# Índice general

# Capítulo 1

# Números aleatorios

En el año 1945, en la Universidad de Pensilvania, fue inventado el primer ordenador electrónico, conocido como *ENIAC*. Al mismo tiempo, John von Newman, un investigador en Los Alamos, estaba muy interesado en las investigaciones termonucleares que estaban siendo llevadas a cabo. Cuando Newman se enteró de la existencia del ENIAC, propuso un modelo relativamente simple para estudiar la viabilidad de un arma termonuclear.

Los resultados obtenidos del ENIAC fueron revisados por varios científicos en 1946, en Los Alamos. Entre ellos se encontraba Stan Ulam, un investigador que tenía gran interés en el estudio de los procesos aleatorios presentes en los juegos de azar como el poker. Él sabía que las técnicas de muestreo estadísticas estaban en desuso, pero al ver los resultados obtenidos pensó que deberían volver a usarse. Ulam discutió la idea con Newman, creando así en 1947, una aproximación para resolver el problema de la difusión de neutrones en material fisionable usando métodos estadísticos.

Debido a las características del método y al hecho de que un tío de Ulam solía pedir prestado dinero a su familia para ir al casino de Monte Carlo, Nicholas Metropolis sugerió darle ese mismo nombre, surgiendo así el *Método de Monte Carlo*.

Este método necesita números aleatorios, ya que los comportamientos de los neutrones siguen ciertas distribuciones de probabilidad, que pueden ser obtenidas a partir de una distribución uniforme. Por esa razón, en este capítulo vamos a ver qué son los números aleatorios, cómo generarlos y cómo comprobar cómo de aleatorios son los números generados.

## 1.1. Noción de secuencia aleatoria

Un número aleatorio es un número generado por un proceso cuyo resultado es impredecible y que no se puede reproducir posteriormente de forma fiable. A primera vista, esta definición no parece práctica, ya que necesitamos algún mecanismo que nos dé números aleatorios. Existen infinidad de

procesos a partir de los cuales podemos obtener números aleatorios: como los juegos de azar perfectos o la medición de ruidos electrónicos. El principal problema de estos procesos es que no son reproducibles, luego no podemos contrastar los resultados obtenidos.

Por esa razón, es preferible usar una definición matemática, que tenga en cuenta ciertas características de los números aleatorios. Como es esperable, esta definición no es única. En este trabajo se ha escogido la que da Antonio Salmerón [1]:

**Definición 1.1.** *Una secuencia de números aleatorios es una sucesión de variables aleatorias independientes* $\{X_1, \dots X_n\}$ *donde* $X_i \rightsquigarrow \mathcal{U}[0,1)$ *para todo* $i = 1, \dots, n$.

Hay que notar que el dominio de cada $X_i$ no tiene por qué ser el intervalo $[0,1)$, sino que puede ser cualquier intervalo de $\mathbb{R}$. Sin embargo, es conveniente añadir esa restricción ya que facilita algunos cálculos, porque como se verá en una sección posterior, a partir de una distribución uniforme se pueden generar otras muchas distribuciones aleatorias. Otro aspecto importante a notar es que Salmerón no define un *número aleatorio*, sino una *secuencia aleatoria*. Esto es debido a que no tiene mucho sentido verificar la aleatoriedad de un único número.

Por último, es importante tener en cuenta que, en principio, es imposible generar números realmente aleatorios usando un ordenador, ya que los ordenadores son máquinas totalmente determinísticas. Por esa razón, a los números generados a partir de un proceso determinista se les denomina números *pseudoaleatorios*.

**Definición 1.2.** *Los números de una secuencia creada a través de un proceso determinista (una rutina, un programa, etc) reciben el nombre de números pseudoaleatorios.*

## 1.2. Métodos de generación

El objetivo de infinidad de científicos, como John von Newman, era generar números aleatorios [7], para posteriormente obtener otras distribuciones probabilísticas. Como ya se ha comentado, es posible obtener estos números a partir de juegos de azar perfectos, pero el proceso es largo y tedioso, luego el interés de estos métodos para fines estadísticos es escaso.

### 1.2.1. Métodos manuales

Tales procedimientos, al ser tan engorrosos, promovieron la creación de tablas de números aleatorios, como la publicada por L.H.C. Tippett, con 40000 entradas diferentes, obtenida a partir de registros del censo [13]. Posteriormente, en 1955, simulando una ruleta en un dispositivo hardware conectado a un ordenador, la RAND Corporation creó una tabla con un millón

de entradas diferentes [14]. Estos métodos eran denominados *métodos manuales* y sí creaban números realmente aleatorios. Esta tabla fue tan exitosa que incluso se crearon *punched cards* de IBM para que fuera posible usarlas desde un ordenador.

Los principales incovenientes del uso de tales tablas eran que requería una gran cantidad de espacio y las secuencias eran siempre las mismas. Estos problemas fueron solventados gracias a el avance de los ordenadores, la creación de dispositivos como el ERNIE (usado para la lotería inglesa) y el desarrollo de los *métodos digitales*.

### 1.2.2. Métodos digitales

Los métodos digitales son algoritmos numéricos que, a partir de una *semilla*, generan secuencias de números de la forma

$$X_{n+1} = f(X_0, \ldots, X_n)$$

donde $f$ es una función de cualquier tipo. La dependencia funcional de $f$ hace evidente que las secuencias generadas por estos métodos no son realmente aleatorias, sino pseudoaleatorias, ya que responden a una formulación matemática totalmente determínistica. Sin embargo, esta pérdida se suple, en cierta medida, por las propiedades descritas en la definición 1.1. Además, estos métodos o generadores tienen que evitar todos los problemas que hemos descrito anteriormente, es decir, deben caracterizarse por su:

- Reproductividad: toda secuencia de números que generen debe poderse reproducir para que los experimentos que se realicen a partir de ella se pueden contrastar.
- Eficiencia: el método debe ser relativamente rápido de ejecutar y debe consumir poca memoria.

A continuación, se va a describir el primero de estos métodos y luego se va a estudiar el caso más simple de los métodos congruenciales.

#### Middle Square method

El primer método digital desarrollado es denominado *método del centro de los cuadrados*, propuesto por John Von Neumann en 1946 [15]. El método es bastante simple:

1. Fijar un número de dígitos $N \in \mathbb{N}$ y una semilla inicial $n \in \mathbb{N}$.
2. Calcular $n^2$ y tomar las $N/2$ cifras del medio.
3. Repetir (2) con el número obtenido. En caso de que $n^2$ no tenga $N$ dígitos, se añadiran tantos ceros a la izquierda como sea necesario.

**Ejemplo 1.1.** *Tomando $N = 4$ y $n = 24$, se obtiene la siguiente tabla:*

| $n$ | $n^2$ | $X_i$ |
|---|---|---|
| *24* | 0\|57\|6 | *0.52* |
| *57* | 3\|24\|9 | *0.24* |
| *24* | 0\|57\|6 | *0.52* |
| *57* | 3\|24\|9 | *0.24* |
| ... | ... | ... |

Como se aprecia en el ejemplo, el método va a repetir los números 24 y 57 sucesivamente. Este hecho da lugar a la siguiente definición:

**Definición 1.3.** *A la longitud de la secuencia máxima que se pueda obtener a partir de un generador de números pseudoaleatorios se le denomina longitud de periodo o simplemente periodo.*

En el ejemplo 1.1, el periodo es igual a 2. Es evidente que un periodo tan bajo no tiene ningún interés, luego en los generadores siempre va a interesar obtener el máximo periodo posible. Este hecho provoca la necesidad de estudiar qué factores afectan al periodo. En principio, lo más lógico es que dependa de la elección del número inicial, $X_0$, al que llamaremos *semilla* y de los propios parámetros del método. Si en el ejemplo anterior se escogiera como semilla el número 1234, se obtendría un periodo igual a 55. Sin embargo, a partir de ese término, lo único que obtendríamos es una sucesión de ceros, con lo cual, hemos perdido la uniformidad, incumpliendo la definición 1.1.

Debido a estos problemas, de los que el propio Newman era consciente [7], generadores más eficaces fueron diseñados, teniendo en cuenta las tres características anteriores: reproductividad, eficiencia y periodo máximo.

### Generadores congruenciales

A se describe una familia de generadores ampliamente usados debido a que tienen una teoría muy bien desarrollada. Además son muy fáciles de implementar y requieren de muy poca memoria para funcionar. En concreto, se va a estudiar el generador congruencial simple, que como su nombre indica, es el caso más simple de esta familia, pero tiene la ventaja de que es posible determinar de forma exacta su periodo máximo. Al final se estudiará otro caso particular de esta familia: el registro de retroalimentación lineal con desplazamiento.

El generador congruencial simple se basa en calcular congruencias módulo un número natural $m$, al resultado de evaluar una determinada función $f$:

$$X_{n+1} = f(X_0, \ldots, X_n) \ mod \ m \tag{1.1}$$

Dentro de este tipo de métodos, se va a estudiar el caso en el que $f$ es una función lineal tal que:

$$X_{n+1} = (aX_n + c) \; mod \; m \tag{1.2}$$

donde

- $a$, es el multiplicador $(0 \leq a < m)$
- $m$, el módulo $(m > 0)$
- $c$, el incremento $(0 < c < m)$
- $X_0$, la semilla $(0 < m)$

A este método se le conoce como Generador Congruencial Lineal (GCL) y fue propuesto por D.H. Lehmer en 1949 [5]. Es fácil deducir que el periodo siempre será igual o menor que $m$ (debido al módulo). Luego cuanto más grande sea $m$, más largo podrá ser el periodo. Esto conduce a la pregunta de cómo influye el resto de los parámetros en el periodo. La respuesta la tiene el siguiente teorema [3], que da condiciones necesarias y suficientes para obtener un GCL de longitud máxima $m$.

**Teorema 1.1** (Hull-Dobell)**.** *La secuencia definida por la equación* (1.2)*, tiene periodo máximo si y solo si:*

*(i)* $c$ *es primo relativo con* $m$*.*

*(ii)* $a \equiv 1 \; (mod \; p)$*, si* $p$ *es un factor primo de* $m$*.*

*(iii)* $a \equiv 1 \; (mod \; 4)$*, si* $4$ *es un factor de* $m$*.*

**Nota 1.1.** *Si* $c = 0$*, el teorema anterior no se podría aplicar. En ese caso, tendríamos otro tipo de generador llamado Generador Congruencial Multiplicativo (GCM), que precisa otro estudio aparte.*

*Demostración.* El caso $a = 1$ es sencillo, ya que si $mcd(c, m) = 1$, el periodo es $m$ por las propiedades del módulo. Por lo tanto, solo es necesario demostrar el caso $a \neq 1$. Partiendo de (1.2), se puede sustituir recursivamente la expresión de $X_n$, obteniéndose:

$$x_n \equiv a^n x_0 + \frac{(a^n - 1)c}{a - 1} \; (mod \; m) \tag{1.3}$$

Se desea encontrar la longitud de ese generador, es decir, el menor número $n \in \mathbb{N}$ tal que $X_n = X_0$. Operando en (1.3) se obtiene:

$$\frac{(a^n - 1)(x_0(a - 1) + c)}{a - 1} \equiv 0 \; (mod \; m)$$

Para reducir esa expresión, se usa que $x_0(a-1)+c$ es primo relativo con $m$. Eso se razona por reducción al absurdo usando las condiciones del teorema. Sea $\lambda = x_0(a-1)+c$, suponiendo $p$ un número primo tal que $p|m$ y $p|\lambda$. Por la condición $ii$, se tiene que $a \equiv 1\ (mod\ p)$, es decir:

$$a = 1 + kp,\ k \in \mathbb{N} \Rightarrow x_0(1+kp-1)+c\ (\ mod\ p) = c \Rightarrow \lambda \equiv c\ (mod\ p)$$

Pero se ha asumido que $p|\lambda$, luego $\lambda \equiv 0\ (mod\ p)$, pero $p \nmid c$ (por $i$), así que es una contradicción. Luego esa parte de la ecuación puede ser obviada, teniendo que resolver en su lugar:

$$\frac{a^n - 1}{a-1} \equiv 0\ (mod\ m) \tag{1.4}$$

Lo que se quiere demostrar es que si $a$ satisface las condiciones del teorema, entonces $n$ es igual a $m$. Para ello, primeramente se va a ver que si $m$ es una potencia de un primo mayor que 2, es decir, $m = p^\alpha$, donde $\alpha \in \mathbb{N}$ y $\alpha \geq 2$ (si $\alpha = 1$ es el caso del principio).

Como $a$ satisface la condición $(ii)$, $a$ se expreserá como:

$$a = 1 + kp^\beta \tag{1.5}$$

donde $mcd(k,p) = 1$, $k \neq 0$ (ya que$a \neq 1$) y $\beta \in \mathbb{N}$. Para comprobar que $n = p^\alpha = m$ satisface (1.4), se sustituye el valor de $n$ y $a$, obteniendo:

$$\begin{aligned}\frac{a^n-1}{a-1} = \frac{(1+kp^\beta)^{p^\alpha}-1}{kp^\beta} = \frac{1+\sum_{j=1}^{p^\alpha}\binom{p^\alpha}{j}(kp^\beta)^j - 1}{kp^\beta} = \\ = \frac{p^\alpha kp^\beta + \frac{p^\alpha(p^\alpha-1)}{2!}(kp^\beta)^2 + \cdots + (kp^\beta)^{p\alpha}}{kp^\beta} = \\ = p^\alpha + \frac{p^\alpha(p^\alpha-1)}{2!}kp^\beta + \cdots + (kp^\beta)^{p^\alpha-1}\end{aligned} \tag{1.6}$$

Ahora solo hay que ver que esa expresión es divisible por $p^\alpha$, o lo que es lo mismo, que cada término es divisble por $p^\alpha$. Para ello, se puede reescribir el término j-ésimo como sigue:

$$\frac{p^\alpha}{j}\binom{p^\alpha}{j-1}k^{j-1}p^{(j-1)\beta},\quad (j>1)$$

Se tiene que $\binom{p^\alpha}{j-1}$ y $k^{j-1}p^{(j-1)\beta}$ son enteros, luego ninguno de ellos *necesitará* parte de $p^\alpha$. Por tanto, el único elemento que puede *tomar* algo de $p^\alpha$ es $j$. Sin embargo, como $j$ toma valores en $\{2,\ldots,p^\alpha\}$, el número de veces que el factor $p$ puede aparecer en $j$ es menor que (usando la fórmula de Legendre):

$$v_p(j!) = \sum_{i=1}^{\infty}\left\lfloor\frac{n}{p^i}\right\rfloor \leq \frac{j}{p} + \frac{j}{p^2} + \frac{j}{p^3} + \cdots = \frac{j}{p-1} \tag{1.7}$$

y por lo tanto, es menor o igual que $j-1$. Pero el factor $p$ aparece al menos $j-1$ veces in $p^{(j-1)\beta}$, ya que $\beta \geq 1$. Por consiguiente, el factor $p^\alpha$ no es necesario para que $j$ sea dividido por un elemento del numerador. Como todo elemento de (1.6) es divisible por $p^\alpha$, la ecuación (1.4) se cumple para $n = p^\alpha$.

Todavía queda demostrar que, efectivamente, $n = p^\alpha$ es el menor valor que satisface (1.4). Hay que ver que ningún valor menor a $p^\alpha$ satisface (1.4). Como (1.4) es equivalente a $a^n \equiv 1 \ (mod \ m)$, se tiene que $a$ pertenece al exponente de $n \, (mod \, m)$. Por tanto, se puede usar el teorema descrito en [8], que dice que si hay otro número $N$ que cumple (1.4), entonces $n$ divide a $N$. Como $n = p^\alpha$, viendo que no se cumple para $p^{\alpha-1}$ es suficiente para asegurar que $n$ es el mínimo.

Repitiendo los mismos pasos que en (1.6), pero con $n = p^{\alpha-1}$, se obtiene la siguiente expresión para el término j-ésimo:

$$\frac{a^n - 1}{a-1} = p^{\alpha-1} + \frac{p^{\alpha-1}(p^{\alpha-1}-1)}{2!}kp^\beta + \cdots + (kp^\beta)^{p^{\alpha-1}-1} \tag{1.8}$$

donde el coeficiente j-ésimo es igual a

$$\frac{p^{\alpha-1}}{j}\binom{p^{\alpha-1}}{j-1}k^{j-1}p^{(j-1)\beta}, \quad (j > 1)$$

Al contrario que antes, ahora se tiene que ver que ese número no es divisible por $p^\alpha$. Evidentemente, el primer término, $p^{\alpha-1}$, no es divisible por $p^\alpha$, por consiguiente, con demostrar que los otros términos sí dividen a $p^\alpha$ es suficiente. El argumento es análogo al anterior, pero esta vez falta otro término $p$. Ese término se obtiene de $p^{(j-1)\beta}$, ya que como $p$ es impar, (1.7) es menor o igual que $j-2$.

Con esto queda concluida la prueba para $m = p^\alpha$ si $p$ es un primo impar. Para $p = 2$, simplemente hay que considerar la fórmula alternativa de Legendre:

$$v_p(n!) = \frac{n - s_p(n)}{p-1} \Rightarrow v_2(j!) = j - s_2(j) \leq j - 1$$

Ahora faltaría el caso en el que $m$ es producto de potencias de primos, es decir, cualquier número natural, pero tomando

$$m = p_1^{\alpha_1} \cdots p_s^{\alpha_s}, \quad a = 1 + kp_1^{\alpha_1} \cdots p_s^{\alpha_s}$$

donde $\{p_1, \ldots, p_s\}$ son primos, $\{\alpha_1, \ldots, \alpha_s\}$ son enteros positivos y $k \neq 0$ con $mcd(k, m) = 1$, la demostración es prácticamente igual al caso que se ha desarrollado. □

Este teorema tiene gran importancia ya que asegura que si se escogen los parámetros $a$ y $c$ cumpliendo ciertas condiciones, se puede obtener un

generador congruencial lineal de longitud arbitraria $m$. Esa característica combinada con lo simple que es de calcular $X_{n+1}$, hace a este tipo de G.C.L una buena elección de generador de números pseudoaleaorios. En [2], puede consultarse una lista de tuplas $(a, c, m)$ escogidas por diferentes implementaciones de este generador.

**Nota 1.2.** *Es importante notar, que aunque el teorema es válido para cualquier producto de primos, casi siempre se toma m como una potencia de 2, normalmente* $2^{32}$ *o* $2^{64}$*. Esto es debido a razones de eficiencia por la forma en la que se almacenan los datos en la memoria de un ordenador.*

Por último, se va a ver una generalizacion del generador congruencial multiplicativo.

### Linear-Feedback Shift Register Generators

**Modificar acorde a Kalos, p.182**

Un registro de retroalimentación lineal con desplazamiento, o LFSR, es un registro de desplazamiento (circular) cuyo bit de entrada es el resultado de evaluar una función lineal usando su estado previo. Este tipo de generadores fue propuesto por Tausworthe en 1965. El bit resultante puede ser expresado mediante la siguiente recurrencia:

$$b_i \equiv (a_p b_{i-p} + a_{p-1} b_{i-p+1} + \ldots + a_1 b_{i-1}) \ (mod\ 2) \tag{1.9}$$

Al igual que en el generador anterior, el interés se encuentra en saber cuál es el periodo máximo que se puede obtener. Para averiguarlo, Gentle identifica esa recurrencia con el siguiente polinomio (que debe ser irreducible)

$$f(z) = z^p - (a_1 z^{p-1} + \ldots + a_{p-1} z + a)$$

sobre el cuerpo de Galois $\mathcal{G}(2)$, definido sobre los enteros $\mathbb{Z}_2$. Aplicando la teoría de Galois, se puede ver que el periodo del generador es $2^p - 1$ (ver [2]). Por eficiencia en los cálculos, la mayoría de los coeficientes suelen ser 0. Los polinomios más usados suelen ser trinomios irreducibles de la forma

$$R(p, r) = x^p + x^r + 1, \quad p, r \in \mathbb{N}, \ \ p > r$$

Ejemplos de tales polinomios pueden ser encontrados en listas como la publicada por Zierler, [16]. Ahora podemos volver a aplicar la identificación anterior entre polinomio y recurrencia, para obtener la expresión asociada al polinomio $R(p, r)$, obteniendo así la expresión de un LFSR:

$$b_i \equiv b_{i-p} + b_{i-r} \ (mod\ 2) \tag{1.10}$$

Una vez se tiene la expresión, solo falta escoger una semilla de $p \in \mathbb{N}$ bits. Para ello, se sigue el siguiente procedimiento:

1. Se escogen $p \in \mathbb{N}$ bits como semilla inicial.
2. Se evalua la expresión 1.10 $l$ veces, con $l \in \mathbb{N}$ y$l \leq p$.
3. Se considera la tupla de $l$ bits como un número en base 2
4. Se vuelve a evaluar la expresión, obteniendose así un nuevo bit.
5. Se desplazan los bits actuales un lugar (hacia la izquierda o derecha) sustituyendo el hueco dejado por el bit anterior y se repite el proceso.

La secuencia de números obtenida se denomina *l-wise decimation*. Si $l$ es primo relativo con $2^p - 1$, se comprueba en [2] que entonces el periodo de las $l-$tuplas es $2^p - 1$.

**Ejemplo 1.2.** *Supongamos que tenemos un LFSR con la siguiente expresión:*

$$b_i \equiv b_{i-5} + b_{i-3} \ (mod\ 2) \tag{1.11}$$

*Para estudiar su periodo, se calcula su polinomio asociado, que en este caso es* $x^5 + x^3 + 1$. *Ese polinomio aparece en la lista de Zierler, por lo que es irreducible, luego el periodo de este generador es* $2^5 - 1 = 31$.

*Para generar esos 31 números, primero es necesario escoger una semilla inicial de 5 bits. Esta elección es arbitraria, así que se puede tomar, por ejemplo, 10011. Fácilmente se puede generar el resto de la secuencia, donde cada bit es el coeficiente de uno de los términos de tales polinomios. Una vez fijada la semilla, hay que escoger un número l que sea primo relativo con 31, como* $l = 17$. *Posteriormente, el método es evaluado 17 veces, obteniendo:*

$$10011010010000010101110$$

*Una vez se tiene la secuencia, hay que escoger los 17 primeros bits e interpretárlos en decimal, obteniendo 20365. Para el siguiente número, se aplica de nuevo la expresión 1.11 y se inserta el resultado por la izquierda, desplazando el resto de bits. Repitiendo el proceso 31 veces, se obtiene:*

*20365, 43082, 15927, 41258, 63709, 33961, 58229, 4775, 36308, 19103, 14160, 10876, 56642, 43505, 29961, 42950, 54309, 40731, 20629, 31854, 16980, 61882, 2387, 50922, 9551, 7080, 38206, 28321, 21752, 47748, 21475, 59922*

## 1.3. Calidad de los generadores

En la sección anterior se han estudiado algunos tipos de generadores. En cada ejemplo se ha hecho un razonamiento sobre cuánto vale el periodo. Ahora lo que se quiere es desarrollar herramientas que permitan comprobar que los números generados cumplen las característicasde la definición 1.1. Para ello, hay gran infinidad de tests contrastes de hipótesis que ayudan a comprobarlo:

### 1.3.1. Test $\chi^2$

El test *chi-cuadrado*, denotado por $\chi^2$, es un test de contraste de hipótesis que puede ser usado para contrastar, a partir de una muestra aleatoria simple de una variable $X$, si $X$ sigue una distribución determinada. En este caso, como se quiere comprobar si los números generados son efectivamente aleatorios, lo que hay que comprobar es si los datos siguen una distribución uniforme. Por lo tanto, las hipótesis del test son:

- $H_0$: los datos proceden de una distribución uniforme.
- $H_1$: los datos no proceden de una distribución uniforme.

Como el test $\chi^2$ es cualitativo, es necesario clasificar los números generados en $d \in \mathbb{N}$ subconjuntos, denominados *clases*. A la cantidad de números dentro de cada clase, se le llama *frecuencia observada de la clase* $i$, y se denota por $N_i$. Calcular estas frecuencias es fácil, ya que podemos realizar una homotecia del intervalo $[0, 1]$ al intervalo $[0, d]$, usando la función parte entera para obtener un índice válido: $C_i = \lfloor dx_i \rfloor$. Una vez se tiene la frecuencia observada, falta calcular la *frecuencia esperada*, $E_i$, que se puede calcular como $E_i = np_i$. Como el objetivo es constrastar que los datos siguen una distribución uniforme, se tiene que $p_i = \frac{1}{d}$, luego $E_i = \frac{n}{d}$.

Una vez obtenidos todos los datos necesarios, se puede usar el estadístico del test $\chi^2$, que sigue una distribución $\chi^2(d-1)$:

$$\chi^2 = \sum_{i=1}^{d} \frac{(N_i - E_i)^2}{E_i} \longrightarrow \chi^2(d-1)$$

Como las frecuencias esperadas son constantes, la expresión anterior puede simplificarse en términos de operaciones a realizar:

$$\chi^2 = \sum_{i=1}^{d} \frac{(N_i - E_i)^2}{E_i} = \frac{d}{n} \sum_{i=1}^{d} (N_i - E_i)^2 = \\ = \frac{d}{n} \left( \sum_{i=1}^{d} N_i^2 + \sum_{i=1}^{d} \frac{n^2}{d^2} - 2\frac{n}{d} \sum_{i=1}^{d} N_i \right) = \frac{d}{n} \sum_{i=1}^{d} N_i^2 - n \quad (1.12)$$

Para poder aplicar este test, es obligratorio que las frecuencias esperadas en cada clase sea mayor que 5, es decir, $E_i > 5$, luego se elige el número de clases $d$ como el mínimo número de clases necesario para cumplir esa condición.

Por último, es necesario fijar un nivel de significación $\alpha$ (normalmente $\alpha = 0.05$), y comprobar si el valor resultanto del estadístico pertenece al siguiente conjunto:

$$\{\chi^2 \leq \chi^2_{d-1,\frac{\alpha}{2}}\} \cup \{\chi^2 \geq \chi^2_{d-1,1-\frac{\alpha}{2}}\}$$

### 1.3.2. Test de Kolmogorov-Smirnov

Este test al igual que el anterior es un contraste de bondad de ajuste, en el que se contrasta si $X$ sigue una distribución determinada. A diferencia del test $\chi^2$, este es cuantitativo y se aplica a variables aleatorias continuas. Se basa en el teorema de Glivenko-Cantelli [12].

Sea $(X_1, \ldots, X_n)$ una muestra aleatoria simple una variable aleatoria $X$ que sigue una distribución $F$ desconocida. Las hipótesis del contraste a resolver son:

- $H_0$: la dsitribución desconocida $F$ es una distribución uniforme.
- $H_1$: la dsitribución desconocida $F$ no es una distribución uniforme.

Para resolverlo es necesario usar el estadístico de Kolmogorov-Smirnov:

$$D(X_1, \ldots, X_n) = \sup_{x \in \mathbb{R}} \left| F^*_{X_1,\ldots,X_n}(x) - U(x) \right|$$

donde $F^*$ es la función de distribución muestral y $U$ es la función de distribución de una variable aleatoria uniforme. El test se rechazará si se da que $D(X_1, \ldots, X_n) \geq d_\alpha$, donde $d_\alpha$ es el valor que cumple:

$$P_{H_0}(D(X_1, \ldots, X_n) \geq d_\alpha) = \alpha$$

siendo $\alpha$ el nivel de significación.

### 1.3.3. Test de rachas

Esta prueba se realiza considerando los dígitos de los números generados. La prueba consiste en contar el número de dígitos (longitud) que aparecen entre ocurrencias sucesivas de un mismo dígito (huecos). Por ejemplo, el número 834938 presenta un hueco de longitud cuatro entre los dos ochos.

La probabilidad de que aparezca cada uno de los tamaños de longitud $i$ se obtiene con la siguiente expresión:

$$p_i = 0.1(0.9)^n \text{ para } i = 0, 1, 2, \ldots$$

Sin embargo, como teóricamente el valor del tamaño del hueco puede ser infinito, es conveniente agrupar las probabilidades para valores de $i$ mayores o iguales a un determinado natural $k$, fijado arbitrariamente por el realizador del test.

Posteriormente, es necesario calcular las fecuencias observadas $N_i$, con $i = 0, \ldots, k$ y las frecuencias esperadas $E_i = np_i$, con $i = 0, \ldots, k$. Una vez obtenidas, se puede calcular el valor del estadístico $\chi^2$:

$$\chi^2 = \frac{d}{n} \sum_{i=1}^{d} E_i^2 - n \longrightarrow \chi^2(k)$$

Y construir un contraste para las hipótesis:

- $H_0$: los datos proceden de una distribución uniforme y son independientes.
- $H_1$: los datos no proceden de una distribución uniforme y no son independientes.

# Bibliografía


[1] María Morales Giraldo Antonio Salmerón Cerdán. *Estadística Computacional*. 2001.

[2] James E Gentle. *Random number generation and Monte Carlo methods*. Springer Science & Business Media, 2006.

[3] Thomas E Hull y Alan R Dobell. "Random number generators". En: *SIAM review* 4.3 (1962), págs. 230-254.

[4] Malvin H Kalos y Paula A Whitlock. *Monte carlo methods*. John Wiley & Sons, 2009.

[5] Derrick H Lehmer. "Mathematical methods in large-scale computing units". En: *Annu. Comput. Lab. Harvard Univ.* 26 (1951), págs. 141-146.

[6] George Marsaglia. "DIEHARD: a battery of tests of randomness". En: *http://stat. fsu. edu/geo* (1996).

[7] N Metropolis. *Beginning of the Monte Carlo Method. 1987*.

[8] Oystein Ore. *Number theory and its history*. Courier Corporation, 1988, pág. 280.

[9] Peter Shirley. *Ray Tracing in One Weekend*. 2020. URL: `https://raytracing.github.io/books/RayTracingInOneWeekend.html`.

[10] Peter Shirley. *Ray Tracing: The Next Week*. 2020. URL: `https://raytracing.github.io/books/RayTracingTheNextWeek.html`.

[11] Peter Shirley. *Ray Tracing: The Rest of Your Life*. 2020. URL: `https://raytracing.github.io/books/RayTracingTheRestOfYourLife.html`.

[12] Michel Talagrand. "The Glivenko-Cantelli problem". En: *The Annals of Probability* (1987), págs. 837-870.

[13] L.H.C Tippett. *Random sampling numbers*. 1927.

[14] John W Tukey. *A Million Random Digits with 100,000 Normal Deviates*. 1955.

[15] John Von Neumann. "13. various techniques used in connection with random digits". En: *Appl. Math Ser* 12.36-38 (1951), pág. 5.

[16] Neal Zierler y John Brillhart. “On primitive trinomials (mod 2), II”. En: *Information and Control* 14.6 (1969), págs. 566-569.