

Capítulo 2: Números aleatorios

1.	Métodos de números pseudoaleatorios	4
1.1.	Generadores Congruenciales Lineales Multiplicativos	10
1.2.	Otros generadores congruenciales	10
2.	Calidad de los Generadores de números aleatorios.	12
2.1.	Test de bondad de ajuste	13
2.1.1.	Contrates de la χ^2 para la bondad de ajuste.....	13
2.1.2.	Contraste de bondad de ajuste de Kolmogorov-Smirnov	17
2.2.	Contrastes de aleatoriedad	20
2.2.1.	Contraste de las rachas arriba y abajo.....	20
2.2.1.	Contraste de media.....	20
2.2.2.	Contraste del Póker	21
2.2.3.	Contraste Gap (de salto)	24
2.3.	Contrastes espectrales	25
2.3.1.	Test de autocorrelación	25
3.	Generador de números aleatorios en software.....	26
4.	Aplicaciones de los números aleatorios	29
4.1.	Resolución de problemas de probabilidades	29
4.2.	Calculo numérico de integrales por el método de Monte Carlo.....	31
4.2.1.	Método de acierto-fallo	31

4.2.2. Método de la media muestral	36
4.3. Comparación de los dos métodos	37
4.4. Optimización de funciones	39

“Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.” Von Neuman

La generación de secuencias de números que siguen una distribución dada se conoce como la generación de variables aleatorias. Dentro de las funciones de distribución tiene especial interés la distribución uniforme porque a partir de ella se puede generar las demás. El estudio de cómo obtener una secuencia de números que provengan de una función de distribución $\mathcal{U}(0, 1)$ y la calidad de la misma es el objeto de este segundo capítulo.

Definición de número aleatorio.

Una secuencia infinita de números se dice que es aleatoria si cualquier secuencia finita, seleccionada previamente a su diseño, es igualmente probable que esté incluida en aquella.

Los procedimientos para obtener números aleatorios se clasifican en tres:

- Generadores de números aleatorios
- Tablas de números aleatorios.
- Métodos de números pseudoaleatorios.

Los primeros que aparecieron fueron los *generadores de números aleatorios* y las tablas de números aleatorios. Los generadores se aprovechaban de las características físicas y probabilísticas de ciertos artilugios como: equipos de bolas, urnas, ruletas, guías de teléfono, ruido eléctrico, etc. Su principal limitación es que proporcionan series de números aleatorios *no reproducibles*.

Las tablas de números aleatorios fue el primer intento de disponer de series de números aleatorios de una forma práctica que permitiera la reproducibilidad de las series utilizadas en un experimento de simulación. Sus mayores limitaciones son la lentitud de uso y la necesidad de almacenamiento. Incluso con la llegada de los ordenadores los problemas de almacenamiento y velocidad de acceso hizo necesario buscar otros sistemas de obtención de números aleatorios.

1. Métodos de números pseudoaleatorios

La necesidad de disponer de números aleatorios en gran *cantidad*, con *rapidez*, *reproducibles* y que consuman *poca memoria* hizo que se desarrollaran unos algoritmos matemáticos que proporcionan unas series numéricas tales que sus características estadísticas son semejantes a las de los verdaderos números aleatorios: los números *pseudoaleatorios*. Los números pseudoaleatorios son secuencias de números que se comportan como una secuencia proveniente de una $\mathcal{U}(0,1)$ y “verifican” las siguientes propiedades:

- I. Tiene una distribución uniforme
- II. Son estadísticamente independiente
- III. Son reproducibles
- IV. Tienen un ciclo no repetitivo tan largo como se quiera
- V. Ocupan poca memoria en el ordenador

Actualmente la condición quinta ha perdido interés, sin embargo, se ha sustituido por la condición que la generación de los números aleatorios sea rápida.

La forma de un generador de números pseudoaleatorios es:

$$x_{j+1} = f(x_j, x_{j-1}, \dots, x_1)$$

el valor de x_{j+1} depende de los valores anteriores generados. De entre las posibles relaciones definidas por f se utiliza la que define a los generadores congruenciales lineales

$$x_{j+1} = (ax_j + b) \bmod m$$

Los generadores congruenciales lineales son uno de los más antiguos y conocidos de entre los métodos que se emplean para generar números aleatorios, fundamentalmente por la facilidad de implementación, su rapidez y la poca necesidad de memoria que requieren para implementarse.

Generador congruencial

Dado un $x_0 \in \{0, 1, \dots, m - 1\}$ inicial (*semilla*) y dos números a y b ($a, b \gg m$) se define una secuencia de números $x_n \in \{0, 1, \dots, m - 1\}$, $n = 0, 1, \dots$ por medio de la relación de recurrencia,

$$x_{j+1} = (ax_j + b) \bmod m$$

Una vez obtenida la secuencia $\{x_n\}_{n=0}^{m-1} \in [0,1)$, basta con calcular

$$u_n = \frac{x_n}{m}, \quad n = 0, 1, \dots, m - 1$$

para obtener una secuencia de números en $[0,1)$.

Bajo ciertas condiciones de a y b se verifica que la secuencia u_0, u_1, \dots se comporta como una m.a.s. proveniente de una $\mathcal{U}(0,1)$ (desde un punto de vista estadístico).

En la práctica se descarta tanto el valor 0 como el 1 (aunque este último por construcción no pueda darse (salvo redondeos)).

Los métodos congruenciales aparecen en un intento de automatizar y obtener secuencias más largas de números pseudoaleatorios. Por ejemplo, si se utilizan fórmulas recursivas como

$$x_{n+1} = \text{parte fraccional } (\pi + x_n)^5 \text{ para } n \geq 0$$

donde $x_0 \in (0,1)$ y recibe el nombre de **semilla**. La secuencia así obtenida es reproducible, ya que queda determinada por la semilla y sus propiedades estadísticas pueden ser estudiadas.

La fórmula anterior se puede escribir como,

$$x_{n+1} = (\pi + x_n)^5 \bmod 1 \quad n \geq 0$$

que es una ecuación congruencial.

Definición de números congruentes

Un número entero x es congruente con otro y modulo m si $x - y$ es divisible por m , es decir x e y dan el mismo resto cuando se dividen por m .

Se representa por $x \equiv y$.

Sean a, b y $m \in \mathbb{Z}$ y $x_0 \in \mathbb{Z}$ que recibe el nombre de semilla la ecuación que determinan es,

$$x_{j+1} = (ax_j + b) \bmod m$$

La operación “modulo m ” consiste en dividir $(ax_j + b)$ entre m y guardar el resto.

Ejemplo: calculamos $6 \bmod 5 \Rightarrow \frac{6}{5} = 5 \cdot 1 + 1 \Rightarrow 6 \bmod 5 = 1$

Las secuencias obtenidas por una ecuación congruencial son cíclicas y su ciclo máximo posible está acotado superiormente por el valor de m .

Teorema de Knuth: Máximo periodo de un Generador Congruencial General con $b \neq 0$

Las siguientes condiciones son necesarias y suficientes para que un *Generador Congruencial General con parámetros m, a y b* , $x_{j+1} = (ax_j + b) \bmod m$, tenga período máximo ($p = m$)

- a) b y m son primos entre sí (coprimos) ($m. c. d (m, b) = 1$).
- b) $a - 1$ es múltiplo de todos los factores primos de m ($a \equiv 1 \bmod g$ para todo factor g primo de m)
- c) Si m es múltiplo de 4, entonces $a - 1$ también lo ha de ser

Ejemplo 1:

1.- Tomando $m = 16$, $a = 5$, $b = 3$ y $x_0 = 0$. ¿El generador $x_{j+1} = (5x_j + 3) \pmod{16}$ es de máximo ciclo? Vamos a comprobar si se verifica las condiciones del teorema de ciclo máximo.

1.- b y m son primos entre sí (no tienen factores comunes).

El factor primo de 16 es 2, ya que $16 = 2^4$
El factor primo de $b = 3$ } $\Rightarrow b$ y m son coprimos

2.- $c = a - 1$ es múltiplo de p para cada p , número primo, factor de m .

$a - 1 = 4$ y es múltiplo de los factores de $m = 16 = 2^4$

3.- c es un múltiplo de 4 si m es múltiplo de 4.

En este caso $m = 16$ es múltiplo de 4 y $(a - 1) = 4$ también es múltiplo de 4.

La conclusión es que este generador congruencial es de ciclo máximo $m = 16$.

Lo comprobamos.

Secuencia	$5x_j + 3$	x_j	u_j
x_0	0	0	$\frac{0}{16} = 0$
x_1	3	3	$\frac{3}{16} = 0.1875$
x_2	18	2	$\frac{2}{16} = 0.125$
\vdots	\vdots	\vdots	\vdots
x_{16}	48	0	$\frac{0}{16} = 0$

La elección de $x_0 = 0$ no es importante. Se puede comenzar con un x_0 arbitrario y se obtendrá una secuencia de periodo m . En R,

```
# Inicializar parámetros
m <- 16
a <- 5
b <- 3
x0 <- 0 # mirar la función set.seed()

# Variables auxiliares
secuencia <- x0
anterior <- x0

for(i in 2:(m+1)){
  actual <- (a*anterior + b) %% m
  secuencia[i] <- actual
  anterior <- actual

  # Números aleatorios
  unif <- secuencia/m
}
```

```
# Resultado
resultado <- data.frame(sec = paste("x", 0:m, sep=""),
                        an = secuencia,
                        un = unif)

resultado

##   sec an    un
## 1  x0  0 0.0000
## 2  x1  3 0.1875
## 3  x2  2 0.1250
## 4  x3 13 0.8125
## 5  x4  4 0.2500
## 6  x5  7 0.4375
## 7  x6  6 0.3750
## 8  x7  1 0.0625
## 9  x8  8 0.5000
##10  x9 11 0.6875
##11 x10 10 0.6250
##12 x11  5 0.3125
##13 x12 12 0.7500
##14 x13 15 0.9375
##15 x14 14 0.8750
##16 x15  9 0.5625
##17 x16  0 0.0000
```


2.- Tomando $m = 10$, $a = 103$, $b = 17$ y $x_0 = 2$. El generador congruencial es

$$x_{j+1} = (103x_j + 17) \pmod{10}$$

Vamos a comprobar si se verifica las condiciones del teorema de ciclo máximo.

1.- b y m son primos entre sí (no tienen factores comunes).

El factor primo de 10 es 2 y 5, ya que $10 = 2 \cdot 5$
El factor primo de $b = 17$ } $\Rightarrow b$ y m son coprimos

2.- $(a - 1)$ es múltiplo de p para cada p , número primo, factor de m .

$(a - 1) = 102$ no es múltiplo de los factores de $m = 17$. No cumple la condición 2

3.- c es un múltiplo de 4 si m es múltiplo de 4.

En este caso $m = 10$ que no es múltiplo de 4 por lo tanto no hay que comprobar si $(a - 1) = 102$ es múltiplo de 4.

La conclusión es que este generador congruencial no es de ciclo máximo $m = 10$.

Lo comprobamos.

```
# Inicializar parámetros
m <- 10
a <- 103
b <- 17
x0 <- 5

# Variables auxiliares
secuencia <- x0
anterior <- x0

for(i in 2:(m+1)){
  actual <- (a*anterior + b) %% m
  secuencia[i] <- actual
  anterior <- actual

  # Números aleatorios
  unif <- secuencia/m
}

# Resultado
resultado <- data.frame(sec = paste("x", 0:m, sep=""),
                       an = secuencia,
```

```

                                un = unif)
resultado
##      sec an  un
## 1   x0  5 0.5
## 2   x1  2 0.2
## 3   x2  3 0.3
## 4   x3  6 0.6
## 5   x4  5 0.5
## 6   x5  2 0.2
## 7   x6  3 0.3
## 8   x7  6 0.6
## 9   x8  5 0.5
## 10  x9  2 0.2
## 11 x10  3 0.3

```

El ciclo en este caso es de longitud 4.

1.1. Generadores Congruenciales Lineales Multiplicativos

Los generadores congruenciales lineales multiplicativos (G.C.M.) son un caso particular de los generadores congruenciales lineales tomando $b = 0$. La expresión del generador es

$$x_{j+1} = (ax_j) \bmod m$$

Los algoritmos que se obtiene con este generador son más rápidos, aunque su periodo máximo es de $(m - 1)$.

1.2. Otros generadores congruenciales

Los generadores congruenciales lineales son un caso particular de los generadores,

$$x_{n+1} = f(x_i, x_{i-1}, \dots) \bmod m$$

donde f es una función sobre los valores anteriores x_i, x_{i-1}, \dots

Generador congruencial cuadrático

Viene definido por la expresión:

$$X_{n+1} = (a_1 \cdot X_n^2 + a_2 \cdot X_n + c) \bmod m .$$

El periodo máximo es m .

Generador congruencial lineal generalizado

Viene definido por la expresión:

$$X_n = (a_1X_{n-1} + a_2X_{n-2} + \dots + a_kX_{n-k}) \text{ mod } m .$$

El periodo máximo es de longitud $m^k - 1$ si m es primo.

Generadores compuestos

Generador Tausworthe

$$X_n = (a_1X_{n-1} + a_2X_{n-2} + \dots + a_kX_{n-k}) \text{ mod } 2 .$$

donde $x_i \in \{0,1\}$. Este tipo de generadores produce secuencias de bits

Generador aditivo de Green

$$X_{n+1} = (X_n + X_{n-k}) \text{ mod } m$$

Para $k = 1$, se obtiene el generador aditivo de Fibonacci.

Generador de Mersenne-Twister

Es un generador con un periodo de $2^{19937} - 1$ con muy buenas características estadísticas, aunque requiere una gran cantidad de recursos tanto de computación como de memoria. Es una forma generalizada de generadores por desplazamiento de registros y generadores retardados del tipo de Fibonacci.

Ejemplo de un generador múltiple combinado es el siguiente

$$X_i = (1403580 \cdot X_{i-2} - 810728 \cdot X_{i-3}) \text{ mod } 4294967087$$

$$Y_i = (527612 \cdot Y_{i-2} - 1370589 \cdot Y_{i-3}) \text{ mod } 429494443$$

y se considera como

$$Z_i = (X_i - Y_i) \text{ mod } 4294967087 \text{ mod } i = 1, 2, \dots$$

los números aleatorios U_i son

$$U_i = \frac{Z_i}{4294967088} \quad \text{si } Z_i > 0$$

$$U_i = 1 \quad \text{si } Z_i = 0$$

la longitud del ciclo es 3.1×10^{57} .

2. Calidad de los Generadores de números aleatorios.

Los resultados anteriores garantizan las condiciones en las que se puede obtener la máxima longitud de ciclo de un generador congruencial, sin embargo, la “calidad” de la secuencia obtenida es más importante que la longitud de la misma. La secuencia definida por los números impares= $\{1, 3, 5, 7, \dots\}$ es de longitud infinita pero no se podría utilizar en ningún modelo de simulación.

Las condiciones 1 y 2 de los números aleatorios:

1. Tener una distribución estadística.
2. Ser independientes.

son características estadísticas que pueden ser comprobadas por medio de contrastes de hipótesis. La clase de contrastes que emplean son:

- **Test de ajuste:**
 - test de la χ^2
 - test de Kolmogorov-Smirnov.
- **Test de aleatoriedad:**
 - test de las rachas,
 - test de la mediana
 - test gap
 - test del Póker.
- **Test espectrales.**
 - test de autocorrelación

En el caso de los números pseudoaleatorios la hipótesis nula es que la secuencia de números pseudoaleatorios se distribuye según una $\mathcal{U}(0, 1)$ y la alternativa que no es así.

Los tests de ajuste y de aleatoriedad son de uso general en estadística, pero el tercero es específico de los números pseudoaleatorios (y de otros temas como series temporales).

2.1. Test de bondad de ajuste

2.1.1. Contrates de la χ^2 para la bondad de ajuste

Supongamos una variable aleatoria X con k categorías y función de distribución $F_X(x) = F_0(x)$.

El test de la χ^2 compara las frecuencias observadas dentro de una muestra, con las frecuencias esperadas suponiendo que es cierta la hipótesis nula. Planteamos el test,

$$\begin{aligned}H_0: & F_X(x) = F_0(x) \\H_1: & F_X(x) \neq F_0(x)\end{aligned}$$

Se define el estadístico,

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

donde,

O_i número de casos observados en la categoría i .

E_i número de casos esperados en la categoría i .

k número de categorías.

Si las frecuencias esperadas y observadas están próximas $(O_i - E_i)^2$ estará cerca de cero y el valor del estadístico χ^2 será pequeño. En caso contrario el valor de la χ^2 será grande.

Bajo la hipótesis nula, χ^2 se distribuye según una χ_{k-r-1}^2 , donde r es el número de parámetros a estimar. La aproximación se considera aceptable si $\min E_i > 5$.

Ejemplo 3: New York Post.

En las carreras de caballos celebradas en hipódromos circulares se sospecha que la posición del caballo en los cajones de salida tiene influencia en la posibilidad de ganar la carrera. El cajón 1º es el más cercano a la valla interna y el cajón 8 el más alejado. Vamos a contrastar el efecto de la posición analizando los resultados de las carreras.

El test queda definido por:

H_0 : No hay diferencia en el número de ganadores en función del cajón de salida.

H_1 : Si hay diferencia.

Región de rechazo: $P(\chi_{k-1}^2 \geq x) \leq \alpha$, con α nivel de significación.

Datos

Tenemos una muestra de 144 ganadores distribuidos como sigue.

	Posiciones de salida							
	1	2	3	4	5	6	7	8
Núm. Victorias	29	19	18	25	17	10	15	11
Núm. Esperado	18	18	18	18	18	18	18	18

$$\chi^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i} = \frac{(29 - 18)^2}{18} + \dots + \frac{(11 - 18)^2}{18} = 16,3$$

$$\chi^2 \sim \chi_{k-1}^2 = \chi_7^2 \text{ y } P(\chi_7^2 \geq 16.3) > 0.02$$

A nivel $\alpha = 0.01$ se acepta H_0 .

Ejemplo 4

Un investigador pasa un test de vocabulario a un grupo de $N = 105$ niños. Él presupone que la puntuación del test se distribuye según una normal. La media muestral es de 108 y la desviación típica es de 12,8.

Para poder aplicar el test de ajuste de la χ^2 debemos definir categorías y determinar las frecuencias esperadas. Tomamos $k = 10$ intervalos de frecuencia. Los valores de corte X_{corte} corresponden a los deciles de la distribución normal con media y desviación típica muestrales. Los deciles se pueden obtener de una tabla de la normal,

CATEGORÍA	Z_{corte}	Probabilidad Acumulada	X_{corte}
1	-1.2816	0.10	91.60
2	-0.8416	0.20	97.23
3	-0.5244	0.30	101.29
4	-0.2534	0.40	104.76
5	0.0000	0.50	108.00
6	0.2534	0.60	111.24
7	0.5244	0.70	114.71
8	0.8418	0.80	118.77
9	1.2816	0.90	124.40
10	∞	1.00	∞

En general, $X_{corte} = \bar{X} + S_X Z_{corte}$

Aquí, $X_{corte} = 108 + 12.8Z_{corte}$

Se obtienen los siguientes datos,

1	2	3	4	5	6	7	8	9	10
8	10	13	15	10	14	12	8	7	6

La frecuencia esperada por categoría es, $\frac{N}{k} = \frac{103}{10} = 10.3$. Fijando $\alpha = 0.05$.

$$\chi^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i} = \frac{(8 - 10.3)^2}{10.3} + \dots + \frac{(6 - 10.3)^2}{10.3} = 8.36$$

$\chi^2 \sim \chi_{k-r-1}^2 = \chi_7^2$. Bajo H_0 , $P(\chi_7^2 \geq 14.07) > 0.05$. Luego, $P(\chi_7^2 \geq 8.36) > 0.05$

Se acepta H_0 .

En nuestro caso, F_0 es la $U(0, 1)$, $r = 0$ y podemos escoger k sub-intervalos de $[0, 1]$ de igual longitud, $E_i = \frac{n}{k}$.

2.1.2. Contraste de bondad de ajuste de Kolmogorov-Smirnov

Es un contraste de bondad de ajuste que mide el grado de similitud entre la distribución de una muestra y una determinada distribución teórica, en este caso la distribución $\mathcal{U}(0,1)$. El test de Kolmogorov-Smirnov supone que F_0 es continua.

Planteamos el test,

$$\begin{aligned}H_0: & F_0(x) = S_n(x) \\H_1: & F_0(x) \neq S_n(x)\end{aligned}$$

Donde S_n es la distribución de frecuencias relativas de una muestra de tamaño n , $S = \frac{\text{card}\{X_i \leq x\}}{n}$, y $F_0(x_i)$ es la proporción esperada de observaciones que son menores o iguales que x_i . Bajo la hipótesis nula la diferencia entre $F_0(x_i)$ y $S_n(x)$ serán pequeñas. El test de Kolmogorov-Smirnov se fija en las máximas desviaciones que se obtienen.

Se define el estadístico,

$$D = \max |F_0(x_i) - S_n(x_i)| \quad i = 1, 2, \dots, n$$

la distribución de D está tabulada para valores ≤ 40 . Para valores mayores que 40 se utiliza la distribución asintótica.

$$\lim_{n \rightarrow \infty} P(\sqrt{n} D_n \leq z) = L(z) = 1 - 2 \sum_{i=1}^{\infty} (-1)^{i-1} e^{-2iz^2} \quad \forall z \geq 0$$

En nuestro caso particular si utilizamos los estadísticos de orden queda,

$$X_{(1)} < X_{(2)} < \dots < X_{(n)}$$

y tenemos, $F_0(x_{(i)}) = x_{(i)}$ y $S_n(x_{(i)}) = \frac{i}{n}$, por lo que resulta,

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

Ejemplo 5

1.- Un dado se lanza 15 veces con el resultado siguiente:

Valor	1	2	3	4	5	6
Frecuencia	0	1	4	0	4	6

$$H_0: P(X = x) = \frac{1}{6} \quad x = 1, 2, \dots, 6$$

$$H_1: P(X = x) \neq \frac{1}{6} \quad x = 1, 2, \dots, 6$$

X:		<1	1	2	3	4	5	6	>6
S ₁₅		0	0	1/15	5/15	5/15	9/15	1	1
F(x)		0	1/6	1/6	1/6	1/6	1/6	1/6	1/6
S ₁₅ - F(x)		0	5/30	8/30	5/30	10/30	7/30	0	0

Entonces,

$$D_{15} = \max|S_{15}(x_i) - F(x_i)| = \frac{10}{30} = 0.333$$

Como $D_{15,0.10} = 0.304$ (es decir $D_{15} > D_{15,0.10}$) rechazamos H_0 .

2.- Unos investigadores han estudiado la duración de diferentes sucesos como trabajos, huelgas o guerras. Una parte de los investigadores creen que la duración de estos eventos puede ser representada por un modelo matemático que se concreta en una cierta función de distribución. Se disponen de los datos de relativos a la duración de las huelgas en U.K. desde 1965 y se quiere contrastar si las predicciones del modelo matemático están de acuerdo con los datos observados.

El test que planteamos es,

H_0 : La duración de las huelgas sigue el modelo matemático

H_1 : No lo sigue

Fijamos un nivel de significación de $\alpha = 0.05$ y el tamaño de la muestra es $N = 840$.

Máxima duración (días)	Frecuencia acumulada		Frecuencia acumulada relativa		
	Observada	Predicha	Observada	Predicha	$ F_0(x_i) - S_n(x_i) $
1-2	203	212.81	0.242	0.253	0.011
2-3	352	348.26	0.419	0.415	0.004
3-4	452	442.06	0.538	0.526	0.012
4-5	523	510.45	0.623	0.608	0.015
5-6	572	562.15	0.681	0.669	0.012
6-7	605	602.34	0.720	0.717	0.003
7-8	634	634.27	0.755	0.755	0.000
8-9	660	660.10	0.786	0.786	0.000
9-10	683	681.32	0.813	0.811	0.002
10-11	697	698.97	0.830	0.832	0.002
11-12	709	713.82	0.844	0.850	0.006
12-13	718	726.44	0.855	0.865	0.010
13-14	729	737.26	0.868	0.878	0.010
14-15	744	746.61	0.886	0.889	0.003
15-16	750	754.74	0.893	0.899	0.006
16-17	757	761.86	0.901	0.907	0.006
17-18	763	768.13	0.908	0.914	0.006
18-19	767	773.68	0.913	0.921	0.008
19-20	771	778.62	0.918	0.927	0.009
20-25	788	796.68	0.938	0.948	0.010
25-30	804	807.86	0.957	0.962	0.005
30-35	812	815.25	0.967	0.971	0.004
35-40	820	820.39	0.976	0.977	0.001
40-45	832	826.86	0.990	0.984	0.006
>50	840	840.01	1.000	1.000	0.000

$D_N = 0.015$ como $N \gg 35$ el valor crítico que obtenemos es $\frac{136}{\sqrt{840}} = 0.047$ como $0.015 < 0.047$. No rechazamos H_0 .

2.2. Contrastes de aleatoriedad

2.2.1. Contraste de las rachas arriba y abajo

El test de rachas examina la disposición de los números para comprobar la hipótesis de independencia. Se define una “racha” como la sucesión de un *evento* precedido y seguido por un *evento* diferente. Por ejemplo, al lanzar una moneda de forma sucesiva podemos obtener una sucesión de caras y cruces,

C C X X C C C X C C C C X C X C X X X X

donde el primer suceso y el último están precedidos y seguidos por un “no suceso”. Esta secuencia tiene 10 rachas con longitud: 2, 2, 3, 1, 4, 1, 1, 1, 1, 4. Las características que nos interesan de una secuencia son:

- Número de rachas
- Longitud de las rachas

además se dice que una racha es “hacia arriba” cuando es seguida de otra con mayor longitud y es “hacia abajo” cuando es seguida por otra de menor longitud.

Bajo la hipótesis nula, sea X la variable aleatoria que mide el número total de rachas de una secuencia de longitud N , se verifica que:

$$\mu_X = \frac{2N - 1}{3} \quad y \quad \sigma_X^2 = \frac{16N - 29}{90}$$

Para valores de $N > 20$ la distribución asintótica de $X \sim N(\mu_X, \sigma_X^2)$.

2.2.1. Contraste de media

Otro test de rachas más preciso para comprobar aleatoriedad es aquel que cuenta el número de valores que hay por encima y por debajo de la media (en general de cualquier cuantil).

Sea n_1 y n_2 el número de observaciones que están por encima y por debajo de la media, respectivamente. Sea Y la variable aleatoria que mide el número total de rachas de una secuencia de longitud N , se verifica que:

$$\mu_Y = \frac{2n_1n_2}{N} + \frac{1}{2} \quad y \quad \sigma_Y^2 = \frac{2n_1n_2(2n_1n_2 - N)}{N^2(N - 1)}$$

Para valores de $N > 20$ la distribución asintótica de $Y \sim N(\mu_X, \sigma_X^2)$.

2.2.2. Contraste del Póker

Este contraste recibe el nombre del juego del póker. Se utiliza para contrastar la aleatoriedad de números aleatorios con 5 cifras o bien si tienen más de 5 dígitos se truncan a cinco o se repite dividiéndolo en grupos de cinco. La idea en la que se basa es la siguiente.

Disponemos de una baraja de 10 cartas,

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9$$

y jugamos con 5 mazos.

En estas condiciones al repartir 5 cartas las posibles jugadas son $10^5 = VR_{10,5}$, estos son los casos posibles. Las jugadas posibles son:

Jugada	Casos Favorables	Probabilidad
Cinco cartas distintas	$10 \cdot 9 \cdot 8 \cdot 7 \cdot 6 = 3024$	0.3024
Pareja	$10 \cdot 1 \cdot 8 \cdot 7 \cdot 6 \binom{5}{2} = 50400$	0.50400
Doble pareja	$\frac{1}{2} \cdot 10 \cdot 1 \cdot 9 \cdot 1 \cdot 8 \binom{5}{2} \binom{2}{2} = 10800$	0.10800
Trio	$10 \cdot 1 \cdot 1 \cdot 9 \cdot 8 \binom{5}{3} = 720$	0.07200
Full	$10 \cdot 1 \cdot 1 \cdot 9 \cdot 1 \binom{5}{4} \binom{2}{2} = 90$	0.00900
Póquer	$10 \cdot 1 \cdot 1 \cdot 1 \cdot 9 \binom{5}{4} = 45$	0.004500
Repóquer	$10 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \binom{5}{5} = 10$	0.00010

Dada una secuencia de números aleatorios de cinco dígitos se calculan las frecuencias observadas de cada una de las jugadas y junto con las frecuencias esperadas se construye el estadístico,

$$\chi_0^2 = \sum_{i=1}^7 \frac{(O_i - E_i)^2}{E_i} \sim \chi_6^2$$

Ejemplo 6: Contraste del Poker

Dado el generador congruencial multiplicativo

$$x_j = 7^5 x_{j-1} \text{ mod } (2^{31} - 1)$$

generar una secuencia de 20 números pseudoaleatorios y comprobar con el contraste del Poker que son aleatorios.

En R, la función `poker.test` {randtoolbox} realiza el contraste de Poker, aunque permite que la *mano* sea de distinto tamaño.

Ejemplo 7: Contraste del Poker en R

Sintaxis de la función `poker.test()`

```
poker.test(u, nbcards = 5, echo = TRUE)
```

donde

- `u` = muestra de números aleatorios en (0,1)
- `nbcards` = número de cartas que se reparte. Debe ser un múltiplo de `u`
- `echo` = variable lógica. Si es TRUE muestra el detalle de los resultados

```
library(randtoolbox)

# Mano de 5 cartas
poker.test(runif(50000))

##
##          Poker test
##
## chisq stat = 4.8, df = 4, p-value = 0.31
##
##      (sample size : 50000)
##
## observed number  18 1012 4736 3866 368
## expected number  16 960 4800 3840 384

# Mano de 4 cartas
poker.test(rnorm(40000), nbcards = 4)

##
##          Poker test
##
## chisq stat = 144802, df = 3, p-value = 0
##
##      (sample size : 40000)
##
## observed number  4818 2794 466 12
## expected number  156 3281 5625 938

# Mano de 42 cartas
poker.test(runif(420000), nbcards = 42)
```

```

##
##          Poker test
##
## chisq stat = 11, df = 41, p-value = 1
##
##          (sample size : 420000)
##
## observed number  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 35 125 387 792 1383 1804 19
30 1615 1088 567 188 62 15 1 0 0 0 0 0 0 0
## expected number  2.8e-63 2.5e-49 8.4e-41 1.4e-34 1.3e-29 1.7e-25 5.5e-22 6.5e-19 3.3
e-16 8.7e-14 1.3e-11 1.1e-09 6.4e-08 2.4e-06 6.5e-05 0.0012 0.017 0.17 1.4 8 37 131 363
791 1357 1833 1949 1628 1064 542 213 64 15 2.5 0.3 0.027 0.0016 6e-05 1.4e-06 1.6e-08 8.
1e-11 9.4e-14

```

2.2.3. Contraste Gap (de salto)

El contraste del salto se utiliza para determinar la significación de la longitud de los saltos que aparecen entre la aparición de un mismo dígito.

Se dice que hay un salto de longitud x cuando hay x posiciones entre la aparición del mismo dígito.

La probabilidad de un determinado salto se puede calcular como un experimento de Bernoulli. Supongamos una secuencia y queremos calcular la probabilidad de que el dígito 3 de un salto de longitud n

$$\Pr(\text{del dígito 3 de un salto de } n) = \Pr(d \neq 3) \cdot \Pr(d \neq 3) \cdots \Pr(d \neq 3) \cdot \Pr(d = 3)$$

suponiendo el sistema decimal $(0, 1, 2, \dots, 9)$

$$\Pr(\text{del dígito 3 de un salto de } n) = 0.9^n \times 0.1$$

luego su función de distribución es,

$$F(x) = \Pr(3 \text{ de un salto} \leq x) = 0.1 \sum_{i=0}^x 0.9^i = 1 - 0.9^{x+1}$$

los pasos a seguir son:

Paso 1. Se especifica la distribución teórica del dígito.

Paso 2. Se determina la función de distribución empírica

Paso 3. Se calcula la máxima diferencia entre la función de distribución teórica y empírica

Paso 4. Se calcula el p-valor asociado

2.3. Contrastes espectrales

2.3.1. Test de autocorrelación

El test de autocorrelación contrasta la dependencia entre números en una secuencia dada. El contraste calcula la autocorrelación entre cada m números empezando con el número i -ésimo.

El valor es ρ_{im} se calcula entre,

$$R_i, R_{i+m}, R_{i+2m}, R_{i+3m}, \dots$$

el valor M es el mayor entero tal que

$$i + (M + 1)m \leq N$$

donde N es la longitud de la secuencia.

Dado que si la autocorrelación es distinta de cero hay ausencia de independencia, el contraste que se propone es,

$$H_0: \rho_{im} = 0$$

$$H_1: \rho_{im} \neq 0$$

Para valores grandes de M , la distribución del estimador de ρ_{im} , que denotamos por $\hat{\rho}_{im}$ se distribuye asintóticamente como una $N(0, \sigma_{\hat{\rho}_{im}})$ si los valores $R_i, R_{i+m}, R_{i+2m}, R_{i+3m}, \dots$ son incorrelados.

El estadístico del test es,

$$Z_0 = \frac{\hat{\rho}_{im}}{\sigma_{\hat{\rho}_{im}}} \sim N(0,1)$$

La expresión del estadístico y su varianza son:

$$\hat{\rho}_{im} = \frac{1}{M+1} \left[\sum_{k=0}^M R_{i+km} R_{(k+1)m} \right] - 0.25 \quad \text{y} \quad \sigma_{\hat{\rho}_{im}} = \frac{\sqrt{13M+7}}{12(M+1)}$$

3. Generador de números aleatorios en software

En R, el generador de números aleatorios es la función `runif()`. Aunque R no utiliza un generador congruencial necesita una semilla para generar números pseudoaleatorios.

Para poder repetir una secuencia de números pseudoaleatorios es necesario conocer:

- La semilla
 - Fijada con `set.seed()`
 - Recuperada con (`.Random.seed()`)
- El algoritmo que utiliza con `RNGkind()`

Ejemplo

Genera 20 números pseudoaleatorios utilizando el generador congruencial multiplicativo

$$x_n = 172 \cdot x_{n-1} \text{ mod } 30307$$

con semilla inicial $x_0 = 17218$. Comprueba la calidad del generador.

Solución

Generación de los números aleatorios

```
# Semilla
X0 <- 17218

# Números a generar
Nsim <- 20
X <- numeric(Nsim)
X[1] <- (172*X0) %% 30307
for(i in 2:Nsim) {X[i] <- (X[i-1] * 172) %% 30307}
U <- X/30307
U

## [1] 0.71656713 0.24954631 0.92196522 0.57801828 0.41914409 0.09278385
## [7] 0.95882139 0.91727984 0.77213185 0.80667833 0.74867192 0.77157092
## [13] 0.71019896 0.15422180 0.52614907 0.49764081 0.59421916 0.20569505
## [19] 0.37954928 0.28247600
```

Test de Bondad de ajuste

```
ks.test(U, punif )

##
## One-sample Kolmogorov-Smirnov test
##
## data: U
```

```
## D = 0.1602, p-value = 0.627
## alternative hypothesis: two-sided
```

Test de Poker

```
library(randtoolbox)
```

```
## Warning: package 'randtoolbox' was built under R version 3.3.3
```

```
## Loading required package: rngWELL
```

```
## Warning: package 'rngWELL' was built under R version 3.3.3
```

```
## This is randtoolbox. For overview, type 'help("randtoolbox")'.
```

```
# Mano de 5 cartas
```

```
poker.test(U)
```

```
##
##          Poker test
##
## chisq stat = 1.7, df = 4, p-value = 0.79
##
##      (sample size : 20)
##
## observed number  0 1 1 2 0
## expected number 0.0064 0.38 1.9 1.5 0.15
```

Test de gap

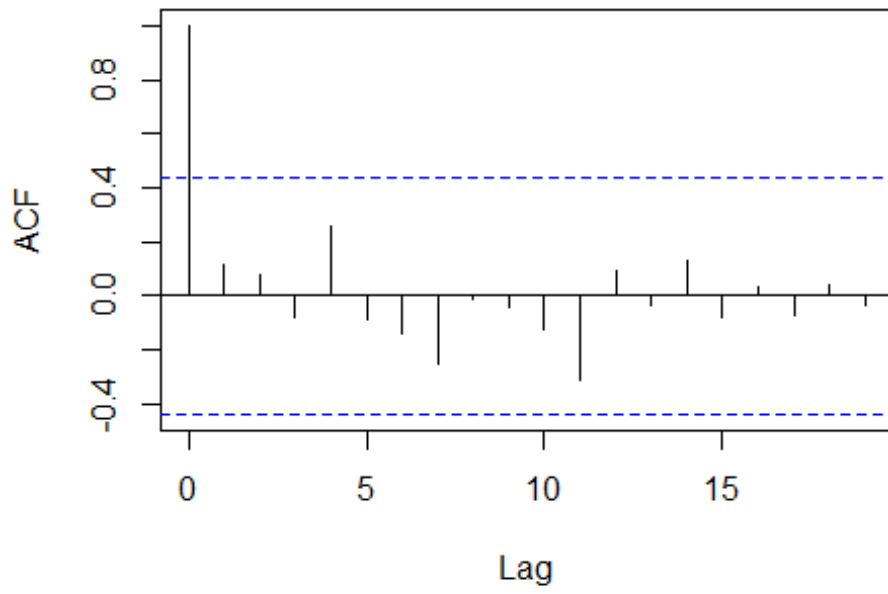
```
gap.test(U)
```

```
##
##          Gap test
##
## chisq stat = 0.84, df = 4, p-value = 0.93
##
##      (sample size : 20)
##
## length  observed freq      theoretical freq
## 1          3          2.5
## 2          1          1.2
## 3          1          0.62
## 4          0          0.31
## 5          0          0.16
```

Función de autocorrelación

```
acf(U, lag.max = 20)
```

Series U



4. Aplicaciones de los números aleatorios

4.1. Resolución de problemas de probabilidades

Probabilidad de obtener exactamente dos caras en tres lanzamientos de una moneda.

Solución mediante un experimento real

Realizar el experimento en una situación real consiste en repetir un número suficientemente grande de veces los tres lanzamientos y observar con qué frecuencia se obtienen exactamente dos caras.

El sistema real es el mecanismo por el cual se realizan los lanzamientos.

Solución teórica

Ejemplo 6: Un modelo teórico es considerar una variable aleatoria $X \sim Bin(3,0.5)$ y calcular la $P(X = 2)$. En este caso la solución es,

$$P(X = 2) = \binom{3}{2} 0.5^2 \cdot 0.5^1 = \frac{3}{8} = 0.375$$

Solución mediante Simulación

Consiste en simular el lanzamiento de tres monedas un número suficiente de veces, y anotar con qué frecuencia se obtiene dos caras.

Un lanzamiento se puede obtener con el código siguiente,

```
sample(x = c(0,1), size = 3, replace = TRUE)
```

donde hemos codificado 0 como cruz y 1 como cara.

El código para realizar 100.000 lanzamientos de tres monedas puede ser,

```
# Semilla
set.seed(1521)

# Simulación Lanzamiento de 3 monedas Nsim veces
Nsim <- 100000 # Número de Lanzamientos

datos <- matrix(numeric(), ncol=3, nrow=Nsim)

for(i in 1:Nsim){
```

```

    datos[i,] <- sample(x = c(0,1), size = 3, replace = TRUE)
}

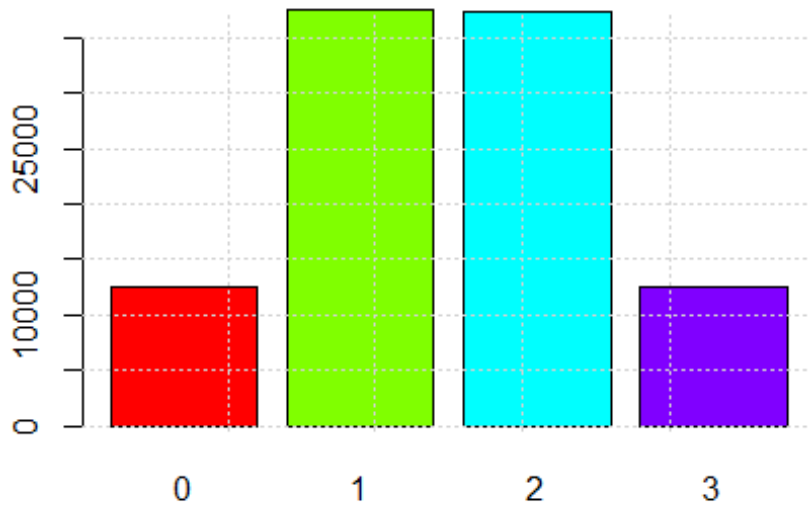
resultado <- rowSums(datos)

# Distribución de resultados
table(resultado)

## resultado
##      0      1      2      3
## 12548 37511 37437 12504

barplot(table(resultado), col=rainbow(4))
grid()

```



```

# hist(resultado, breaks=17,xlab="Valores Generados",main="", ylab="Frecuencia")

# Probabilidad estimada
prob <- length(which(resultado == 2))/Nsim
cat('La probabilidad estimada es ', prob, '\n')

## La probabilidad estimada es 0.37437

```

4.2. Cálculo numérico de integrales por el método de Monte Carlo

Entre las aplicaciones de los números aleatorios se encuentra el cálculo de integrales definidas.

Sea la integral,

$$S = \int_0^1 g(x) dx, \quad \text{con } 0 \leq g(x) \leq c, \quad \forall x \in [0,1]$$

donde c es una cota de la función $g(x)$.

Para el cálculo de S presentaremos dos métodos. El método de acierto-fallo y el de la media muestral, en los dos casos se dará una estimación de S . Sea $\theta \equiv S$ la cantidad que se quiere hallar.

4.2.1. Método de acierto-fallo

Para calcular la integral anterior por medio del método de acierto-fallo realizamos el siguiente experimento.

1. Generamos n puntos (x, y) tales que $(x, y) \in U_{(0,1) \times (0,c)}$
2. Definimos la variable aleatoria W que cuenta los puntos que verifican, $y \leq g(x)$.

Llamando éxito a que un punto verifique la condición y fracaso a que no la verifique. El valor de W es el número de éxitos ocurridos en n pruebas, por lo tanto $W \in B(n, p)$ donde p es la probabilidad de éxito. En estas condiciones,

- $E[W] = n \cdot p$
- $Var(W) = n \cdot p \cdot (1 - p)$

El valor de p se puede expresar como la relación de las áreas del rectángulo definido por $(0,1) \times (1, c)$ y el área definida por el hipografo de la curva $g(x)$ en $(0,1) \times (1, c)$.

$$p = \frac{\text{Área de la integral}}{\text{Área de rectángulo}} = \frac{\int_0^1 g(x) dx}{c} = \frac{S}{c}$$

Un estimador p es,

$$\hat{p}_1 = \frac{W}{n}$$

El estimador \hat{p}_1 es insesgado

$$E(p_1) = \frac{E(W)}{n} = \frac{n \cdot p}{n} = p$$

y su varianza es

$$Var(p_1) = Var\left(\frac{W}{n}\right) = \frac{1}{n^2} Var(W) = \frac{n \cdot p \cdot (1-p)}{n} = \frac{p \cdot (1-p)}{n}$$

y por Teorema Central del Límite

$$\hat{p}_1 = \frac{W}{n} \sim N\left(p, \sqrt{\frac{p \cdot (1-p)}{n}}\right)$$

luego un intervalo de confianza para \hat{p}_1 de nivel $1 - \alpha$ es,

$$p \in \frac{W}{n} \pm z_{\alpha/2} \sqrt{\frac{p \cdot (1-p)}{n}}$$

sustituyendo p por su estimación,

$$p \in \frac{W}{n} \pm z_{\alpha/2} \sqrt{\frac{\frac{W}{n} \left(1 - \frac{W}{n}\right)}{n}}$$

Por último, el valor de la integral es

$$p = \frac{S}{c} \Rightarrow S = c \cdot p \Rightarrow \hat{S}_1 = c \cdot \hat{p}_1$$

y un intervalo de confianza para este estimador del valor de la integral es,

$$S \in c \cdot \left(\frac{W}{n} \pm z_{\alpha/2} \sqrt{\frac{\frac{W}{n} \left(1 - \frac{W}{n}\right)}{n}} \right)$$

En el caso general,

$$S = \int_a^b g(x) dx, \text{ con } 0 \leq g(x) \leq c, \forall x \in [a, b]$$

Entonces, $\hat{p}_1 = \frac{W}{n}$, $E(\hat{p}_1) = p$ y $Var(\hat{p}_1) = \frac{p \cdot (1-p)}{n}$. En esta situación,

$$p = \frac{S}{c \cdot (b-a)} \Rightarrow S = c \cdot (b-a) \cdot p \Rightarrow \hat{S} = c \cdot (b-a) \cdot \hat{p}_1$$

donde

$$\hat{S} = c \cdot (b-a) \cdot \left(\frac{W}{n} \pm z_{\alpha/2} \sqrt{\frac{\frac{W}{n} \left(1 - \frac{W}{n}\right)}{n}} \right)$$

Otra opción es realizar un cambio de variable para transformar el recinto de integración al intervalo $[0,1]$.

$$y = \frac{x-a}{b-a} \Rightarrow dy = \frac{1}{(b-a)} dx \Rightarrow dx = (b-a)dy$$

$$S = \int_0^1 g(x) dx = \int_0^1 g(y)(b-a) dy = (b-a) \int_0^1 g(y) dy$$

Ejemplo 7: $S = \int_0^1 e^{-x^2} dx = 0.7569$.

Método de acierto-fallo

Los cálculos los ordenamos en forma de tabla:

	$x_j \in U(0, 1)$	$y_j \in U(0, c = 1)$	$exp(-x^2)$	Éxito
1	0.58477822	0.31818013	0.55722943	1
2	0.05836590	0.42287308	0.8898238	1
3	0.24783713	0.21981840	0.47544151	1
4	0.04786834	0.32632966	0.82574161	1
5	0.09495608	0.77496894	0.62202164	0
		SUMA	3.37025801	4
		Estimación	0.67405160	0.6666

En R

```
# Generar n puntos (x,y) ~ U(0,1)x(0,1)
Nsim <- 10000
n <- data.frame(x = numeric(Nsim), y = numeric(Nsim),
                g = numeric(Nsim), éxito = numeric(Nsim))

n$x <- runif(Nsim)
n$y <- runif(Nsim)

# función g(x)
c <- 1

g <- Vectorize(function(x) exp(- x^2))

n[3] <- g(n$x)
n[4] <- 1*(n[2] <= n[3])

# Número de éxitos
W <- sum(n[4])/Nsim

# Datos
head(n)
##           x           y           g y
## 1 0.2580588 0.56450590 0.9355747 1
## 2 0.7999220 0.06933147 0.5273583 1
## 3 0.2001411 0.91697132 0.9607352 1
## 4 0.4126511 0.53180103 0.8434278 1
## 5 0.3442973 0.12335169 0.8882157 1
## 6 0.1971113 0.77583710 0.9618922 1
# Estimación de la integral
Int <- W*c
Int
```

```
## [1] 0.7508
```

```
# Gráfico
```

```
# Puntos rojos
```

```
PR <- n[n$éxito == 1,]
```

```
# Puntos verdes
```

```
PV <- n[n$éxito == 0,]
```

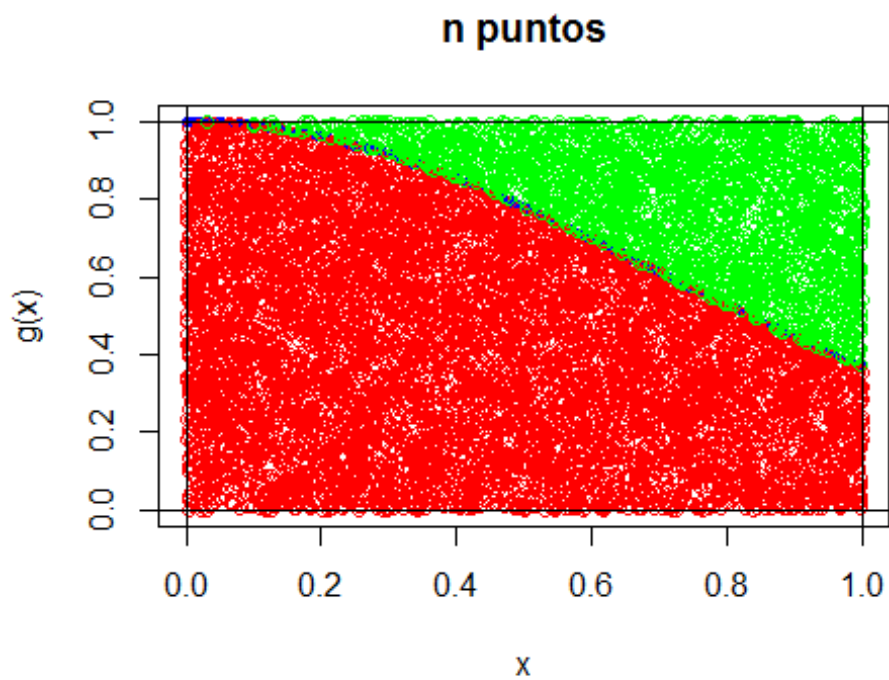
```
curve(g, 0, 1, col = "blue", ylim = c(0,1), lwd = 5, main = "n puntos")
```

```
points(PR$x, PR$y, col = "red")
```

```
points(PV$x, PV$y, col = "green")
```

```
abline(h=c(0,1))
```

```
abline(v=c(0,1))
```



4.2.2. Método de la media muestral

El problema es el mismo, calcular

$$S = \int_0^1 g(x) dx, \text{ con } 0 \leq g(x) \leq c, \forall x \in [0,1]$$

pero lo escribimos como,

$$S = \int_0^1 g(x) \cdot f(x) dx, \text{ con } 0 \leq g(x) \leq c, \forall x \in [0,1]$$

donde $f(x)$ es la función de densidad de una $\mathcal{U}(0,1)$, ($f(x) = 1$). De esta forma,

$$S = \int_0^1 g(x) \cdot f(x) dx = E_U[g(X)]$$

para obtener una estimación de S , se genera una muestra de tamaño n de la distribución $\mathcal{U}(0,1)$ y se calcula:

$$\hat{S}_2 = \frac{1}{n} \sum_{j=1}^n g(x_j)$$

que es un estimador insesgado de $E_U[g(X)]$ (Ley Fuerte de los Grandes Números),

$$E[\hat{S}_2] = \frac{1}{n} \sum_{j=1}^n E[g(x_j)] = \frac{1}{N} \cdot N \cdot \theta = \theta$$

$$V(\hat{S}_2) = \frac{1}{N^2} \sum_{j=1}^n V(g(x_j)) = \frac{\int_0^1 (g(x) - S)^2 dx}{n}$$

pero $V(\hat{\theta}_2)$ no es conocido por lo que lo estimo por,

$$S_{\hat{S}_2}^2 = \frac{1}{n} \cdot \left[\frac{1}{n-1} \sum_{j=1}^n (g(x_j) - \hat{S}_2)^2 \right]$$

Un intervalo de confianza, de nivel $1 - \alpha$, para \hat{S} es,

$$\hat{S} \in \theta_2 \pm z_{\alpha/2} \cdot S_{\theta_2}$$

Ejemplo 8: $S = \int_0^1 e^{-x^2} dx = 0.7569$.

Método de la media muestral

$$S = \int_0^1 e^{-x^2} dx = \int_0^1 e^{-x^2} \cdot 1 \cdot dx = E_U[e^{-x^2}] \approx \frac{1}{n} \sum_{j=1}^n e^{-x_j^2}, \quad \text{donde } x_j \sim U(0,1)$$

Los cálculos los ordenamos en forma de tabla:

	$x_j \in U(0,1)$	$exp(-x^2)$
1	0.58477822	0.55722943
2	0.05836590	0.8898238
3	0.24783713	0.47544151
4	0.04786834	0.82574161
5	0.09495608	0.62202164
		3.37025801
	Estimación:	0.67405160

En resumen, tenemos dos estimadores insesgados, ¿cuál es el que se debe elegir?

4.3. Comparación de los dos métodos

Su pongamos que se desea estimar,

$$S = \int_0^1 g(x) dx, \quad \text{con } 0 \leq g(x) \leq c, \quad \forall x \in [0,1]$$

el estimador del método acierto fallo lo denotamos por \hat{S}_1 ,

$$\hat{S}_1 = c \cdot \frac{W}{n}, \quad \text{donde } E[\hat{S}_1] = S \quad \text{y}$$

$$V(\hat{S}_1) = c^2 \cdot \frac{p(1-p)}{n} = c^2 \left(\frac{\frac{S}{c} \left(1 - \frac{S}{c}\right)}{n} \right) = \frac{cS - S^2}{N}$$

al estimador del método de la media muestral lo denotamos por \hat{S}_2

$$\hat{S}_2 = \frac{1}{n} \sum_{j=1}^n g(x_j), \quad \text{donde } E[\hat{S}_2] = S \quad \text{y} \quad V(\hat{S}_2) = \frac{\int_0^1 (g(x) - S)^2 dx}{n}$$

Entonces,

$$\begin{aligned} V(\hat{S}_1) &= \frac{cS - S^2}{N} = \frac{1}{N}(cS - S^2) = \frac{1}{N} \left[\int_0^1 cg(x) dx - \left(\int_0^1 g(x) dx \right)^2 \right] \\ &\geq \frac{1}{N} \left[\int_0^1 g(x) dx - \left(\int_0^1 g(x) dx \right)^2 \right] = V(\hat{S}_2) \end{aligned}$$

Es decir, de entre los dos métodos el segundo es el de menor varianza y por lo tanto se debe elegir el método de la media muestral.

4.4. Optimización de funciones

El objetivo es hallar el óptimo (máximo o mínimo) de una función en un intervalo o región determinada.

El proceso es:

1. Se genera una secuencia de número aleatorios uniformemente distribuidos en el intervalo o región donde está definida la función.
2. Se evalúa la función en ese conjunto de puntos.
3. Se elige el valor óptimo y el punto donde se alcanza.

Ejemplo 9: Hallar el mínimo de

$$f(x) = \frac{x^4}{4} - \frac{17x^3}{36} + \frac{5x^2}{24}$$

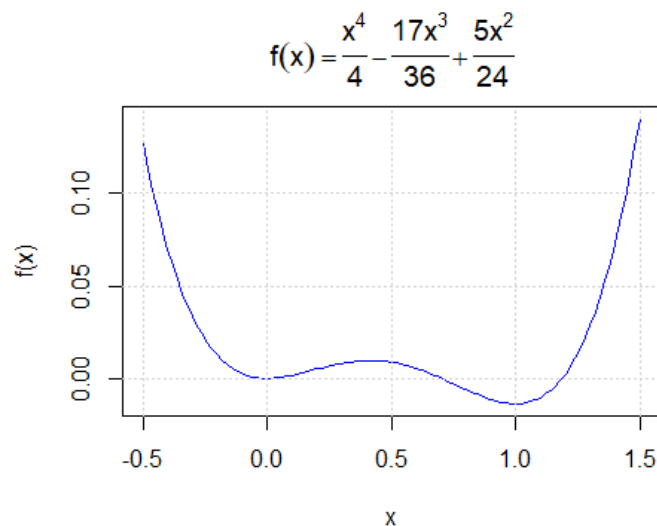
en el intervalo $-0.5 \leq x \leq 1.5$.

En primer lugar, dibujamos la función en el intervalo de definición.

Función a optimizar

```
f <- function(x) x^4/4 - (17*x^3)/36 + (5*x^2)/24
```

```
curve(f, from = -0.5, to = 1.5, col = "blue",  
      main = expression(f(x) == frac(x^4,4) - frac(17*x^3,36) + frac(5*x^2,24))  
      )  
grid()
```



```

# Optimización con simulación
Nsim <- 10
n <- runif(n = Nsim, min = -0.5, max = 1.5)

y <- f(n)
y

## [1] 0.0006500744 0.0517496630 0.0868718581 0.0109640725 -0.0091765981
## [6] -0.0010734598 0.0010536873 0.0154111175 0.0323518016 0.0020842252

min(y)

## [1] -0.009176598

which(y == min(y))

## [1] 5

```

Nota:

El método de Monte Carlo no es de los más eficientes ni para el cálculo de integrales simples ni para la optimización de funciones univariantes siendo más recomendable el uso de métodos de análisis numérico (búsqueda binaria, métodos del gradiente, etc.). Sin embargo, para las integrales múltiples y la optimización de funciones multivariantes puede ser una buena opción.