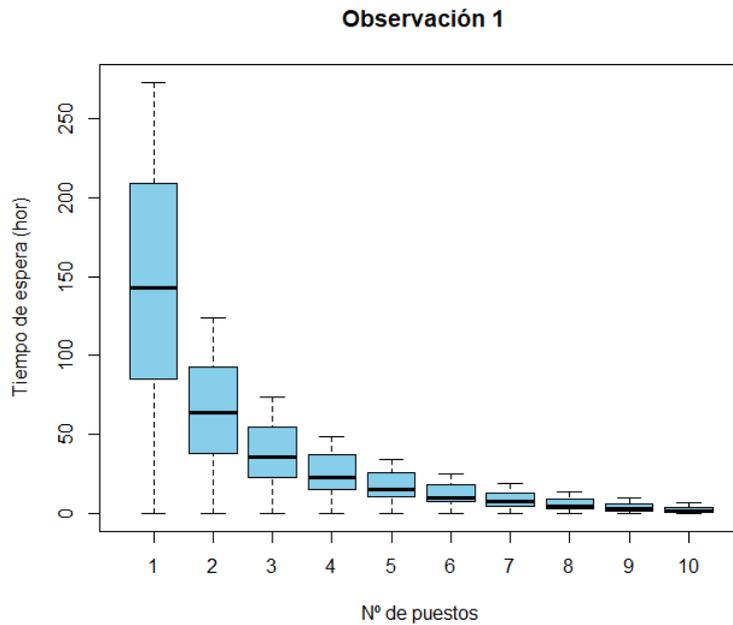


Figura 4.4: Evolución de la espera en observación.

como consecuencia del largo tiempo de servicio (entorno a las 2 horas). Otra discrepancia es la evolución de la espera, la cual se va reduciendo de forma gradual a medida que las consultas se incrementan. No obstante, tal y como podemos observar en la Figura 4.4 a partir de 7 consultas la espera permanece inalterable.

Ahora, vamos a repetir el experimento con otros valores en los parámetros de las variables aleatorias que determinaban el número de pacientes que llegaban por hora por cada vía. Lo repetiremos dos veces, siendo los nuevos parámetros en cada caso:

- Simulación 2: $\lambda_1 = 7$, $\lambda_2 = 4$ y $\lambda_3 = 0,75$.
- Simulación 3: $\lambda_1 = 5$, $\lambda_2 = 3,5$ y $\lambda_3 = 1$.

Así, siguiendo el proceso descrito anteriormente tenemos que la espera en cada simulación se desarrolla de la siguiente forma:

- Simulación 2.

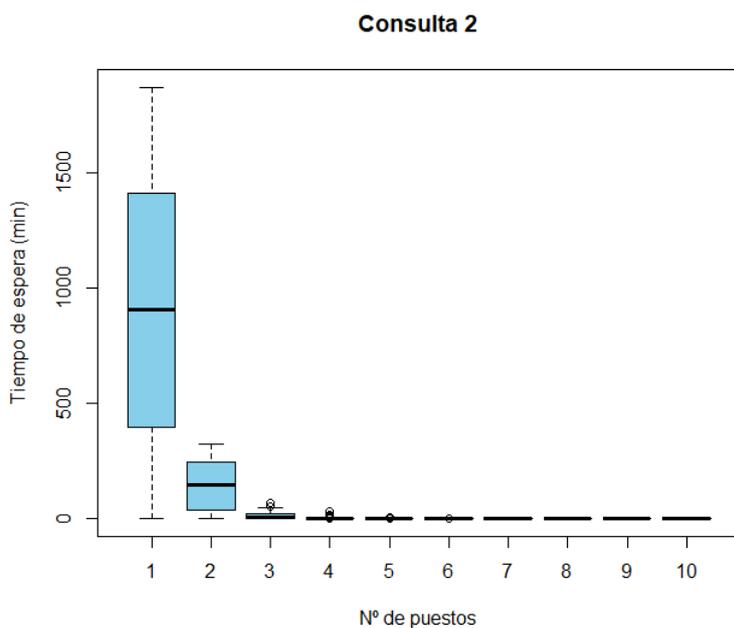


Figura 4.5: Evolución de la espera en consulta.

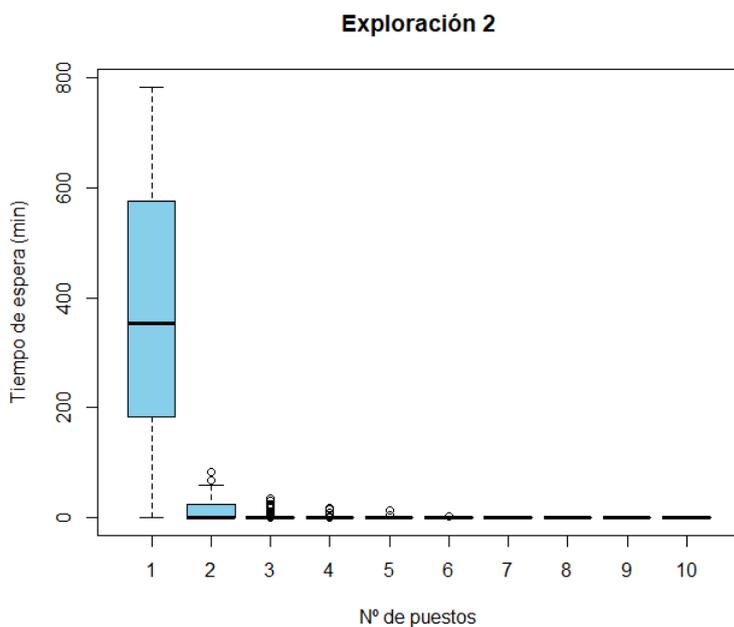


Figura 4.6: Evolución de la espera en exploración.

De forma similar a la simulación anterior podemos ver como la espera de los pacientes (Figuras 4.5 y 4.6) se reduce considerablemente a partir de dos consultas.

En la Figura 4.7 podemos observar una evolución de la espera muy similar al caso anterior donde a partir de 7 consultas la espera apenas se ve alterada.

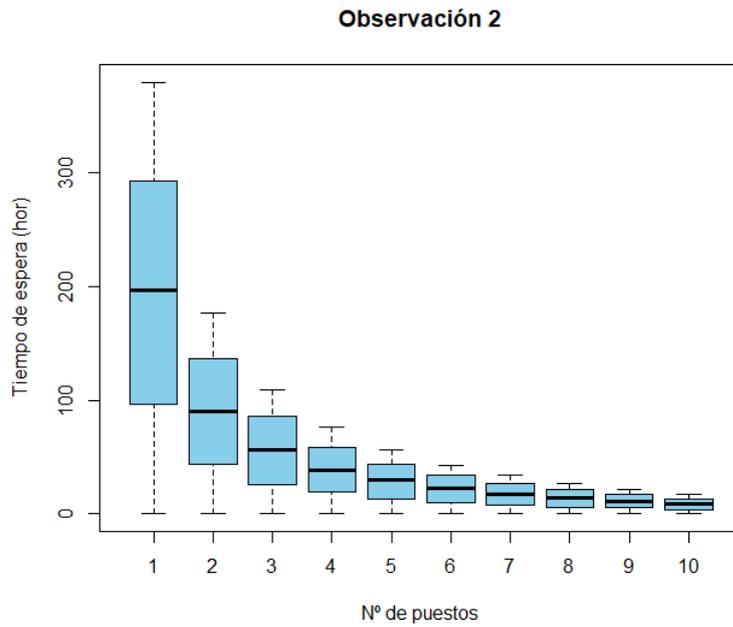


Figura 4.7: Evolución de la espera en observación.

■ Simulación 3.

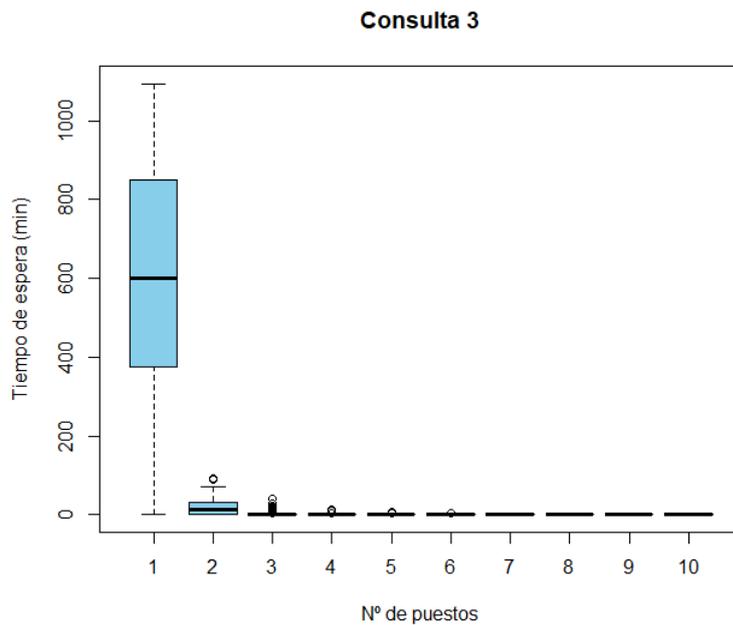


Figura 4.8: Evolución de la espera en consulta.

En este último proceso podemos ver una mayor diferencia de los tiempos de espera cuando se pasa de una consulta a dos. Lo mismo ocurre con la gráfica de observación (Figura 4.10). Podemos apreciar como a partir de 5 consultas la espera es mínima.

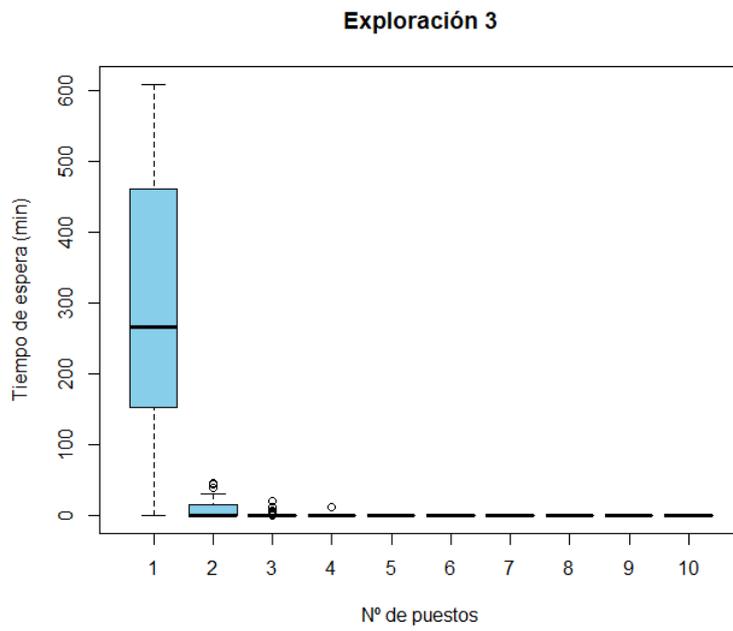


Figura 4.9: Evolución de la espera en exploración.

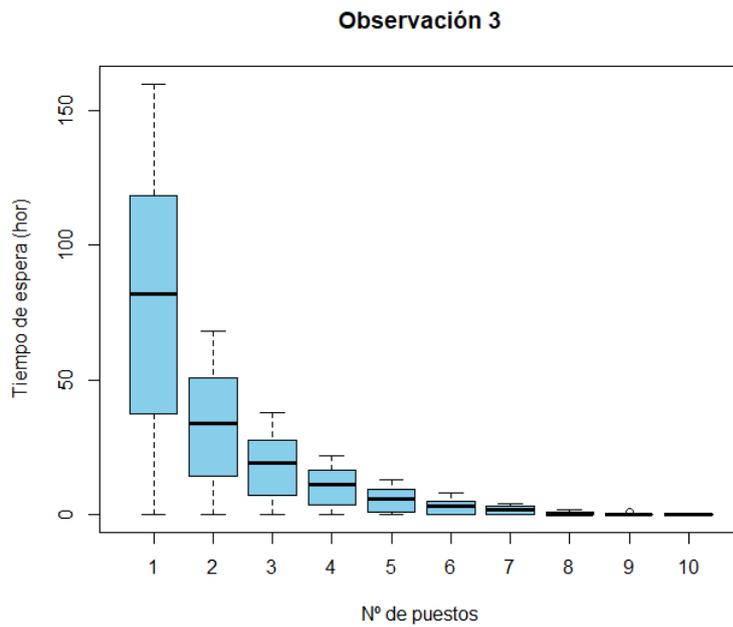


Figura 4.10: Evolución de la espera en observación.

Conclusión

Las conclusiones obtenidas como consecuencia de este trabajo son varias. Por un lado, hemos estudiado, principalmente, los métodos de Monte Carlo y hemos podido apreciar que constituyen una herramienta versátil para obtener soluciones aproximadas a problemas de una complejidad considerable.

Por otro lado, hemos tratado la teoría de colas y hemos visto cómo permite abordar situaciones presentes en la realidad que representan sistemas dinámicos. Así mismo, hemos observado cómo los métodos de Monte Carlo pueden usarse para decidir la asignación de recursos en los dominios analizados.

Particularmente, su aplicación al problema de calcular el número de consultas destinadas a consulta, exploración y observación nos permitiría determinar cuántos puestos se precisarían de manera que los pacientes esperaran cierto tiempo.

Bibliografía

- [1] R. Cao, *Introducción a la Simulación y a la Teoría de Colas*, Netbiblo, 2002.
- [2] T.E. Hull, A.R. Dobell, *Random Number Generators*, SIAM Review, 4.3 (1962) 230-254.
- [3] D. P. Kroese, *Monte Carlo Methods*: <https://people.smp.uq.edu.au/DirkKroese/mccourse.pdf>
- [4] A. M. Law, *Simulation Modeling and Analysis*, McGraw-Hill Education, 2015.
- [5] R. Y. Rubinstein, *Simulation and the Monte Carlo method*, Wiley, 1981.
- [6] A. Salmerón, M. Morales, *Estadística Computacional*: <https://w3.ual.es/~asalmero/papers/libro.pdf>
- [7] Capítulo 8. Introducción al método de simulación de Monte Carlo: <https://uplamcdn.files.wordpress.com/2009/04/libro-cap-08.pdf>