UNIVERSIDAD DE BOGOTÁ JORGE TADEO LOZANO

Departamento de Ciencias Básicas y Modelado

SIMULACION ESTOCASTICA

Tema 2

El método de Monte Carlo: Integración de Monte Carlo

Profesor: Javier Riascos Ochoa, MSc, PhD

(javier.riascos@utadeo.edu.co)

Monte Carlo

Monte Carlo, Casino

- El término Monte Carlo es usado para referirse a toda clase de técnicas que involucran la simulación computacional a través del uso de generadores de números aleatorios.

- Alude a los juegos de azar en los casinos de Monte Carlo, Mónaco.

Monte Carlo

- La idea básica es repetir un experimento numérico *muchas veces* para obtener cantidades de interés usando la Ley de los Grandes Números y otros métodos.

- Actualmente, los métodos de Monte Carlo son una herramienta esencial en muchas investigaciones cuantitativas en ciencias, ingeniería, finanzas, etc.

- Aplicaciones básicas en: Muestreo, Estimación, Optimización, Integración numérica.

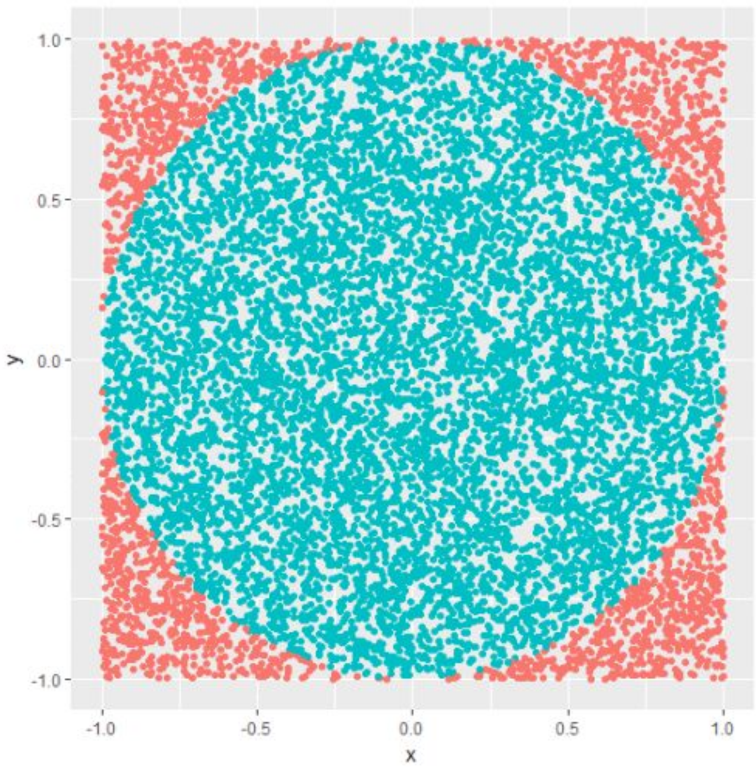
Stanislaw Ulam


John Von Neumann

Monte Carlo

- Actualmente, los métodos de Monte Carlo son una herramienta esencial en muchas investigaciones cuantitativas en ciencias, ingeniería, finanzas, etc.

- Aplicaciones básicas en: Muestreo, Estimación, Optimización, Integración numérica.



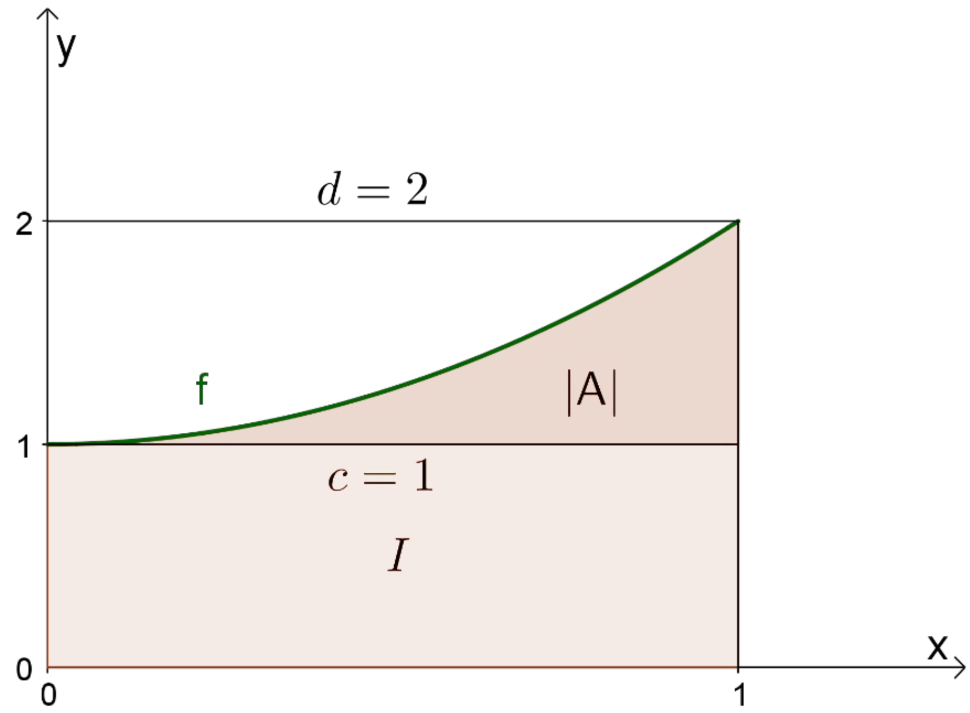
Método de Hit-and-miss de integración

Queremos computar la integral:

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

Si $f(x) \in [c, d]$, entonces:

$$I = |A| + c(b - a)$$



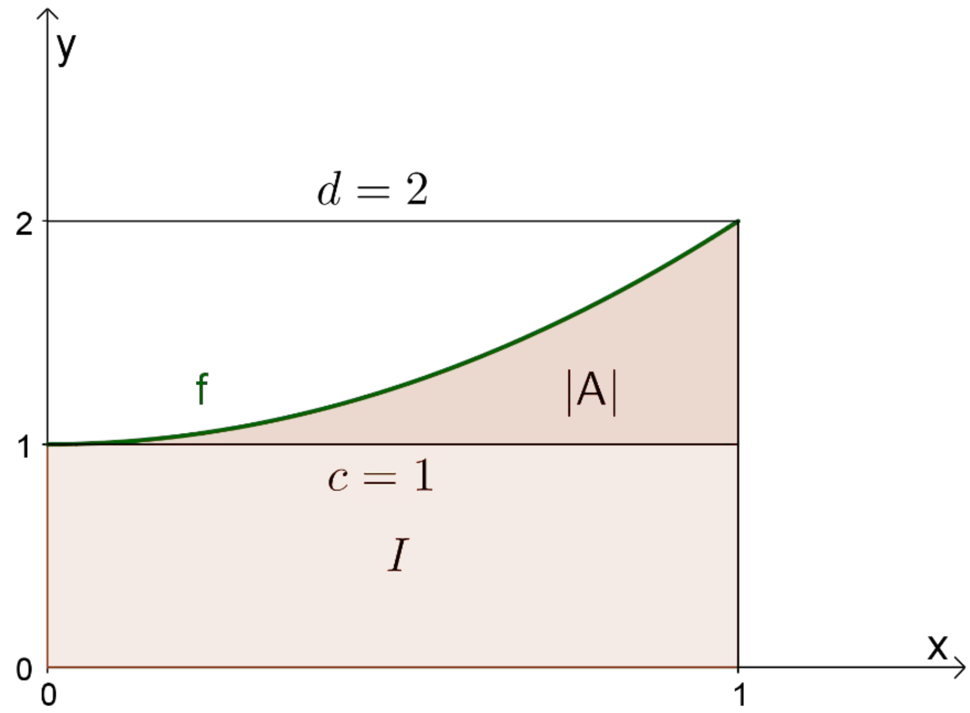
Método de Hit-and-miss de integración

Queremos computar la integral:

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

Si $f(x) \in [c, d]$, entonces:

$$I = |A| + c(b - a)$$



Generamos variables aleatorias X, Y

$$X \sim U(a, b) \qquad Y \sim U(c, d)$$

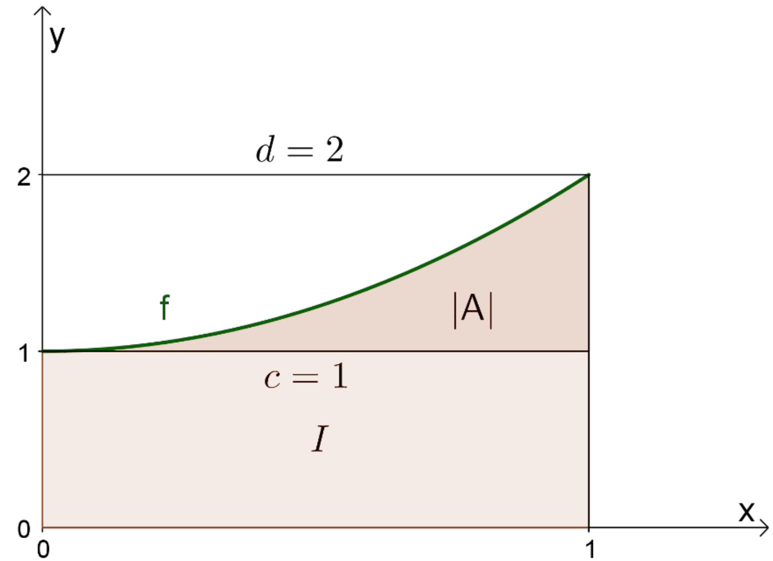
Entonces (X, Y) está uniformemente distribuido sobre la caja $[a, b] \times [c, d]$

Luego: $|A| = P\big((X, Y) \in A\big) * (b - a)(d - c)$

Método de Hit-and-miss de integración

Y si definimos: $Z = \begin{cases} 1 & si \quad (X,Y) \in A \\ 0 & si \quad (X,Y) \notin A \end{cases}$

entonces: $P\big((X,Y) \in A\big) = E[Z]$



La idea es generar un número suficientemente grande *n* de variables *iid*

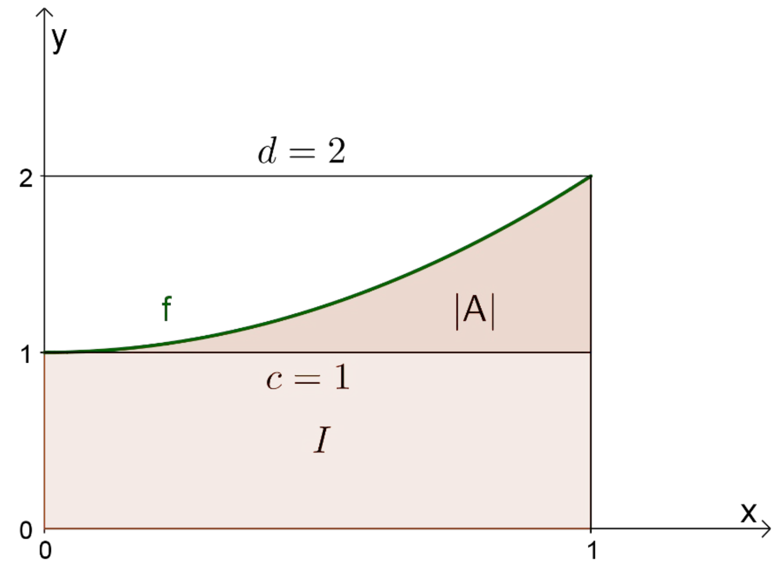
$$X_i \sim U(a,b)$$
$$Y_i \sim U(c,d)$$

De las cuales se obtiene Z_i.

Método de Hit-and-miss de integración

Y si definimos: $Z = \begin{cases} 1 & si \quad (X,Y) \in A \\ 0 & si \quad (X,Y) \notin A \end{cases}$

entonces: $P\big((X,Y) \in A\big) = E[Z]$



La idea es generar un número suficientemente grande *n* de variables *iid*

$$X_i \sim U(a,b)$$
$$Y_i \sim U(c,d)$$

De las cuales se obtiene Z_i.

El promedio de estos números Z_i se aproximará a su valor esperado, para un número *n* suficientemente grande:

$$\sum_{i=1}^{n} \frac{Z_i}{n} \to E[Z] \quad para \quad n \to \infty$$

Ley fuerte de los grandes números

Del cual se obtendrá el estimado de $|A|$ y de ahí de la integral *I*.

Método de Hit-and-miss de integración

Programa:

Este programa realiza integración por el método de Hit-and-miss de la función ftn sobre el intervalo [a,b] y plotea las sucesivas aproximaciones a la integral en función de *n*

```
hit_miss2 <- function(ftn, a, b, c, d, n) {
 # Monte-Carlo integration using the hit & miss method
 # partially vectorised version
 X <- runif(n, a, b)
 Y <- runif(n, c, d)
 Z <- (Y <= sapply(X, ftn))
 I <- (b - a)*c + (cumsum(Z)/(1:n))*(b - a)*(d - c)
 plot(1:n, I, type = "l")
 return(I[n])
}
```

Ejecución:

```
> source('hit_miss2.r')
> f <- function(x) x^3 - 7*x^2 + 1
> hit_miss2(f, 0, 1, -6, 2, 10000)
[1] -1.052
> lines(c(1, 10000), c(-13/12, -13/12))
```

Método de Hit-and-miss de integración

Programa: *Hit_miss2(modified).R*

Versión modificada del código: imprime todos los valores de *I* obtenidos de cada iteración. Plotea en color rojo el valor real.

Método mejorado: Integración de Monte Carlo

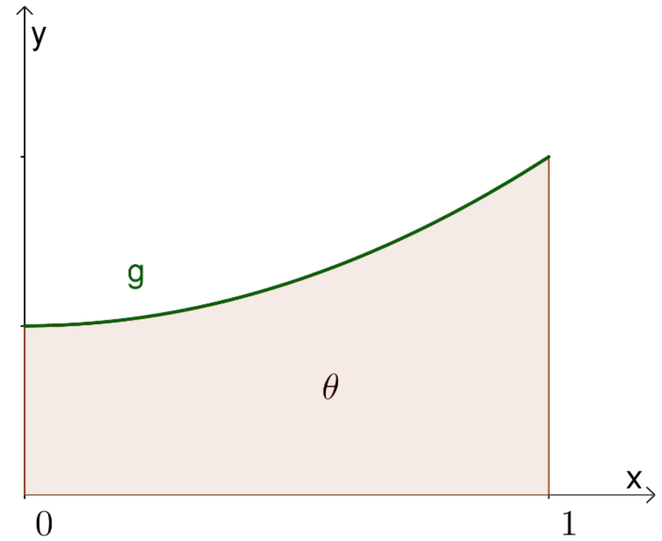
Queremos computar θ donde:

$$\theta = \int_0^1 g(x)dx$$

Sea $X \sim U(0,1)$ entonces:

$$\theta = E[g(X)]$$

(Demostración en el tablero)



Método mejorado: Integración de Monte Carlo

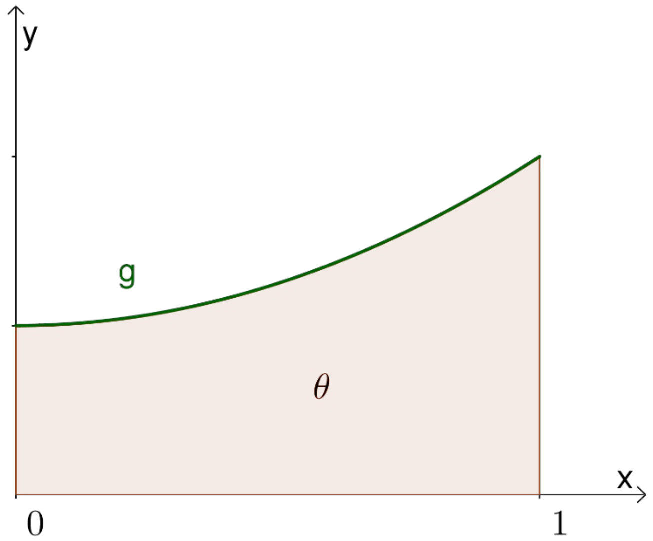
Queremos computar θ donde:

$$\theta = \int_0^1 g(x)dx$$

Sea $X \sim U(0,1)$ entonces:

$$\theta = E[g(X)]$$

(*Demostración en el tablero*)



Si generamos *n* v.a.'s *iid:* $X_i \sim U(0,1)$, las v.a.'s $g(X_i)$ también son *iid*, con valor esperado θ.

Luego,

$$\sum_{i=1}^{n} \frac{g(X_i)}{n} \rightarrow E[g(X)] = \theta \quad para \quad n \rightarrow \infty$$

El promedio de $g(Xi)$ se aproximará a su valor esperado que es el valor de la integral

Método mejorado: Integración de Monte Carlo

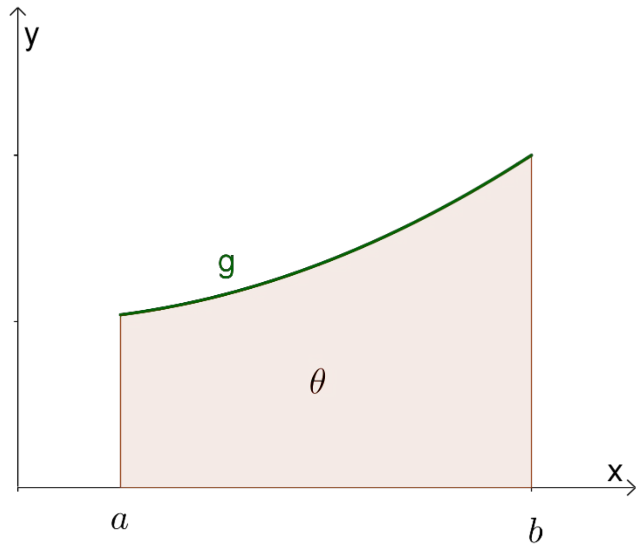
Ahora: Si queremos computar θ donde:

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

Definimos $X \sim U(a, b)$ entonces:

$$\theta = E[g(X)] \times (b - a)$$

Valor esperado de la altura — Ancho



Método mejorado: Integración de Monte Carlo

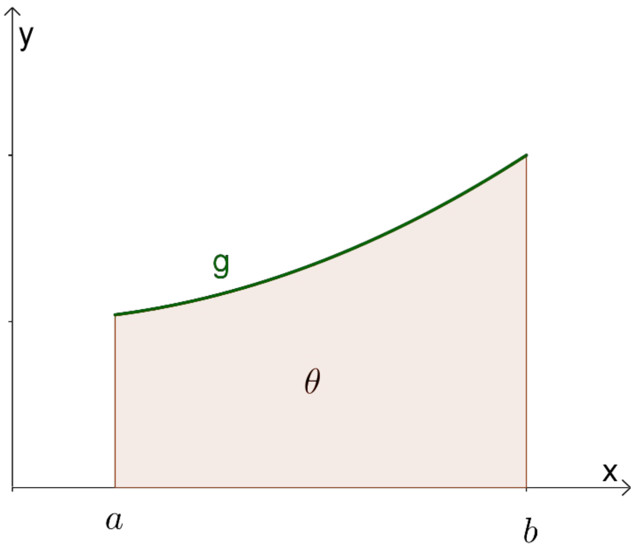
Ahora: Si queremos computar θ donde:

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

Definimos $X \sim U(a,b)$ entonces:

$$\theta = E[g(X)] \times (b-a)$$

Valor esperado de la altura   Ancho



Si generamos *n* v.a.'s *iid:* $X_i \sim U(a,b)$, las v.a.'s $g(X_i)$ también son *iid*, con valor esperado:

$$\frac{\theta}{b-a}.$$

Luego,

$$\boxed{\sum_{i=1}^{n} \frac{g(X_i)}{n} \rightarrow E[g(X)] = \frac{\theta}{b-a} \quad para \quad n \rightarrow \infty}$$

Método mejorado: Integración de Monte Carlo

Programa: *mc_integral.R*

Este programa realiza integración de Monte Carlo de la función *ftn* sobre el intervalo [a,b].

```
mc_integral <- function(ftn, a, b, n) {
 # Monte Carlo integral of ftn over [a, b] using a sample of size n
 u <- runif(n, a, b)
 x <- sapply(u, ftn)
 return(mean(x)*(b-a))
}
```

Ejercicio:

Plotee las sucesivas aproximaciones a la integral, en función de *n.*

Método mejorado: Integración de Monte Carlo

Ahora: Si queremos computar θ donde:

$$\theta = \int_0^\infty g(x)dx$$

Hacemos el cambio de variables

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

Entonces, con $h(y) = \frac{g\left(\frac{1}{y}-1\right)}{y^2}$:

$$\theta = \int_0^1 h(y)dy$$

(*Demostración en el tablero*)

Luego, si generamos n v.a.'s *iid:* $Y_i \sim U(0,1)$, las v.a.'s $h(Y_i)$ también son *iid*, con valor esperado:

$$\frac{\theta}{1-0} = \theta$$

Integración de Monte Carlo en dimensiones superiores

El método de Monte Carlo es más adecuado para evaluar integrales en dimensiones superiores *d*:

$$\theta = \int_0^1 \int_0^1 \cdots \int_0^1 g(x_1, x_2, \cdots, x_d) dx_1\, dx_2 \cdots dx_d$$

Entonces $\theta = E[g(X_1, X_2, \dots, X_d)]$
donde $X_1, X_2, \dots, X_d$ son v.a's independientes distribuidas $U(0,1)$.

Entonces, si generamos n conjuntos independientes de v.a's *iid* para cada una de las *d* variables X_i con distribución $U(0,1)$:

$$\begin{array}{c} X_1^1, X_2^1, \dots, X_d^1 \\ X_1^2, X_2^2, \dots, X_d^2 \\ \dots \\ X_1^n, X_2^n, \dots, X_d^n \end{array}$$

Se aproxima θ a partir del promedio:

$$\theta \sim \sum_{i=1}^{n} \frac{g(X_1^i, X_2^i, \dots, X_d^i)}{n}$$

Ejercicio: Plantee el caso

$$\theta = \int_{a_d}^{b_d} \cdots \int_{a_1}^{b_1} g(x_1, x_2, \cdots, x_d) dx_1\, dx_2 \cdots dx_d$$

Precisión de la Integración por Monte Carlo

Notación O grande

Que el error de un método numérico sea $O(f(n))$, con n el número de llamadas de función en el programa, significa que para n suficientemente grande, el **error** es a lo más $k \cdot f(n)$, para alguna constante k.

Precisión de MC integration en más dimensiones

Sea *d* la dimensión y *n* el número de llamadas de función en el programa. La precisión de diferentes métodos de integración es:

Method	Error
Trapezoid	$O(n^{-2/d})$
Simpson's rule	$O(n^{-4/d})$
Hit-and-miss Monte Carlo	$O(n^{-1/2})$
Improved Monte Carlo	$O(n^{-1/2})$

El error del Método de Monte Carlo no depende de d. Se aconseja utilizarlo cuando:

$$\boldsymbol{d > 8}$$