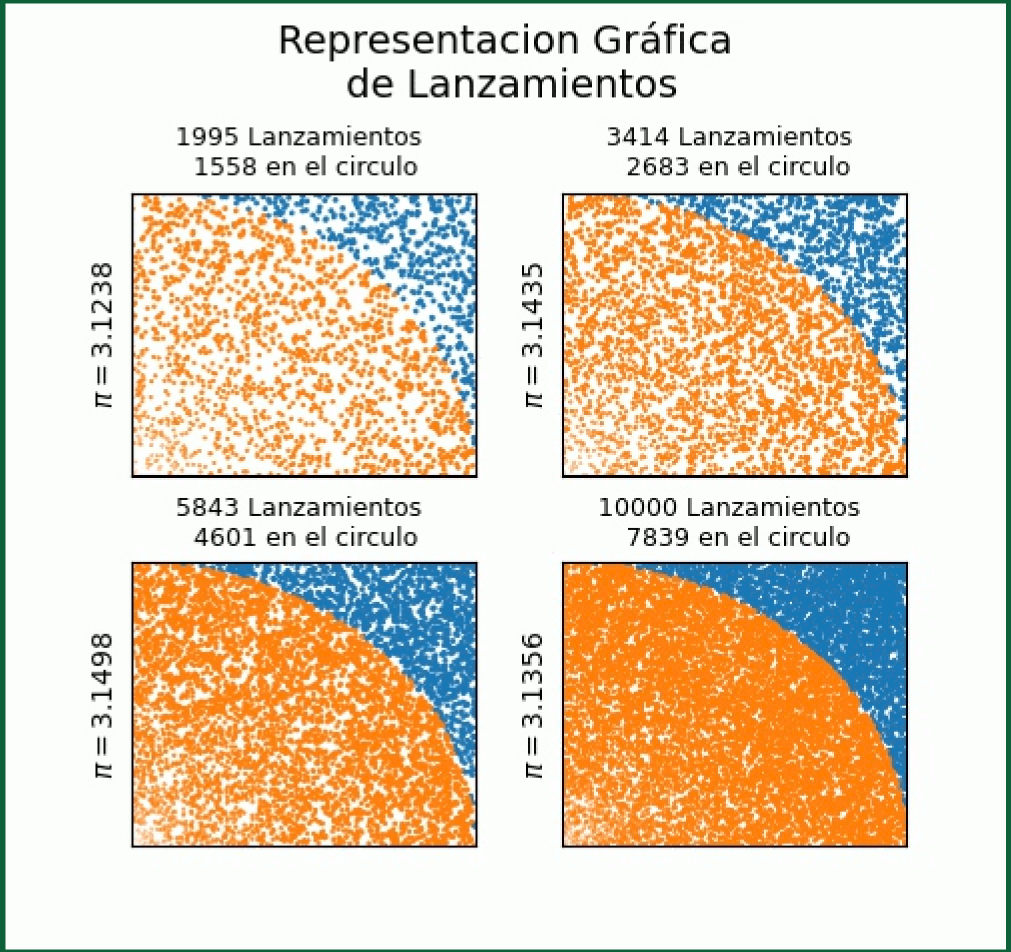
El método de Montecarlo

El método de Montecarlo es una técnica matemática utilizada para aproximar soluciones numéricas a problemas complejos mediante el uso de muestreo aleatorio y probabilidad. Su nombre proviene del famoso casino de Monte Carlo, debido al enorme papel del azar en este método.



Cómo funciona

El método de Montecarlo utiliza simulaciones repetitivas basadas en números aleatorios para aproximar soluciones a problemas complejos. Se suele emplear en situaciones donde es difícil o imposible obtener una solución analítica exacta. Su procedimiento básico sigue estos pasos:

1. Definir el problema y el dominio de posibles valores.
2. Generar valores aleatorios dentro del dominio.
3. Evaluar cada valor en la función del problema.
4. Calcular la estimación basada en los resultados obtenidos.

Características Principales

1. Se basa en repeticiones aleatorias (simulaciones) para estimar resultados.
2. Es útil cuando el problema es demasiado complicado para resolverse analíticamente.
3. Se aplica en áreas como física, finanzas, ingeniería, inteligencia artificial y mas.

Ejemplo clásico (estimar π (pi).

El objetivo es estimar pi usando aleatoriedad.

$$\boxed{\pi \approx 4\left(\frac{puntos\ dentro\ del\ circulo}{total\ de\ puntos}\right)}$$

Algunos pasos:

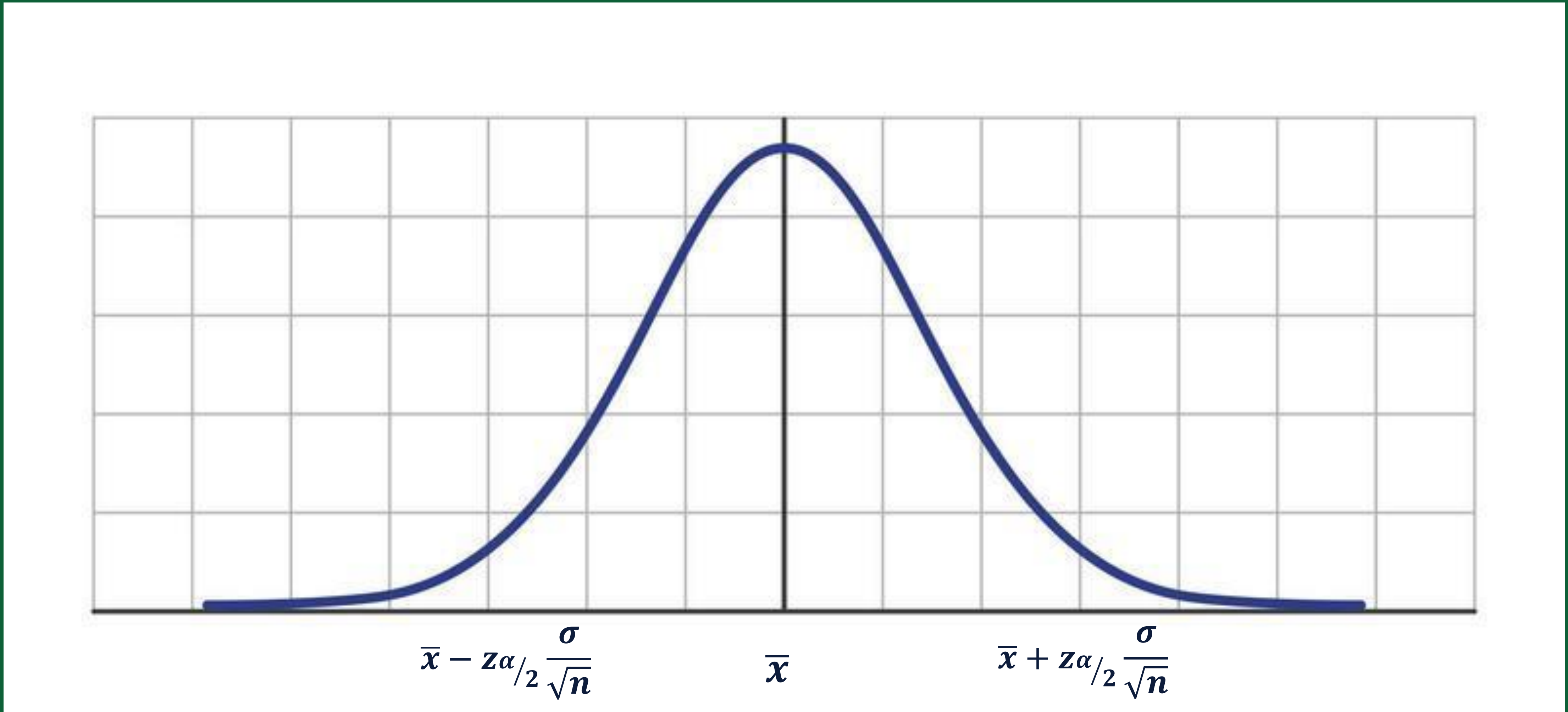
1. Dibuja un cuadrado de lado 2 y un circulo inscrito de radio 1 (área del circulo = π).

2. Genera puntos aleatorios dentro del cuadrado.

3. Cuenta los puntos que caen dentro del circulo (puntos de éxito), la distancia al centro <= 1

```
import random

# Número de puntos aleatorios
num_puntos = 1000000
puntos_dentro = 0  # Contador de puntos dentro del círculo

# Generamos los puntos aleatorios
for _ in range(num_puntos):
    x = random.uniform(-1, 1)  # Coordenada x aleatoria en [-1,1]
    y = random.uniform(-1, 1)  # Coordenada y aleatoria en [-1,1]

    # Si el punto cae dentro del círculo unitario, sumamos al contador
    if x**2 + y**2 <= 1:
        puntos_dentro += 1

# Estimación de pi usando la relación entre área del círculo y cuadrado
pi_estimado = (puntos_dentro / num_puntos) * 4

print(f"Estimación de π usando Montecarlo: {pi_estimado}")
```

Fundamentos Estadísticos



Fundamentos Estadísticos

$$\sigma = \sqrt{\frac{1}{(n-1)}(f(x_i) - \bar{x})^2}$$

$\sigma = desviación\ estándar$

$$\hat{I} = \int_a^b f(x)dx = (b-a)\frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

$$EE = \frac{\sigma}{\sqrt{n}}$$

$EE = Errror\ estándar$

$$\hat{I} \pm z_{\alpha/2}\frac{\sigma}{\sqrt{n}}$$

$\hat{I} = Integral\ estimada$

$$\left[\hat{I} - z_{\alpha/2}\frac{\sigma}{\sqrt{n}}, \hat{I} + z_{\alpha/2}\frac{\sigma}{\sqrt{n}}\right]$$

$Intervalo\ de\ confianza$

IC	$z_{\alpha/2}$
90%	1.645
95%	1.96
99%	2.576.

Métodos para generar números con distribución uniforme
Los generadores más comunes en programación incluyen:

- en Python: ranom.uniform(a,b) Devuelve un número aleatorio en el intervalo ([a, b]).

```
import random

# Generar 5 números aleatorios uniformes en el rango [0, 1]
for _ in range(10):
    print(random.uniform(0, 1))
```

Primera ejecución	Segunda ejecución
0.5415297812161269	0.15314364228942168
0.08528259751738321	0.0006898010800105991
0.978464626756584	0.3174500651784191
0.6283191030955314	0.3762803645039583
0.6194501801034179	0.48383798739750516

- en Python: ranom.seed(42) Devuelve un número aleatorio en el intervalo ([a, b]). Con semilla en 42 y siempre va devolver el mismo resultado

```
import random

random.seed(42)  # Establece la semilla para la generación de números aleatorios

# Generar 10 números aleatorios uniformes en el rango [0, 1]
for _ in range(5):
    print(random.uniform(0, 1))
```

Primera ejecución

0.6394267984578837
0.025010755222666936
0.27502931836911926
0.22321073814882275
0.7364712141640124

Segunda ejecución

0.6394267984578837
0.025010755222666936
0.27502931836911926
0.22321073814882275
0.7364712141640124

- Aplicación (calculo de área)

Método Integración Montecarlo

La media ($\bar{x}$) de las muestras se utiliza como estimador del valor de la integral porque bajo ciertas condiciones converge al valor esperado de la función que estamos integrando. Esto se basa en la ley de los grandes números (LGN) y en propiedades estadísticas de los estimadores.

1. Fundamento matemático: La integral como valor esperado $I = \int f(x)\,dx$, sobre un dominio multidimensional.

Si adaptamos el problema para muestrea x uniformemente, podemos expresar la integral como un valor esperado.

$$I = Volumen(\quad).E|f(x)|$$

$E|f(x)|$ es el valor esperado de $f(x)$ cuando x se distribuye uniformemente

Estimación Montecarlo por promedio muestral tomando n= muestral $x_1, x_2, \ldots, x_n$
Uniformemente distribuidos.

Calculamos el promedio de las evaluaciones de f en estas muestras

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

$\hat{I}$= Es la estimación de la integral

$$\boxed{\hat{I} = Volumen(\quad)\frac{1}{n}\sum_{i=1}^{n} f(x_i)}$$

Convergencia por la ley de los grandes números

Esta ley establece que si las muestras son independientes e idénticamente distribuidas, entonces:

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} f(x_i) = E|f(x)|$$

$\hat{I}$= Es la estimación de la integral converge al valor verdadero I cuando $n \to \infty$

Casi 1: integral de una variable 1D, en el intervalo $[a, b]$:

$$I = \int_a^b f(x)dx = (b-a)\frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

Integral con dominio no estandarizado

$$I = \int_a^b f(x)dx = (b-a)\frac{1}{n}\sum_{i=1}^{n} f(x_i) \qquad \begin{cases} volumen\ (b-a) \neq 1 \\ Muestra\ x_i \sim uniforme\ (a, b) \end{cases} \qquad \begin{matrix} x_i = a + (b-a)u_i \\ u_i \in (0,1) \end{matrix}$$

$\hat{I}$= Es la estimación de la integral converge al valor verdadero I cuando $n \to \infty$

Caso 2: Integrales múltiples:

$$I = \int_a^b \int_c^d f(x, y)dydx \qquad \hat{I} = (b-a)(c-d)\frac{1}{n}\sum_{i=1}^{n} f(x_i, y_i) \quad \begin{cases} volumen\ (b-a)(d-c)\ area\ del\ rectangulo \\ Muestra\ x_i \sim uniforme\ en\ [a, b].[c, d] \end{cases}$$

❖ Cálculos de Áreas en integrales

Características	Montecarlo Típico	Montecarlo para área bajo la curva
Muestreo	En 1D $\rightarrow x_i$	En 2D (x_i, y_i)
Dominio	Intervalo $[a, b]$	Rectángulo contenedor D
Calculo	Promedio de $f(x_i)$	Proporción de puntos bajo la curva
Función usada	$f(x)$ directamente	Función indicadora $y \leq f(x)$
Área/volumen	Longitud (b-a)	Área del rectángulo
Aplicación típica	Integrales de funciones generales	Áreas geométricas o integrales con fronteras visuales