Integración Monte Carlo

Simulación

2020-10-29

Contents

1 Introducción

En el calculo de integrales por el método de Monte Carlo nos basamos en dos resultados conocidos.

Primero Sea X es una variable aleatoria con función de densidad f (o discreta con función de probabilidad de masa p) y $g : \mathbb{R} \to \mathbb{R}$ una función entonces:

$$E[g(X)] = \int_{-\infty}^{\infty} g(x)f(x)dx \quad X \text{ va continua}$$
$$E[g(X)] = \sum_i g(x_i)p(x_i) \qquad X \text{ va discreta}$$

Segundo

Ley Fuerte de los Grandes números Sea $X_1, X_2, \ldots, X_n$ una secuencia de variables aleatorias independientes e igualmente distribuidas con media μ, entonces,

$$\Pr\left(\lim_{n\to\infty} \frac{X_1 + \cdots + X_n}{n} = \mu\right) = 1$$

Supongamos que se desea calcular una cierta cantidad θ que puede expresarse como

$$\theta = E[g(X)] = \int_{\mathbb{R}} g(x)dx, \text{ tal que } X \text{ va con distribución conocida}$$

el método de Monte Cralo consiste en tomar una muestra $(X_1, X_2, \ldots, X_n)$ independientes y con la misma distribución que X, y estimar el valor de θ según

$$\frac{g(X_1) + g(X_2) + \cdots + g(X_n)}{n} \to_n E[g(X_i)] = E[g(X)] = \theta$$

1.1 Propiedades del estimador $T(\vec{X}) = \frac{1}{n}\sum_{i=1}^{n} g(X_i)$

Esperanza del estimador Dada una muestra aleatoria $X_1, X_2, \ldots, X_n$ el estimador $T(\vec{X}) = \frac{1}{n}\sum_{i=1}^{n} g(X_i)$ es un estimador centrado,

$$E[T(\vec{X})] = E\left[\frac{1}{n}\sum_{i=1}^{n} g(X_i)\right] = E[g(X)]$$

Varianza del estimador La varianza del estimador es

$$Var(T(\vec{X})) = Var\left(\frac{1}{n}\sum_{i=1}^{n} g(X_i)\right) = \frac{1}{n^2}\cdot n\cdot Var(g(X)) = \frac{1}{n}\int_{\mathbb{R}} (g(x) - E[g(X)])^2\, dx = \frac{E[g^2(x)] - E^2[g(X)]}{n} = \frac{E[g^2(x)] - I^2}{n}$$

llamando $\sigma^2_{g(X)} = Var(T(X))$:

- el error que se comete al usar el método de Monte Carlo es de aproximadamente $\frac{\sigma_{g(X)}}{\sqrt{n}}$
- un intervalo de confianza a nivel $1 - \alpha$ es

$$\left[\hat{I} - z_{\alpha/2}\frac{\sigma_{g(X)}}{\sqrt{n}}, \hat{I} + z_{\alpha/2}\frac{\sigma_{g(X)}}{\sqrt{n}}\right]$$

- la longitud del intervalo de confianza es $2 \cdot z_{\alpha/2}\frac{\sigma_{g(X)}}{\sqrt{n}}$

En la práctica, el valor de $\sigma^2_{g(X)}$ es desconocido por lo que debe ser estimado. El estimador de la cuasi varianza muestral es

$$\widehat{\sigma^2_{g(X)}} = \frac{1}{n-1}\sum_{i=1}^{n}(g(X_i) - \hat{I}_n)^2$$

este estimador permite determinar el tamaño muestral n necesario para obtener el valor de la integral con una precisión determinada.

2 Integrales simples

2.1 Integración sobre (0,1)

Sea,

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

Tomamos $f(x) = 1, \quad X \sim \mathcal{U}(0,1)$, entonces

$$I = \int_0^1 g(x)\ dx = \int_0^1 g(x) \cdot f(x)\ dx = \int_0^1 g(x) \cdot \mathbf{1}\ dx = E_{\mathcal{U}(0,1)}[g(X)]$$

- Dadas $U_1, U_2, \cdots$ variables aleatorias e identicamente distribuidas según una $U(0,1)$, entonces $g(U_1), g(U_2), \cdots$ son variables aleatorias e identicamente distribuidas con media I. Por lo tanto,

  $$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} g(U_i) = I$$

  tomamos,

  $$\hat{I} = \frac{1}{n}\sum_{i=1}^{n} g(U_i)$$

Ejemplo 1

Calcular $I = \int_0^1 e^{-x^2} dx = 0.7468$

Algoritmo

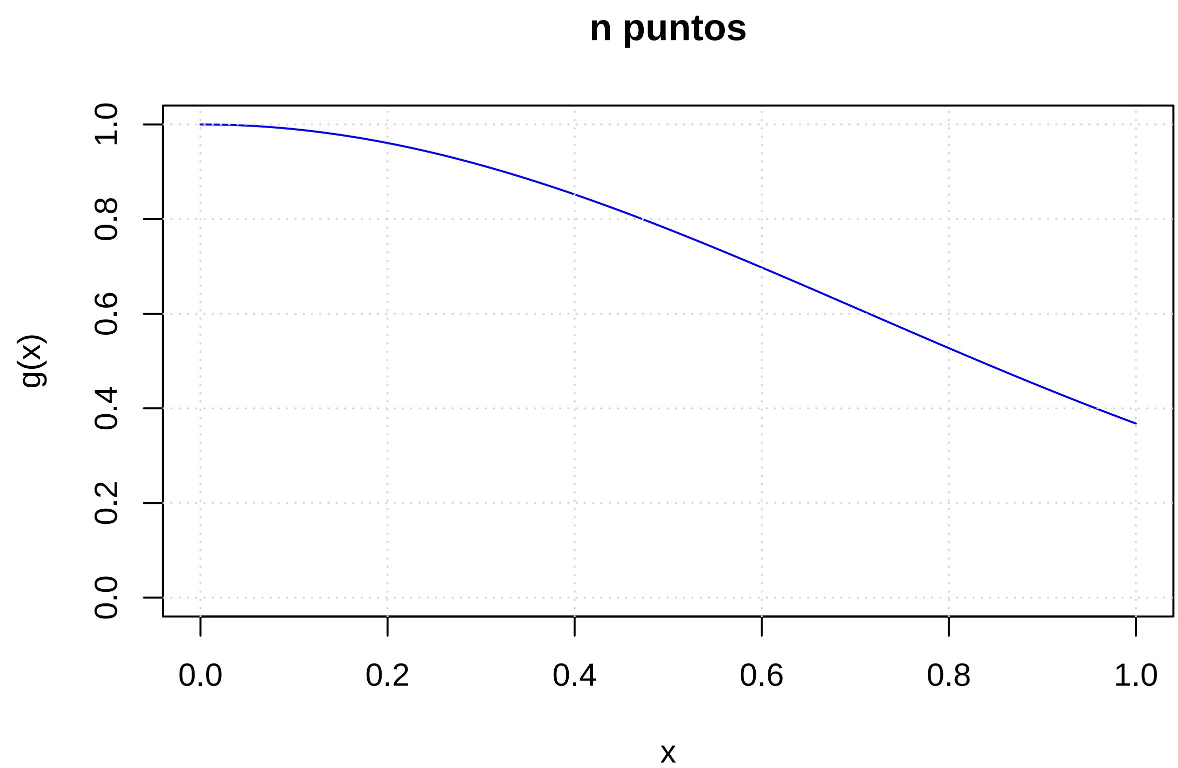
```
1.- Generar U1,...,Un ~ U(0,1)
2.- Hacer   Xi = Ui,          i = 1,...n
3.- Hacer   Yi = 2*exp(- Xi^2), i = 1,...n
4.- Devolver I = sum(Yi)/n
```

```
# Generar n valores de una U(0,1)
set.seed(1714)
n <- 10^4                # Tamaño de la muestra
U <- runif(n)            # Generar n valores de una U(0,1)

# función a integra g(x) = exp{-x^2}

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

# Gráfico de g(x)
curve(g, 0, 1, col = "blue", ylim = c(0,1), lwd = 1, main = "n puntos")
grid()
```




```r
# valores de Y
Y <- g(U)

I.estimado <- mean(Y)
IC_095     <- c(I.estimado - qnorm(0.025, lower.tail = FALSE)*sd(Y)/sqrt(n), I.estimado + qnorm(0.025, l

cat(' Valor estimado  ', I.estimado, "I.C.0.95%", IC_095, '\n')
```

```
##  Valor estimado   0.7451735 I.C.0.95% 0.7412613 0.7490856
```

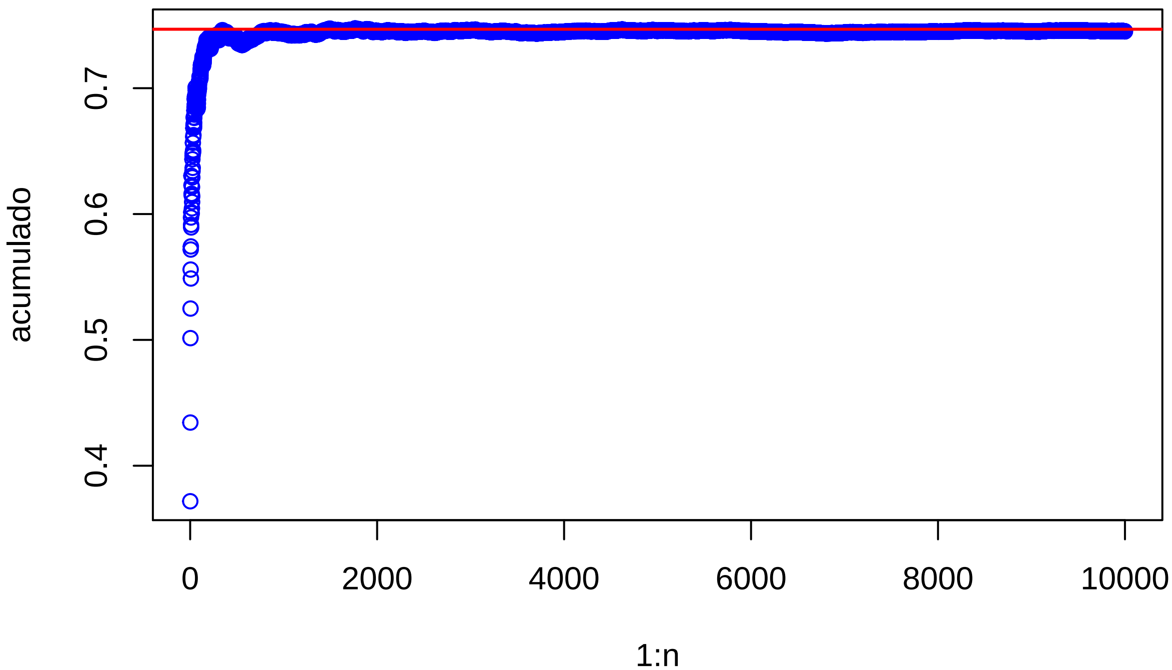
```r
# Valor de la integral

integrate(g, 0,1)
```

```
## 0.7468241 with absolute error < 8.3e-15
```

La evolución de la estimación en función de n se puede apreciar con el siguiente código,

```r
acumulado <- cumsum(Y)/(1:n)
plot(1:n, acumulado, col = "blue")
abline(h = integrate(g, 0,1)$value, col = "red", lwd = 1.5)
```



Evolución del error de la estimación en función de n

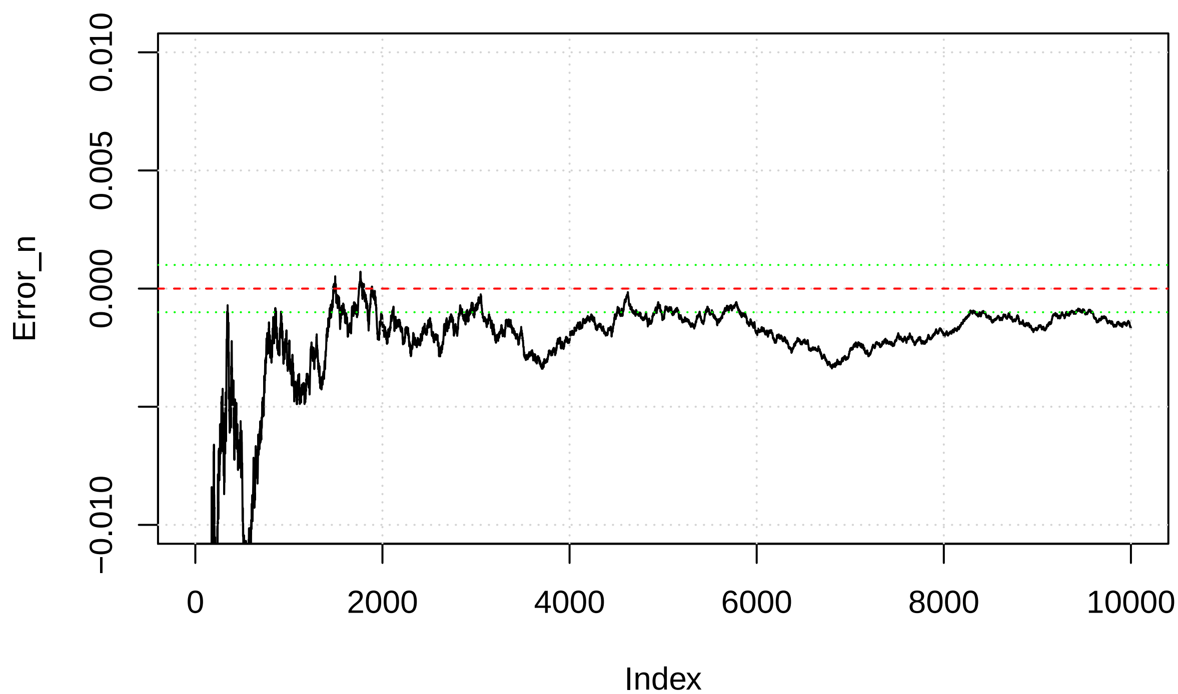
Definimos el error en una muestra de tamaño n como la cantidad, $I - I_n$

```
# error I - I_n

Error_n  <- numeric(n)
I        <- integrate(g, 0, 1)$value

for(i in 1:n){
  Error_n[i] <- mean(Y[1:i]) - I
}

plot(Error_n, ylim = c(-0.01, 0.01), type = "l")
grid()
abline(h = 0, lty = 2, col = "red")
abline(h = c(-0.001, 0.001), lty = 3, col = "green")
```



2.2 Integración sobre (a,b)

Calcular,

$$I = \int_a^b g(x)dx$$

se puede abordar de desde dos puntos de vista

2.2.1 Opción 1: Reducir está integral al caso de una integral entre 0 y 1

Reducimos esta integral al caso anterior mediante un cambio de variable,

Sea

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

obtenemos

$$I = \int_a^b g(x)dx = \int_0^1 g(a+(b-a)\cdot y)(b-a)\ dy = \int_0^1 h(y)\ dy$$

2.2.2 Opción 2: Método de las variables aleatorias

Consideramos la función de densidad $f(x) \sim U(a,b)$, entonces

$$I=\int_a^b g(x)\,dx=\int_a^b \frac{g(x)}{f(x)}\cdot f(x)\,dx=\int_a^b (b-a)\cdot g(x)\cdot\frac{1}{b-a}\,dx=(b-a)\int_a^b \cdot g(x)\cdot\frac{1}{b-a}\,dx=(b-a)E_X[g(X)],\qquad X\sim\mathcal{U}(a,b)$$

Ejemplo 2

Calcular $I=\int_0^2 e^{-x^2}dx$

Solución

La $I=\int_0^2 e^{-x^2}dx=\int_0^2 g_1(x)dx$, luego $g_1(x)=e^{-x^2}$

Opción 1

Hago el cambio de variable $y=\frac{x}{2}$, $dx=2dy$, con $y\in[0,1]$. La integral queda,

$$I=\int_0^2 e^{-x^2}dx=\int_0^1 e^{-(2y)^2}2\cdot dy=\int_0^1 2\cdot e^{-4y^2}dy,\quad y\in[0,1]$$

Llamamos, $g_2(Y)=e^{4Y^2}$, entonces

$$I=E[g_2(Y)]=\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^n g_2(Y_i),\quad Yi\sim\mathcal{U}(0,1)$$

por lo tanto,

$$\hat{I}=\frac{1}{n}\sum_{i=1}^n g_2(Y_i)=\frac{1}{n}\sum_{i=1}^n 2\cdot e^{4Y_i^2}$$

```
# Monte Carlo
n <- 1000                            # tamaño de la muestra
U <- runif(n, min = 0, max = 1)      # Generación  U ~U(0,1)
Y <- 2*exp(-4*U^2)
I.est <- mean(Y)
```

El valor estimado de la integral es $\hat{I}=0.8875133$.

Opción 2

$I=\int_0^2 e^{-x^2}dx$

Tomamos, $X\sim\mathcal{U}(0,2)$, $f(x)=\frac{1}{2}$, $\quad x\in\sim\mathcal{U}[0,2]$. Entonces,

$$S=\int_0^2 e^{-x^2}dx=2\int_0^2 e^{-x^2}\cdot\frac{1}{2}dx=2\cdot E[g(X)]$$

donde $g(X)=e^{-X^2}$

por lo tanto,

$$\hat{I}=2\cdot\frac{1}{n}\sum_{i=1}^n g(X_i)=2\cdot\frac{1}{n}\sum_{i=1}^n e^{-x_i^2},\quad X_i\sim\mathcal{U}(0,2)$$

Algoritmo

```
1.- Generar X1,...,Xn ~ U(0,2)
2.- Hacer   Yi = exp(- Xi^2), i = 1,...n
4.- Devolver I = 2*sum(Yi)/n
```

Ejemplo 3

Calcular $I = \int_0^1 (1-x^2)^{3/2}\ dx$

Esta integral tiene solución analítica.

Primero hacemos el cambio de variable, $x = sen(\theta) \Rightarrow\ dx = cos(\theta)d\theta$

y obtenemos,

$$I = \int (1-x^2)^{3/2}\ dx = \int (1-sen^2(\theta))^{3/2}\ d\theta = \int cos^2(\theta)^{3/2} cos(\theta)\ d(\theta) = \int cos^4(\theta)\ d\theta$$

Recordado que,

$$cos^2(\theta) = \frac{1+cos(2\theta)}{2} \qquad (1) cos^2(2\theta) = \frac{1+cos(4\theta)}{2} \qquad (2)$$

la integral puede escribirse como,

$$\int cos^4(\theta)\ d\theta = \int (\frac{1+cos(2\theta)}{2})^2\ d\theta + \int \frac{1}{4}\ d\theta + \int \frac{cos^2(2\theta)}{4}\ d\theta + \int \frac{cos(2\theta)}{2}\ d\theta$$

aplicando ahora la relación trigonométrica (2) al segundo término obtenemos,

$$\int cos^2(2\theta)\ d\theta = \int \frac{1+cos(4\theta)}{2}\ d\theta$$

Con lo que obtenemos,

$$\int cos^4(\theta)\ d\theta = \int \frac{1}{4}\ d\theta + \frac{1}{4}\int \frac{1+cos(4\theta)}{2}\ d\theta + \int \frac{1}{4}\ d\theta + \int \frac{cos(2\theta)}{2}\ d\theta$$

por último como

$$\int cos(a\theta) d\theta = \frac{sen(a\theta)}{a}$$

Obtenemos que,

$$I = \int (1-x^2)^{3/2}\ dx = \frac{3}{8}\frac{\pi}{2} + \frac{1}{32} sen(2\pi) = \frac{3}{16}\pi \approx 0.589048$$

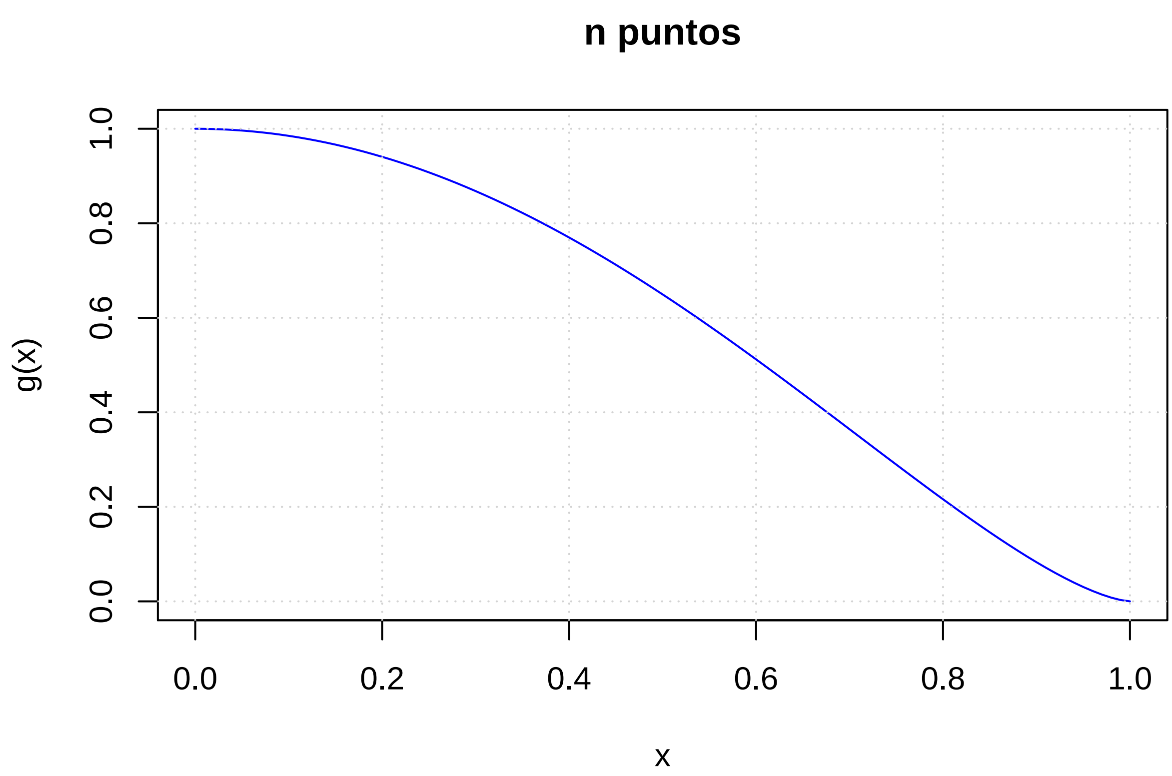
Mediante el método de integración Monte Carlo,

```r
# Generar Nsim valores de una U(0,1)
Nsim <- 10^6
n    <- runif(Nsim)

# función a integra g(x) = (1 - x^2)^(3/2)

g <- Vectorize(function(x) (1 - x^2)^(3/2))

# Gráfico de g(x)
curve(g, 0, 1, col = "blue", ylim = c(0,1), lwd = 1, main = "n puntos")
grid()
```




```
y <- g(n)

I.estimado <- mean(y)
IC_095     <- c(I.estimado - qnorm(0.025, lower.tail = FALSE)*sd(y)/sqrt(Nsim), I.estimado + qnorm(0.025

cat(' Valor estimado  ', I.estimado, ' I.C0.95% ', IC_095, '\n')
```

```
##  Valor estimado   0.5889608  I.C0.95%  0.5883101 0.5896116
```

```
# Valor de la integral

integrate(g, 0,1)
```

```
## 0.5890486 with absolute error < 1.3e-05
```

Ejemplo 4

a.- $\int_{-1}^{1} e^{x+x^2} 4\ dx$

Aquí,

$$\left.\begin{array}{rcr} a & = & -1 \\ b & = & 1 \end{array}\right\} \Rightarrow y = \frac{x-a}{b-a} = \frac{x+1}{2} \Rightarrow dy = \frac{1}{2}dx \Rightarrow \left\{\begin{array}{lcl} x=-1 & \rightarrow & y=0 \\ x=1 & \rightarrow & y=1 \end{array}\right. \quad \text{además } x = 2y-1.$$

La integral queda,

$$\int_{-1}^{1} e^{x+x^2} 4\ dx = \int_{-1}^{1} e^{(2y-1)+(2y-1)^2} 4{\cdot}2\ dy = 8\int_{0}^{1} e^{4Y^2-2y}\ dy = E_{\mathcal{U}}[e^{4Y^2-2y}] = E[g(Y)] \quad \text{donde } Y \sim \mathcal{U}(0,1)$$

Algoritmo

```
1. Generar U1,..., Un ~ U(0,1)
2. Hacer   Y = exp(4*U^2 - 2*U)
3. Hacer   I = 8*mean(Y)
4. Devolver I
```

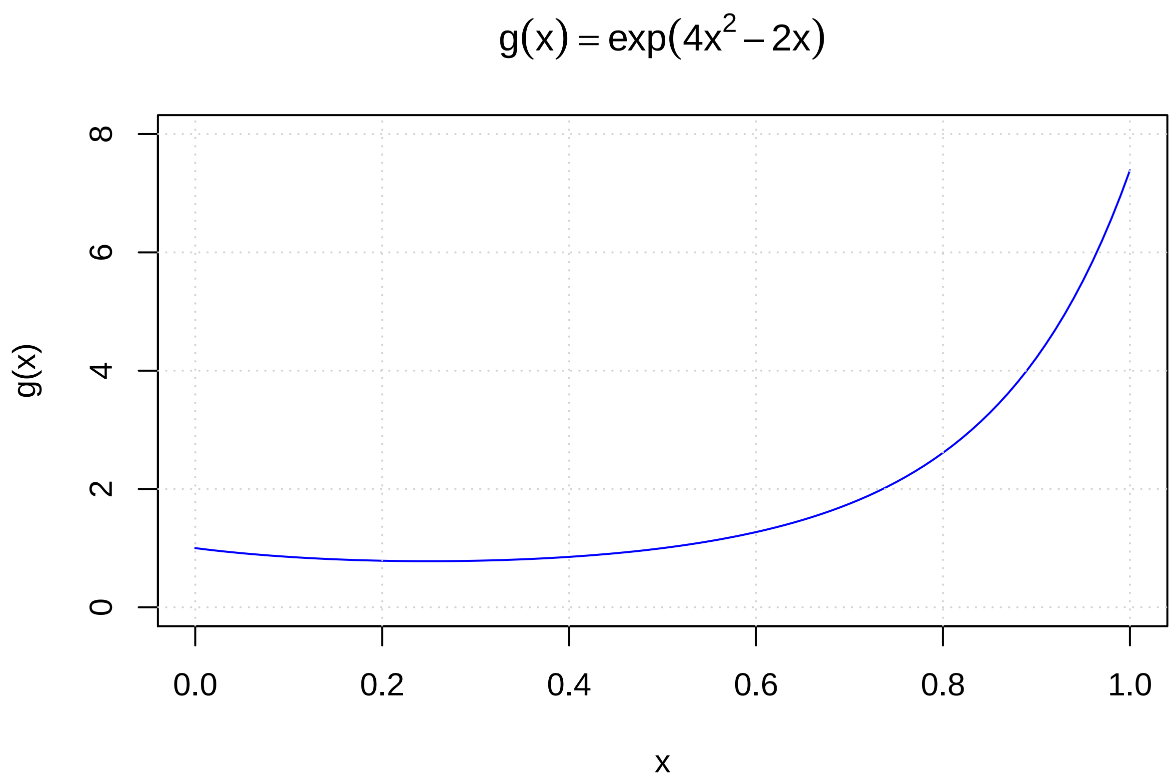
En R,

```r
# Semilla de aleatorización
set.seed(1714)

Nsim <- 10^6    # Tamaño de la muestra

# Integrando g(x) = exp(4x^2 - 2x)
g <- function(x) exp(4*x^2 - 2*x)

# Gráfico  de g(X)
curve(g, 0, 1, col = "blue", ylim = c(0,8), lwd = 1, main = expression(g(x) == exp(4*x^2 - 2*x)))
grid()
```




```r
# Estimación Monte Carlo
u <- runif(Nsim)
y <- g(u)
```

```
I.estimado <- 8*mean(y)
IC_095     <- c(I.estimado - qnorm(0.025, lower.tail = FALSE)*sd(y)/sqrt(Nsim), I.estimado + qnorm(0.025

cat('Valor estimado  ', I.estimado, 'I.CO.95% ', IC_095, '\n')
```

```
## Valor estimado   14.36015 I.CO.95%  14.35716 14.36314
```

```
# Valor de la integral

8*integrate(g, 0,1)$value
```

```
## [1] 14.35517
```

b.- $\int_0^{2\pi} sen(x)\ dx$

c.- $\int_{\pi}^{3\pi} cos(x)\ dx$

2.3 Integración sobre $(0, \infty)$

Calcular,

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

Reducimos esta integral al primer caso mediante un cambio de variable,

Sea,

$$y = \frac{1}{x+1} \qquad \Rightarrow dy = -\frac{1}{(1+x)^2}\ dx = -y^2\ dx$$

obtenemos,

$$S = \int_0^{\infty} g(x)dx = \int_0^1 \frac{g(\frac{1}{y}-1)}{y^2}\ dy = \int_0^1 h(y)\ dy$$

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

1.- $g(x) = e^{-x}$

2.- $g(x) = \frac{1}{2+x^2}$

3.- $g(x) = \frac{x}{(1+x^2)^2}$

3 Convergencia del método de Monte Carlo

En general el método de Monte Carlo converge lentamente y los métodos de análisis numérico clasicos tiene mejor eficiencia. Sin embargo, cuando se trata de integrales múltiples este método es competitivo e incluso mejor.

Veamos un algoritmo para estimar un intervalo de confianza del 95% de la esperanza de la función $g(X)$, donde $X \sim \mathcal{U}(0, 1)$.

Algoritmo

```
Inicialización
    Var(i)  = Varianza acumulada hasta la iteración i
    I(i)    = Media acumulada hasta la iteración i
Paso 1:
    Var(1)  = 0
    I0      = 0
    Generar U1 ~ U(0,1)
    Hacer   I(1) = g(U1)
Paso 2: Para i = 2 hasta n
    Generar Ui ~ U(0,1)
Paso 3:
    I(i+1)   = I(i) + (g(U(i+1) - I(i))/(i+1)
    Var(i+1) = (1 - 1/i) Var(i) + (1 + i) ( I(i+1) - I(i))^2
Paso 4: El intervalo es
    [I(n) - 1.96*sqrt(Var(n))/sqrt(n), I(n) + 1.96*sqrt(Var(n))/sqrt(n) ]
```

4 Integrales Multiples

Supongamos que se quiere calcular

$$I = \int_0^1 \int_0^1 \cdots \int_0^1 g(u_1, u_2, \ldots, u_n) du_1 du_2 \ldots du_n$$

que se puede expresar como

$$I = E[g(u_1, u_2, \ldots, u_n)]$$

donde $U_1, U_2, \ldots, U_n$ son variables aleatorias e independientes cuya distribución es una $\mathcal{U}(0,1)$.

Tomamos m conjuntos independientes, cada uno con n variables independientes distribuidas según la $\mathcal{U}(0,1)$,

$$\begin{array}{cccc} U_1^1 & U_2^1 & \cdots & U_n^1 \\ U_1^2 & U_2^2 & \cdots & U_n^2 \\ \cdots & \cdots & \cdots & \cdots \\ U_1^m & U_2^m & \cdots & U_n^m \end{array}$$

luego,

$$\begin{array}{cccc} g(U_1^1) & g(U_2^1) & \cdots & g(U_n^1) \\ g(U_1^2) & g(U_2^2) & \cdots & g(U_n^2) \\ \cdots & \cdots & \cdots & \cdots \\ g(U_1^m) & g(U_2^m) & \cdots & g(U_n^m) \end{array}$$

son independientes e identicamente distrinbuidas con media I.

Un estimador de I es,

$$\hat{I}_n = \frac{1}{n} \sum_{i=1}^{n} g(U_1^i, U_2^i, \ldots, U_n^i)$$

por lo tanto, el algoritmo es el mismo que en el caso unidimensional.

Ejemplo 5 Estima el valor de

$$\int_0^1 \int_0^1 e^{(x+y)^2} dx dy$$

y da un intervalo de confianza al 95%. ($I = 4.89916$)

Solución Hay que estimar,

$$\int_0^1 \int_0^1 e^{(x+y)^2} dxdy = \int_0^1 \int_0^1 g((x,y))\ dxdy$$

para ello generamos n puntos $(x_i, y_i) \in \mathcal{U}_{([0,1]\times[0,1])}$ y consideramos el estimador,

$$\hat{I} = \frac{1}{n}\sum_{i=1}^{n} g((x_i, y_i))$$

Algoritmo

```
1. Generar n puntos (U1,U2) ~ U([0,1]x[0,1])
2. Hacer   Y = g(U1,U2)
3. Hacer   I = media(Y)
4. Devolver I
```

En R,

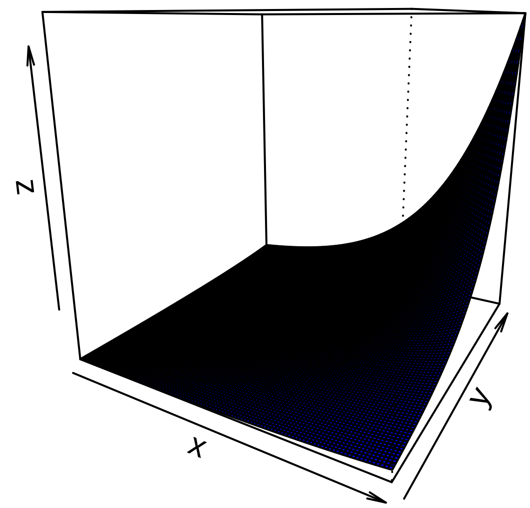
```r
# Semilla de  aleatorización
set.seed(1898)
Nsim <- 10^6    # Tamaño de la muestra

# Integrando
g <- function(x) {exp((x[1] + x[2])^2)}

#  Grafico de la función g(x) = exp((x^2+y^2)^2)

x <- seq(0,1,length=100)
y <- seq(0,1,length=100)

z <- outer(x, y, function(x,y) exp((x+y)^2))
persp(x,y,z, col = 'blue2', theta = 30, phi = 15)
```




```r
# Estimación Monte Carlo
u <- matrix(runif(2*Nsim), ncol = 2)
y <- apply(u,1,g)

I.estimado <- mean(y)
IC_095     <- c(I.estimado - qnorm(0.025, lower.tail = FALSE)*sd(y)/sqrt(Nsim), I.estimado + qnorm(0.025

cat('Valor estimado  ', I.estimado, 'I.C0.95% ', IC_095, '\n')
```

```
## Valor estimado   4.897694 I.C0.95%  4.886034 4.909355
```

Ejemplo 6 Estima el valor de

$$\int_0^1 \cdots \int_0^1 2^{-7} \left(\sum_{i=1}^{8} x_i \right)^2 dx_1 dx_2 \cdots dx_8$$

y da un intervao de confianza al 95%. ($I = 0.130208$)

Solución

Generamos n puntos del hipercubo $[0,1] \times \ldots \times [0,1]$ y consideramos el estimador,

$$\hat{I} = \frac{1}{n} \sum_{i=1}^{n} g((x_i, y_i))$$

Algoritmo

```
1. Generar n puntos (U1,U2,...,U8) ~ U([0,1]x...x[0,1])
2. Hacer   Y = g(U1,U2,...,U8)
3. Hacer   I = media(Y)
4. Devolver I
```

En R,

```r
# Semilla de  aleatorización
set.seed(1898)
Nsim <- 10^6    # Tamaño de la muestra

# Integrando
g <- function(x) { (sum(x))^2/2^7}


# Estimación Monte Carlo
u <- matrix(runif(8*Nsim), ncol = 8)
y <- apply(u,1,g)

I.estimado <- mean(y)
IC_095     <- c(I.estimado - qnorm(0.025, lower.tail = FALSE)*sd(y)/sqrt(Nsim), I.estimado + qnorm(0.025

cat('Valor estimado  ', I.estimado, 'I.C0.95% ', IC_095, '\n')
```

```
## Valor estimado   0.1301713 I.C0.95%  0.1300704 0.1302722
```