

# EST-24107: Simulación

**Profesor:** Alfredo Garbuno Iñigo — Otoño, 2023 — Integración Monte Carlo.

**Objetivo.** Estudiar integración numérica en un contexto probabilístico. Estudiar, en particular, el método Monte Carlo y entender sus bondades y limitaciones como un método de aproximación de integrales.

**Lectura recomendada:** Una lectura mas técnica sobre reglas de cuadratura se puede encontrar en la sección 3.1 de Reich and Cotter [1]. Y una buena referencia (técnica) sobre el método Monte Carlo lo encuentran en Sanz-Alonso et al. [3]. En el curso estamos explorando mas allá de lo que ofrece el capítulo 3 de Robert and Casella [2].

## 1. Introducción

En muchas aplicaciones nos interesa poder resolver integrales de manera numérica. Estas pueden ser de cualquier forma. Por ejemplo, nos puede interesar resolver

$$\int_{\Theta} h(\theta) d\theta, \quad (1)$$

que bien puede ser reexpresada como una integral bajo una medida de probabilidad.

Por ejemplo,

$$\int_{\Theta} f(\theta) \pi(\theta) d\theta, \quad (2)$$

de tal forma que podemos pensar en la ecuación de arriba como un valor esperado de una variable  $\theta \sim \pi(\cdot)$  y calcular mediante un método numérico.

- La pregunta clave (I) es: ¿qué distribución podemos utilizar?
- La pregunta clave (II) es: ¿con qué método numérico resuelvo la integral?
- La pregunta clave (III) es: ¿y si no hay método numérico?

## 2. Integración numérica

Recordemos la definición de integrales Reimann:

$$\int f(x) dx \approx \sum_{n=1}^N f(u_n) \Delta u_n =: \hat{\pi}_N^R(f).$$

La aproximación utilizando una malla (cuadrícula) de  $N$  puntos sería:

$$\sum_{n=1}^N f(u_n) \Delta u_n.$$

El método es útil cuando las integrales se realizan cuando tenemos pocos parámetros. Es decir, cuando el dominio de integración es  $\mathcal{X} \subseteq \mathbb{R}^p$  con  $p$  pequeña.

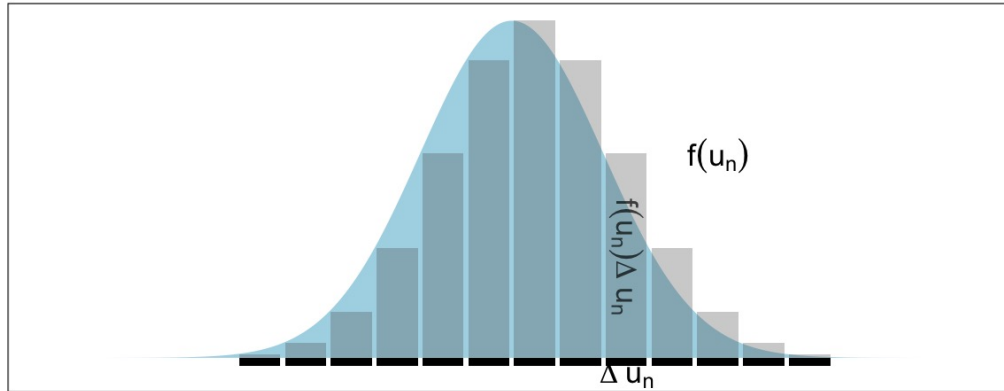


FIGURA 1. Integral por medio de discretización con  $N = 11$ .

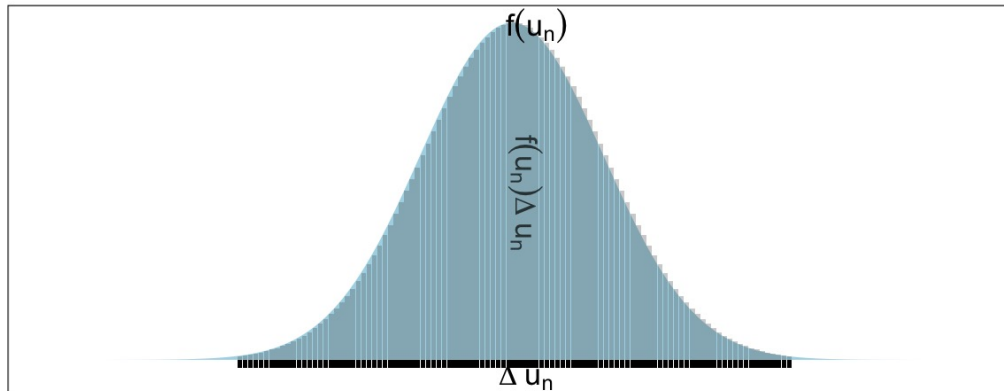


FIGURA 2. Integral por medio de una malla fina,  $N = 101$ .

### 2.1. Análisis de error

El concepto de **integrabilidad de Darboux** nos puede ayudar a acotar el error cometido por nuestra estrategia de integración. Por ejemplo, para una partición  $\rho_N$  (malla) del intervalo tenemos que

$$L_{f,\rho_N} \leq \hat{\pi}_N^R(f) \leq U_{f,\rho_N} . \quad (3)$$

Lo que recordarán de sus cursos de cálculo es que

$$\lim_{N \rightarrow \infty} |U_{f,\rho_N} - L_{f,\rho_N}| = 0 , \quad (4)$$

y que además se satisface

$$\int f(x) dx = \lim_{N \rightarrow \infty} U_{f,\rho_N} = \lim_{N \rightarrow \infty} L_{f,\rho_N} . \quad (5)$$

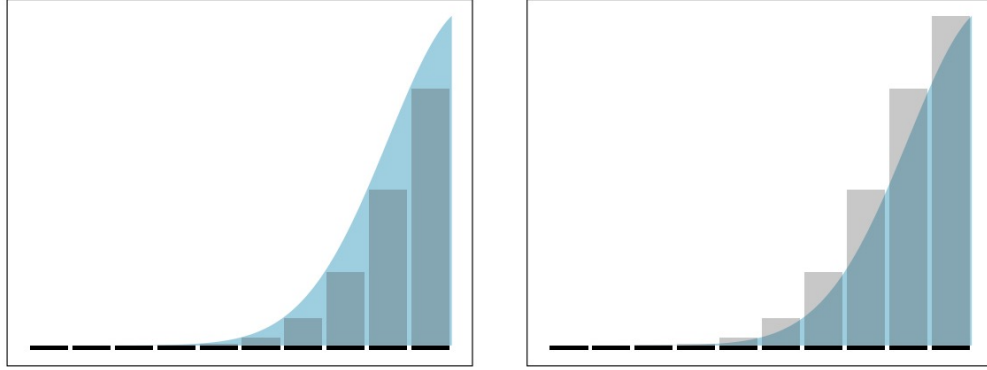


FIGURA 3. Integrales y cotas de Darboux.

## 2.2. Más de un parámetro

Consideramos ahora un espacio con  $\theta \in \mathbb{R}^p$ . Si conservamos  $N$  puntos por cada dimensión, ¿cuántos puntos en la malla necesitaríamos? Lo que tenemos son recursos computacionales limitados y hay que buscar hacer el mejor uso de ellos. En el ejemplo, hay zonas donde no habrá contribución en la integral.

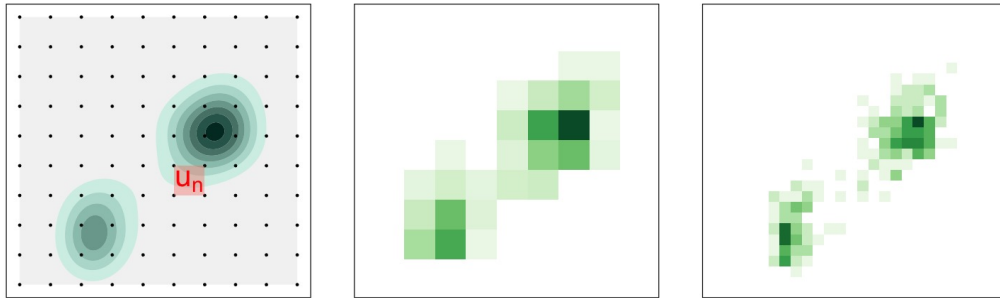


FIGURA 4. Integral multivariada por método de malla.

## 2.3. Reglas de cuadratura

Por el momento hemos escogido aproximar las integrales por medio de una aproximación con una malla uniforme. Sin embargo, se pueden utilizar aproximaciones

$$\int f(x) dx \approx \sum_{n=1}^N f(\xi_n) \omega_n.$$

Estas aproximaciones usualmente se realizan para integrales en intervalos cerrados  $[a, b]$ . La regla de cuadratura determina los pesos  $\omega_n$  y los centros  $\xi_n$  pues se escogen de acuerdo a ciertos criterios de convergencia.

Por ejemplo, se consideran polinomios que aproximen con cierto grado de precisión el integrando. Los pesos y los centros se escogen de acuerdo a la familia de polinomios. Pues para cada familia se tienen identificadas las mallas que optimizan la aproximación. Ver sección 3.1 de Reich and Cotter [1].

### 3. Integración Monte Carlo

$$\pi(f) = \mathbb{E}_\pi[f] = \int f(x)\pi(x)dx,$$

$$\pi_N^{\text{MC}}(f) = \frac{1}{N} \sum_{n=1}^N f(x^{(n)}), \quad \text{donde } x^{(n)} \stackrel{\text{iid}}{\sim} \pi, \quad \text{con } n = 1, \dots, N,$$

$$\pi_N^{\text{MC}}(f) \approx \pi(f).$$

#### 3.1. Ejemplo: Dardos

Consideremos el experimento de lanzar dardos uniformemente en un cuadrado de tamaño 2, el cual contiene un círculo de radio 1.

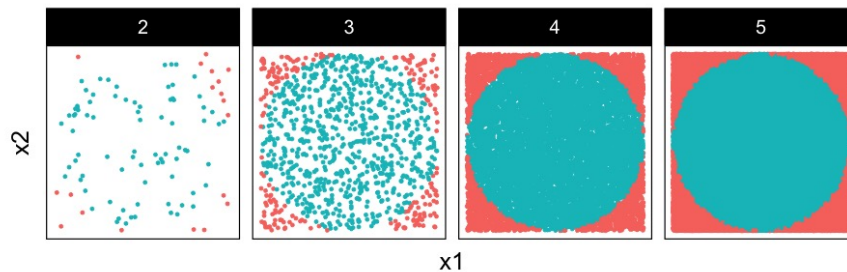


FIGURA 5. Integración Monte Carlo para aproximar  $\pi$ .

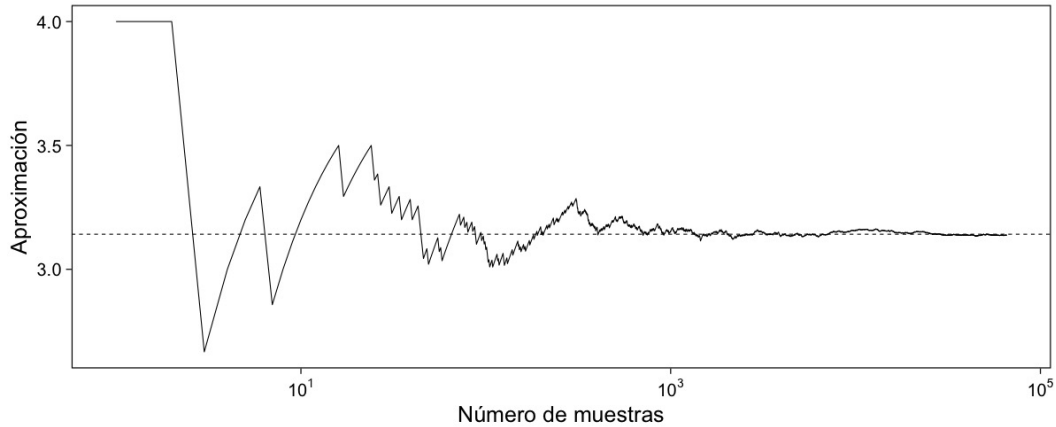
Si escogemos  $N$  suficientemente grande entonces nuestro promedio converge a la integral. En Figura 6 se muestra para cada  $n$  en el eje horizontal cómo cambia nuestra estimación  $\hat{\pi}_n^{\text{MC}}(f)$ .

También podemos replicar el experimento unas  $M$  veces y observar cómo cambiaría nuestra estimación con distintas semillas. Por ejemplo, podemos replicar el experimento 100 veces. En R y python lo usual es utilizar arreglos multidimensionales para poder guardar muestras bajo distintas replicas.

```
1 set.seed(108)
2 nsamples <- 10**4; nexp <- 100
3 U <- runif(nexp * 2 * nsamples)
4 U <- array(U, dim = c(nexp, 2, nsamples))
```

```
1 str(U)
```

```
1 num [1:100, 1:2, 1:10000] 0.455 0.404 0.351 0.664 0.463 ...
```

FIGURA 6. Estimación  $\pi_N^{MC}(f)$  con  $N \rightarrow \infty$ .

Hay que tener cuidado como accedamos a un arreglo multidimensional. Por ejemplo, ¿cómo podemos acceder las primeras 5 réplicas de lanzar 10,000 dardos al plano?

```
1 U[1:5] > str()
```

```
1 num [1:5] 0.455 0.404 0.351 0.664 0.463
```

El código anterior no funcionó. El que usamos es el siguiente:

```
1 U[1:5,,] > str()
```

```
1 num [1:5, 1:2, 1:10000] 0.455 0.404 0.351 0.664 0.463 ...
```

Una pregunta natural es: ¿cómo aplicamos un resumen (función) a cada experimento? Para esto, podemos usar la función `apply` que ya hemos usado anteriormente.

```
1 apply(U[1:5,,], 1, str)
```

```
1 num [1:2, 1:10000] 0.455 0.164 0.529 0.415 0.474 ...
2 num [1:2, 1:10000] 0.404 0.9282 0.0883 0.3126 0.4386 ...
3 num [1:2, 1:10000] 0.351 0.449 0.369 0.814 0.695 ...
4 num [1:2, 1:10000] 0.664 0.627 0.185 0.882 0.113 ...
5 num [1:2, 1:10000] 0.4635 0.0115 0.0117 0.5086 0.9384 ...
6 NULL
```

Vamos a usar una función para encapsular el resumen:

```
1 cuenta_dardos <- function(x){
2   ## obtiene la norma 2
3   dardos <- apply(x**2, 2, sum)
4   ## cuenta si los dardos estan en el circulo unitario
5   exitos <- ifelse(dardos <= 1, 1, 0)
6   ## obtiene frecuencias relativas
```

### 3 Integración Monte Carlo

```
7   prop  <- mean(exitos)
8   4 * prop
9 }
```

```
1 resultados <- apply(U, 1, cuenta_dardos)
2 resultados
```

```
1 [1] 3.152 3.139 3.163 3.140 3.151 3.147 3.146 3.139 3.163 3.129
2 [11] 3.146 3.118 3.148 3.116 3.134 3.127 3.161 3.128 3.139 3.136
3 [21] 3.132 3.124 3.140 3.159 3.142 3.158 3.201 3.118 3.140 3.152
4 [31] 3.175 3.133 3.132 3.142 3.135 3.122 3.132 3.126 3.145 3.162
5 [41] 3.148 3.117 3.173 3.150 3.154 3.146 3.102 3.197 3.140 3.144
6 [51] 3.132 3.144 3.124 3.128 3.156 3.159 3.124 3.136 3.141 3.148
7 [61] 3.146 3.136 3.153 3.114 3.123 3.136 3.143 3.128 3.169 3.124
8 [71] 3.145 3.130 3.164 3.188 3.111 3.132 3.154 3.139 3.112 3.122
9 [81] 3.143 3.170 3.170 3.152 3.155 3.114 3.124 3.120 3.164 3.146
10 [91] 3.147 3.136 3.163 3.146 3.164 3.153 3.131 3.156 3.134 3.158
```

Nota cómo podemos obtener la distribución de distintos promedios usando un histograma de los resultados.

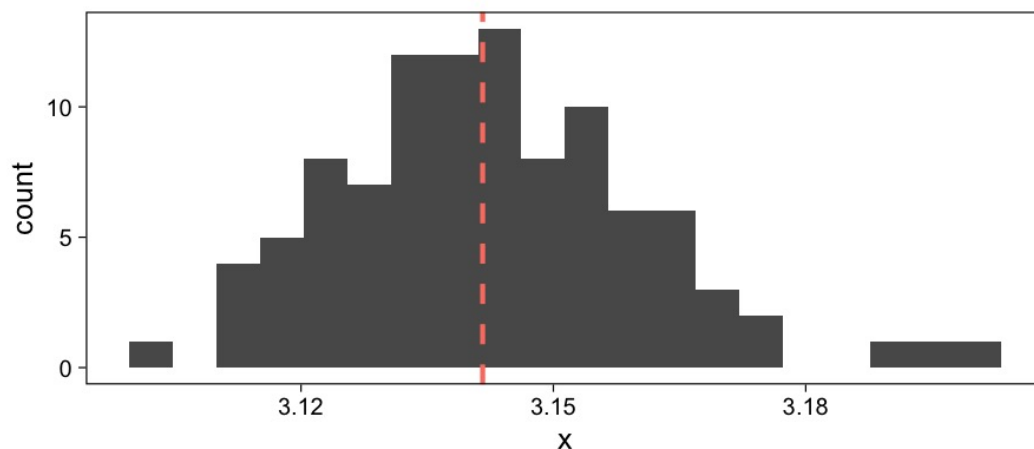


FIGURA 7. Distribución de posibles promedios bajo distintas semillas.

También es usual utilizar un gráfico de **trazas** o **trayectorias** para ver comportamientos interesantes. Por ejemplo, podemos usar la función `cuenta_acumulacion_dardos` como se muestra a continuación.

```
1 cuenta_acumulacion_dardos <- function(x){
2   ## obtiene la norma 2
3   dardos <- apply(x**2, 2, sum)
4   ## cuenta si los dardos estan en el circulo unitario
5   exitos <- ifelse(dardos <= 1, 1, 0)
6   ## obtiene frecuencias relativas mientras avanza N
7   prop  <- cummean(exitos)
8   4 * prop
9 }
```

```

1 resultados <- apply(U, 1, cuenta_acumulacion_dardos)
2 resultados > str()
3 ## Nota como transpone el resultado

```

```

1 num [1:10000, 1:100] 4 4 4 4 4 4 4 4 4 4 ...

```

Lo cual nos permite realizar distintos escenarios posibles.

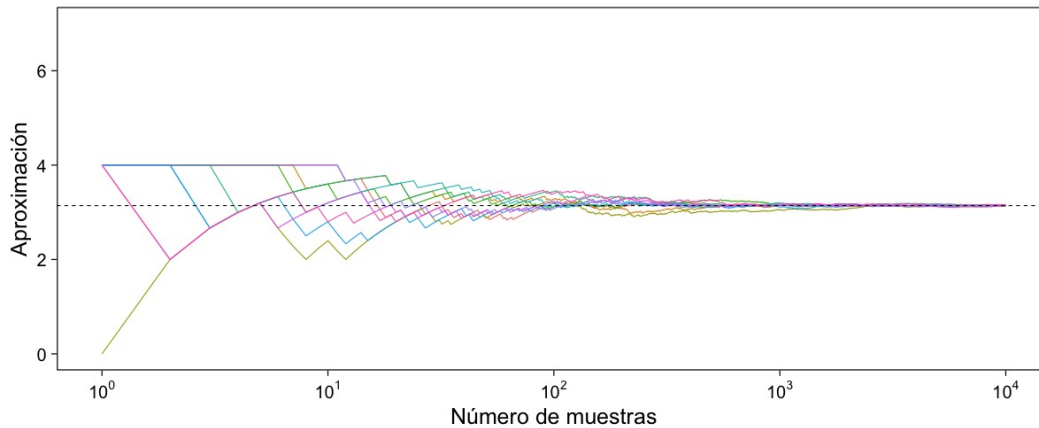


FIGURA 8. Réplica de las trayectorias de diversas realizaciones de la aproximación de la integral.

Bajo ciertas consideraciones teóricas podemos esperar un buen comportamiento de nuestro estimador de la integral. E incluso podríamos (si el número de simulaciones lo permite) aproximar dicho comportamiento utilizando distribuciones asintóticas, (TLC).

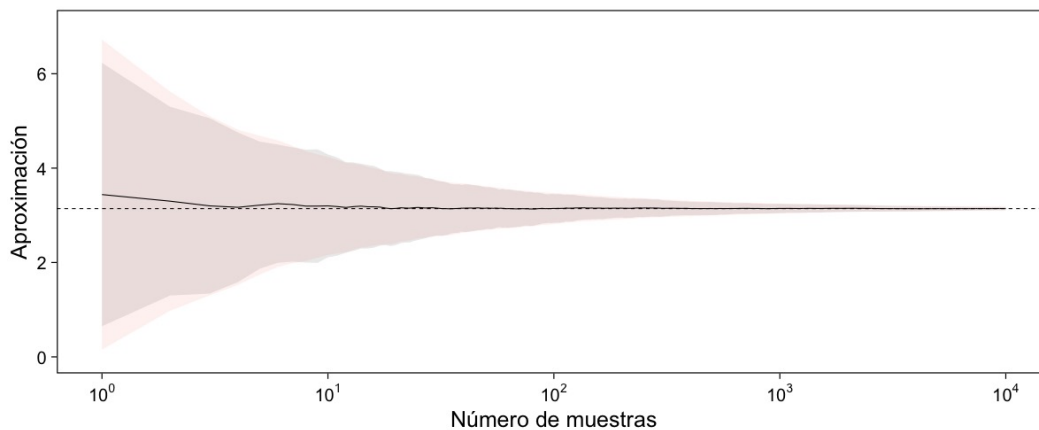


FIGURA 9. Comportamiento promedio e intervalos de confianza.

Podemos explicar la reducción de los intervalos de confianza por medio de la varianza de la estimación de la integral en las distintas réplicas que tenemos. Mas adelante explicaremos de dónde viene la línea punteada. Vemos cómo, aunque captura bien la reducción en varianza, puede sub- o sobre-estimarla.

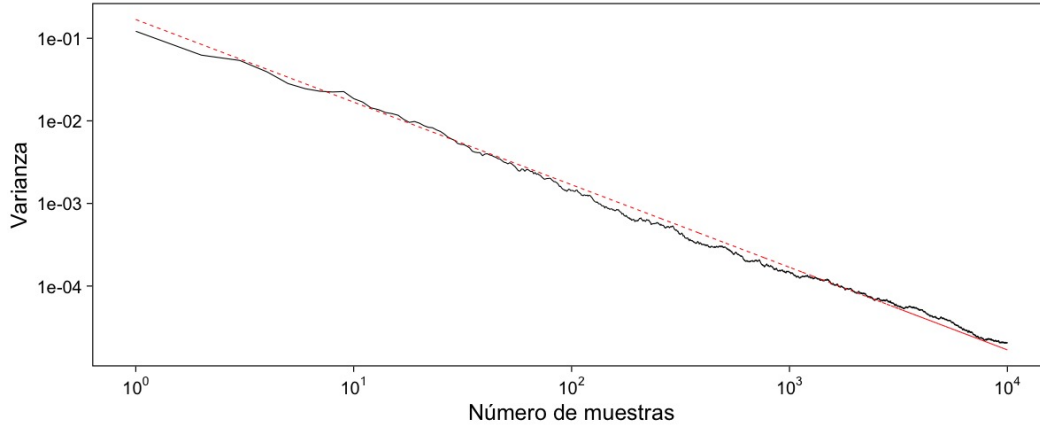


FIGURA 10. Comportamiento promedio e intervalos de confianza.

### 3.2. Propiedades

A continuación enunciaremos algunas propiedades clave del método Monte Carlo. Poco a poco las iremos explicando y en particular discutiremos mas a fondo algunas de ellas.

**Teorema 3.1** (Error Monte Carlo). Sea  $f : \mathbb{R}^p \rightarrow \mathbb{R}$  cualquier función bien comportada<sup>†</sup>. Entonces, el estimador Monte Carlo es **insesgado**. Es decir, se satisface

$$\mathbb{E} \left[ \hat{\pi}_N^{\text{MC}}(f) - \pi(f) \right] = 0, \quad (6)$$

para cualquier  $N$ .

Usualmente estudiamos el error en un escenario pesimista donde medimos el **error cuadrático medio** en el peor escenario.

**Proposición 3.2** (Máximo error Monte Carlo posible). Sea  $\mathcal{F}$  una familia de funciones bien comportadas. Entonces, el error de aproximación de dicha familia está dado por

$$\sup_{f \in \mathcal{F}} \mathbb{E} \left[ \left( \hat{\pi}_N^{\text{MC}}(f) - \pi(f) \right)^2 \right] \leq \frac{1}{N}.$$

Esta desigualdad nos muestra una de las propiedades que usualmente se celebran de los métodos Monte Carlo. La integral y nuestra aproximación de ella por medio de simulaciones tiene un error acotado proporcionalmente por el número de simulaciones.

**Proposición 3.3** (Error Monte Carlo). La varianza del estimador Monte Carlo (**error estándar**) satisface la igualdad

$$\text{ee}^2 \left( \hat{\pi}_N^{\text{MC}}(f) \right) = \frac{\mathbb{V}_{\pi}(f)}{N}.$$



Esta igualdad, aunque consistente con nuestra desigualdad anterior, nos dice algo mas. El error de nuestra aproximación **depende** de la varianza de  $f$  bajo la distribución  $\pi$ .

**Teorema 3.4** ( TLC para estimadores Monte Carlo). Sea  $f$  una función **bien comportada** <sup>††</sup>, entonces bajo una  $N$  suficientemente grande tenemos

$$\sqrt{N} \left( \hat{\pi}_N^{\text{MC}}(f) - \pi(f) \right) \sim \mathcal{N}(0, \mathbb{V}_\pi(f)) . \quad (7)$$

**Observación.** El estimador Monte Carlo del que hablamos,  $\hat{\pi}_N^{\text{MC}}(f)$ , es una estimación con una **muestra finita de simulaciones**. En ese sentido podemos pensar que tenemos un *mapeo* de muestras a estimador

$$(x^{(1)}, \dots, x^{(N)}) \mapsto \hat{\pi}_N^{\text{MC}}(f) , \quad (8)$$

con  $x^{(i)} \stackrel{\text{iid}}{\sim} \pi$  .

De lo cual es natural pensar: ¿y si hubiéramos observado otro conjunto de simulaciones? Nuestro proceso de estimación es el mismo pero la muestra puede cambiar.

En este sentido nos preguntamos por el **comportamiento promedio** bajo distintas muestras observadas (nota las variables aleatorias con respecto a lo que estamos tomando el valor esperado).

$$\mathbb{E}[\hat{\pi}_N^{\text{MC}}(f)] = \mathbb{E}_{x^{(1)}, \dots, x^{(N)}} [\hat{\pi}_N^{\text{MC}}(f)] . \quad (9)$$

De la misma manera nos podemos preguntar sobre la **dispersión alrededor de dicho promedio** (varianza)

$$\mathbb{V}[\hat{\pi}_N^{\text{MC}}(f)] = \mathbb{V}_{x^{(1)}, \dots, x^{(N)}} [\hat{\pi}_N^{\text{MC}}(f)] . \quad (10)$$

Al ser un ejercicio de **estimación** la desviación estándar del estimador recibe el nombre de **error estándar**. Lo cual denotamos por

$$\text{ee}[\hat{\pi}_N^{\text{MC}}(f)] = \left( \mathbb{V}[\hat{\pi}_N^{\text{MC}}(f)] \right)^{1/2} = \left( \frac{\mathbb{V}_\pi(f)}{N} \right)^{1/2} . \quad (11)$$

**Observación.** Para algunos estimadores la fórmula del error estándar se puede obtener de manera analítica (curso de **Inferencia Matemática**). Para otro tipo, tenemos que utilizar propiedades asintóticas (p.e. cota de Cramer-Rao).

Hay casos en los que no existe una fórmula asintótica o resultado analítico, pero podemos usar simulación [ 8 ] para cuantificar dicha dispersión (lo veremos en otra sección del curso).

**Observación.** Hay situaciones en las que la distribución normal asintótica no tiene sentido. Para este tipo de situaciones también veremos cómo podemos utilizar simulación para cuantificar dicha dispersión.

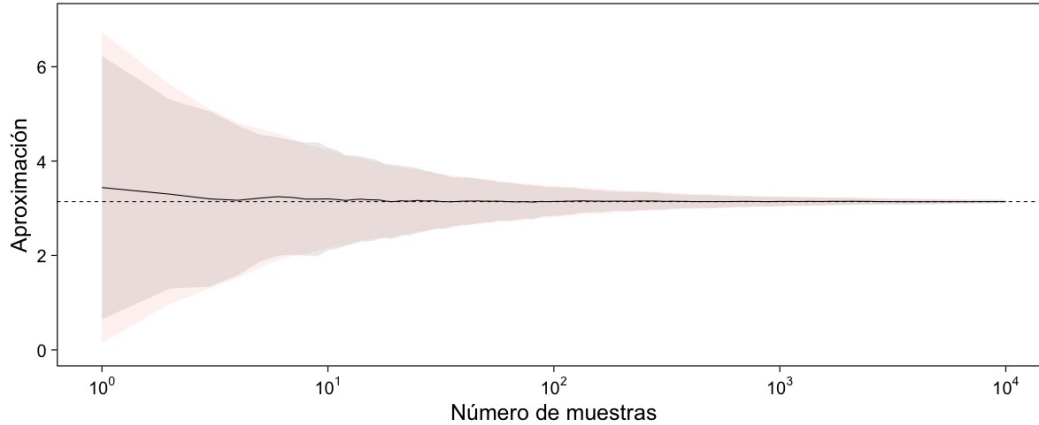


FIGURA 11. Comportamiento promedio e intervalos de confianza con aproximación asintótica.

### 3.3. Estimación de una proporción

El lanzamiento de dardos que vimos es un ejemplo de una situación muy usual en estimación de integrales. Queremos estimar la tasa de éxito a partir de ver el éxito o fracaso de experimentos Bernoulli.

Si denotamos por  $\theta$  la tasa de éxito. Entonces nuestro experimento (lanzar dados dentro del círculo) determina que  $S_n \sim \text{Binomial}(N, \theta)$  y que  $\bar{X}_n$  es un **estimador** de  $\theta$ . Por lo tanto,

$$\hat{\theta}_n := \bar{X}_n \approx \theta \quad (12)$$

**Pregunta.** ¿Cuál es la fórmula del error estándar para este estimador?

**Pregunta.** ¿Cuántas muestras necesitamos para tener una *buena* aproximación?

## 4. Desigualdades de concentración

En muchas situaciones nos interesa establecer cuántas simulaciones necesitamos para poder aproximar nuestras integrales hasta cierto orden. Por ejemplo, la tabla en Listado 1 muestra la aproximación conforme aumenta  $N$ .

```

1 # A tibble: 4 × 3
2   N estimado diff.abs
3   <int>      <dbl>    <dbl>
4 1    10         4      0.858
5 2   100        3.12   0.0216
6 3  1000        3.16   0.0224
7 4 10000        3.15   0.0100

```

LISTING 1. Aproximación de la proporción de dardos dentro de la diana.

### 4.1. Desigualdad de Chebyshev

Lo que queremos es encontrar una  $N$  tal que con una **alta probabilidad** nuestro **estimador** sea **cercano al parámetro** que se está ajustando. Esto lo escribimos como:

$$\text{Prob}(|\hat{\theta}_N - \theta| < \epsilon) \geq 1 - \delta. \quad (13)$$

**Teorema 4.1** (Desigualdad de Chebyshev). Sea  $X$  una variable aleatoria con media y varianza finita denotadas por  $\mu$  y  $\sigma^2$  respectivamente. Entonces para cualquier constante positiva  $k \in \mathbb{R}$ , tenemos que

$$\text{Prob}(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}. \quad (14)$$

Lo cual podemos utilizar para encontrar una cota inferior para  $N$ .

**Pregunta.** Calcula la desigualdad y obtén el número de simulaciones necesarias para encontrar un estimador con nivel de precisión  $\epsilon$  con una probabilidad  $\alpha$ .

#### 4.2. Desigualdad de Hoeffding

La desigualdad de Chebyshev nos permite encontrar cotas para prácticamente cualquier situación<sup>†</sup>. Sin embargo, el precio es la calidad de la estimación.

Una alternativa es utilizar la desigualdad de Hoeffding que nos permite establecer cotas para variables aleatorias acotadas en un intervalo.

Aunque la discusión es a nivel variable aleatoria, lo que estamos discutiendo es relevante en integración Monte Carlo. Pues, si  $f : X \rightarrow [a, b]$  podemos pensar en  $f(X)$  como una variable aleatoria acotada en  $[a, b]$  y nuestra discusión procede.

**Teorema 4.2** (Desigualdad de Hoeffding). Sea  $X_1, \dots, X_n$  una muestra iid de variables aleatorias con valores en  $[a, b]$ . Entonces para cualquier  $t \geq 0$  y usando  $S_n = X_1 + \dots + X_n$  tenemos que

$$\text{Prob}(|S_n - \mathbb{E}[S_n]| \geq t) \leq 2 \exp\left(-\frac{2t^2}{n(b-a)^2}\right). \quad (15)$$

**Pregunta.** Establece el tamaño de muestra necesario,  $N$ , para garantizar con probabilidad al menos  $1 - \delta$  que nuestro estimador será  $\epsilon$  preciso.

### 5. Justificación del método Monte Carlo

Lo que hemos discutido hasta ahora nos permite ver que el método Monte Carlo—aproximar integrales con promedios— tiene buenas propiedades. El tiro de gracia es el siguiente resultado.

**Teorema 5.1** (Ley fuerte de los grandes números). Sea  $X_1, \dots, X_n$  una muestra de variables iid y sea  $X$  una variable con la misma distribución. Si utilizamos una  $f : \mathbb{R} \rightarrow \mathbb{R}$  acotada, entonces  $h(X_1), \dots, h(X_n)$  son variables independientes y acotadas con media finita. De tal forma que se satisface que

$$\text{Prob}\left(\lim_{N \rightarrow \infty} \hat{\pi}_N^{\text{MC}}(h) = \pi(h)\right) = 1. \quad (16)$$

**Teorema 5.2** (Ley débil de los grandes números). De los resultados anteriores ya sabíamos

que

$$\lim_{N \rightarrow \infty} \text{Prob} \left( \left| \hat{\pi}_N^{\text{MC}}(h) - \pi(h) \right| < \epsilon \right) = 1. \quad (17)$$

## 6. Consideraciones

Supongamos que queremos resolver la integral

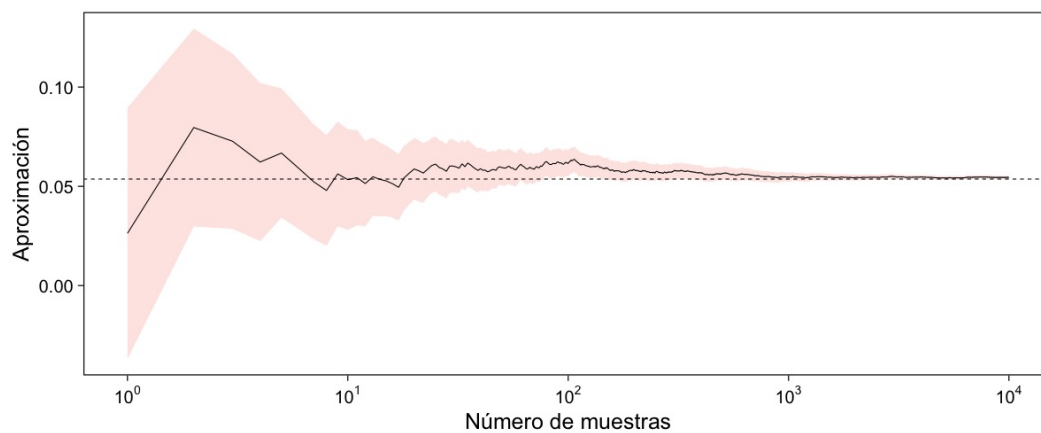
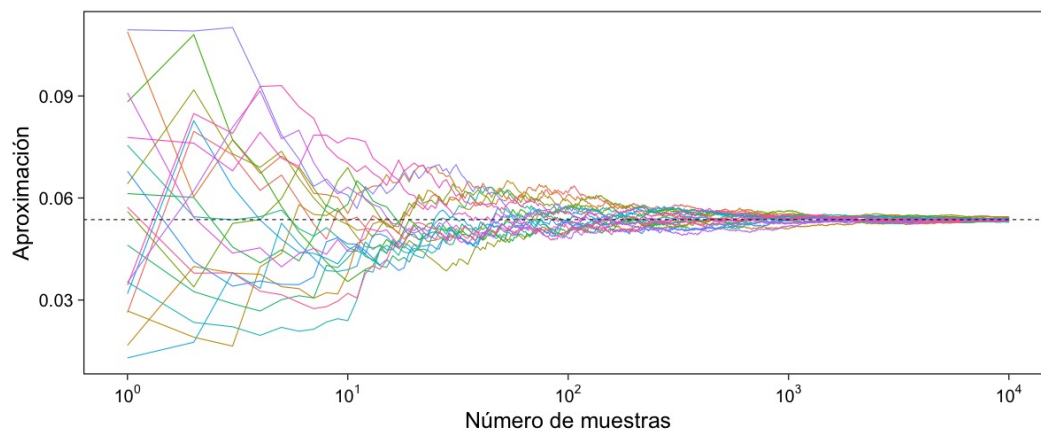
$$\int_2^3 e^{-\frac{x^2}{2}} dx. \quad (18)$$

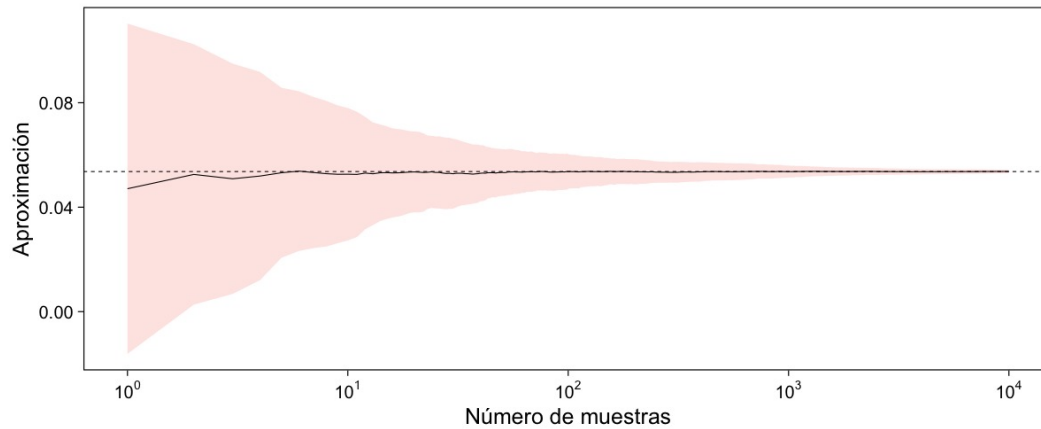
### 6.1. Primera estrategia

Al tener un intervalo acotado podemos pensar en una  $U(2, 3)$ .

```
1 set.seed(108); nsamples <- 10**4; nexp <- 100
2 h <- function(x){ exp(-x**2/2) }
3 u <- runif(nexp * nsamples, min = 2, max = 3)
4 x <- array(u, c(nexp, nsamples))
5 h_x <- h(x)
```

```
1 estimador_mc <- apply(h_x, 1, cummean)      # ojo, transpone el resultado
2 media_mc <- apply(estimador_mc, 1, mean)
3 error_mc <- apply(estimador_mc, 1, sd)
```



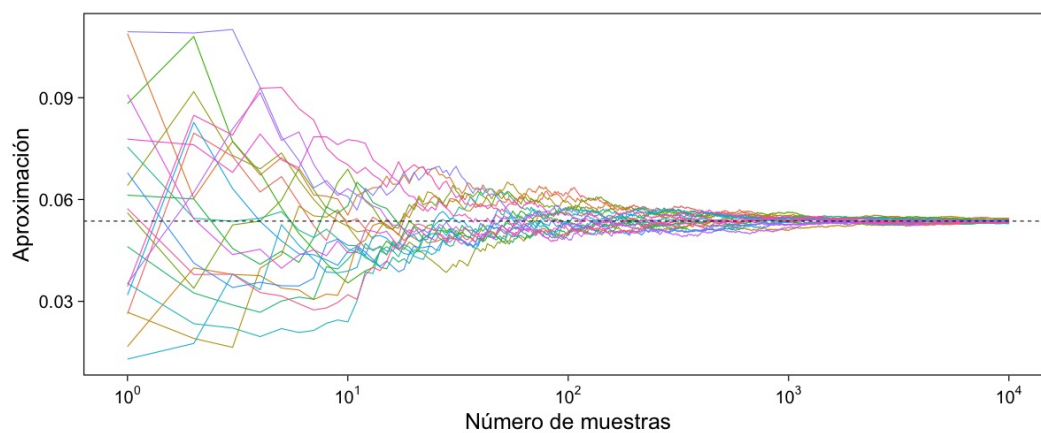


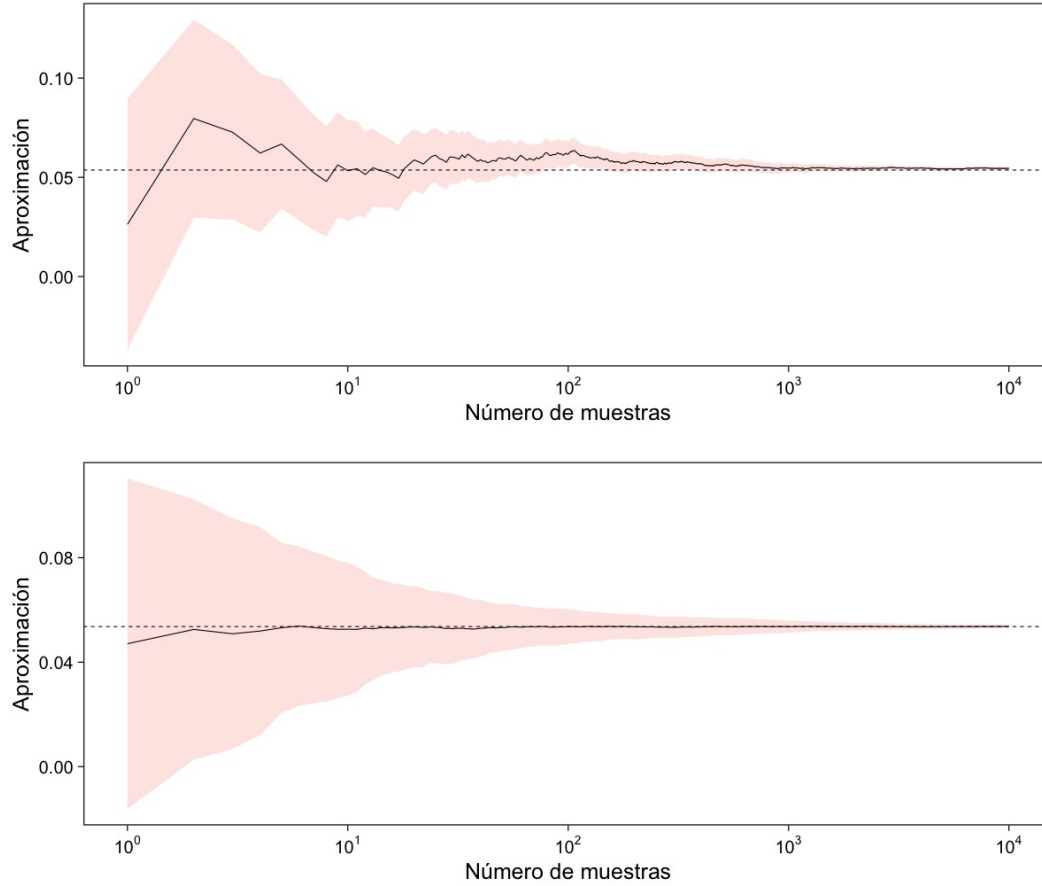
## 6.2. Segunda estrategia

Por la forma que tiene la integral podemos considerar también una integral bajo una distribución normal.

```
1 set.seed(108); nsamples <- 10**4; nexp <- 100
2 f <- function(x){ ifelse(x >= 2 & x <= 3, sqrt(2 * pi), 0) }
3 u <- rnorm(nexp * nsamples)
4 x <- array(u, c(nexp, nsamples))
5 f_x <- f(x)
```

```
1 estimador_mc <- apply(f_x, 1, cummean)      # ojo, transpone el resultado
2 media_mc <- apply(estimador_mc, 1, mean)
3 error_mc <- apply(estimador_mc, 1, sd)
```





### 6.3. Comparación

¿Cuál método preferimos?

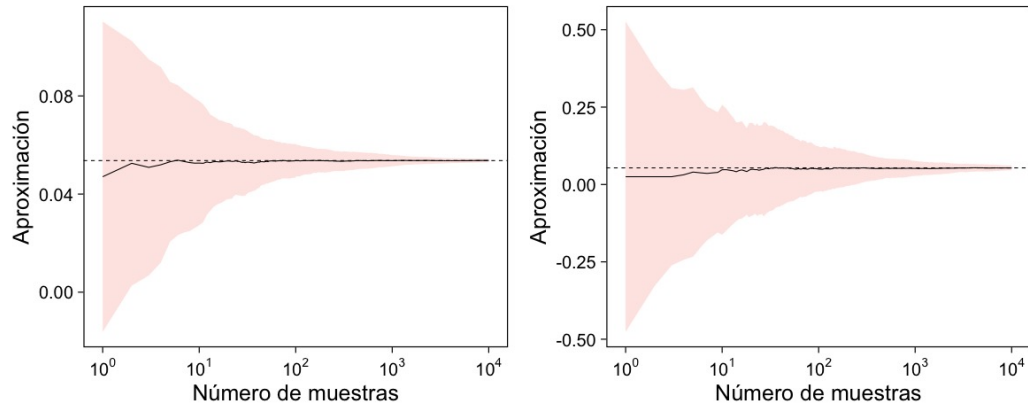


FIGURA 12. Resultados de la estimación Monte Carlo usando una uniforme (izquierda) y una normal (derecha).

### 6.4. Intervalos de confianza

Los intervalos que hemos construido y se muestran en las figuras anteriores son de la forma

$$\hat{\pi}_N^{\text{MC}}(f) \pm 2 \text{ee}(\hat{\pi}_N^{\text{MC}}(f)). \quad (19)$$

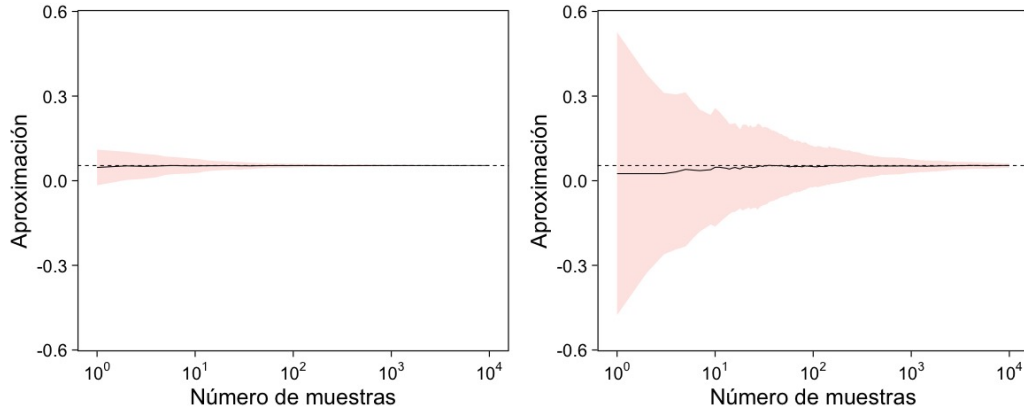


FIGURA 13. Resultados de la estimación Monte Carlo usando una uniforme (izquierda) y una normal (derecha).

Cuando platicamos de la desigualdad de Chebyshev, construimos cotas para que con una alta probabilidad  $(1 - \delta)$  nuestro estimador tenga una precisión dada  $(\epsilon)$ .

Nota que este tipo de argumentos habla sobre **nuestro método** de estimación—el método Monte Carlo para integrales—. Es decir, establece que la probabilidad de que nuestra estimación no diste mas de  $\epsilon$  unidades del verdadero valor sea tan alta como el usuario espera (al menos  $1 - \delta$ ).

**Observación.** Jamás mencionamos que el verdadero valor del parámetro se encuentre dentro del intervalo con cierta probabilidad. Es decir, **jamás** mencionamos una propiedad probabilística del verdadero valor,  $\pi(f)$ , que en nuestra argumentación suponemos desconocido.

#### 6.4.1. Experimentación:

¿Qué pasa si hacemos un ejercicio de réplicas de experimentos? Es decir, si repetimos  $M$  veces una estimación de  $\hat{\pi}_N^{\text{MC}}(f)$  y registramos cuantas veces el intervalo cubre el verdadero valor del parámetro para distintas  $N$ ?

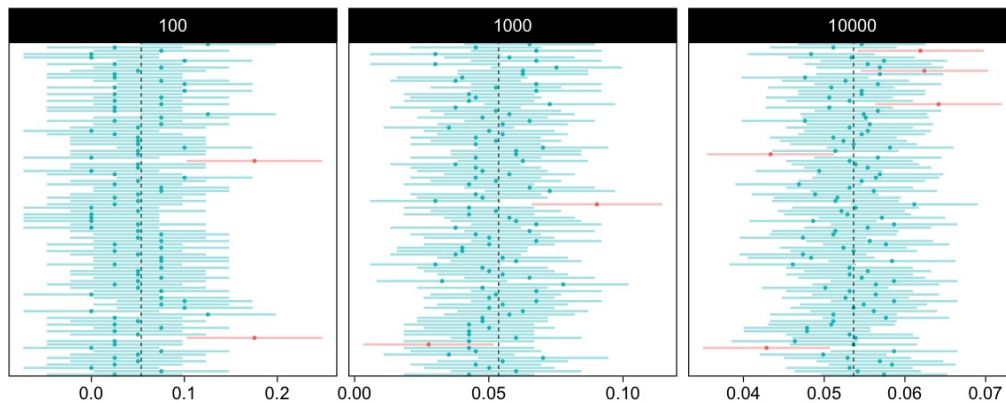


FIGURA 14. Intervalos de confianza para la integral con respecto a una Normal.

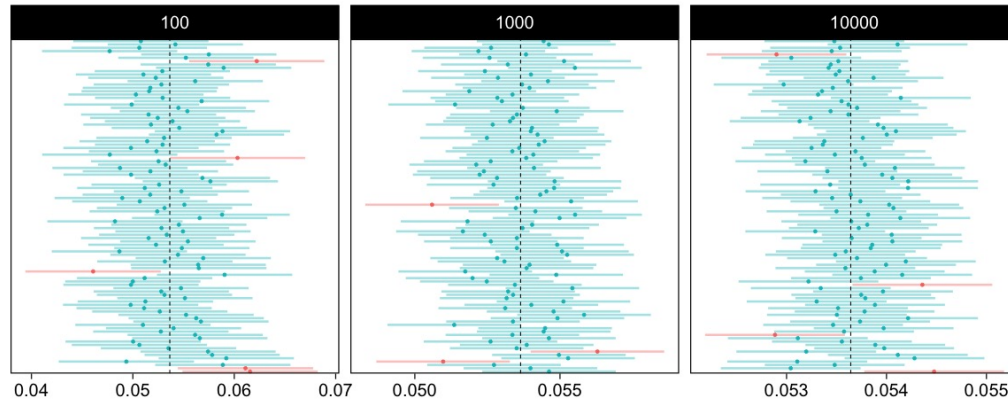


FIGURA 15. Intervalos de confianza para la integral con respecto a una Uniforme.

### 6.5. Lentitud con estimadores

Veremos un caso donde existen problemas en la estimación Monte Carlo. En particular, para estimar el error estándar. Esto se traduce en una reducción mas lenta en la incertidumbre de nuestra estimación. Y en algunos casos nuestros intervalos de confianza podrían no ser de confianza.

Supongamos que tenemos un modelo de mezcla Normal-Cauchy. Es decir,

$$\pi(x|\theta) = \frac{1}{\sqrt{2\pi}} \exp \left[ -\frac{1}{2}(x - \theta)^2 \right], \quad (20)$$

$$\pi(\theta) = \frac{1}{\pi(1 + \theta^2)}. \quad (21)$$

Nos interesa encontrar la marginal  $\pi(x)$  para valores de  $x = 0, 2, 4$ .

Nota que  $\pi(x)$  es un valor esperado bajo una distribución Cauchy. ¿Cuál es el integrando para  $\pi(f)$ ?

En algunas aplicaciones nos interesa calcular el valor esperado de  $\theta|x$  cuando asumimos una observación de  $x$ . La regla de probabilidad condicional nos permite calcular esto en términos de dos integrales.

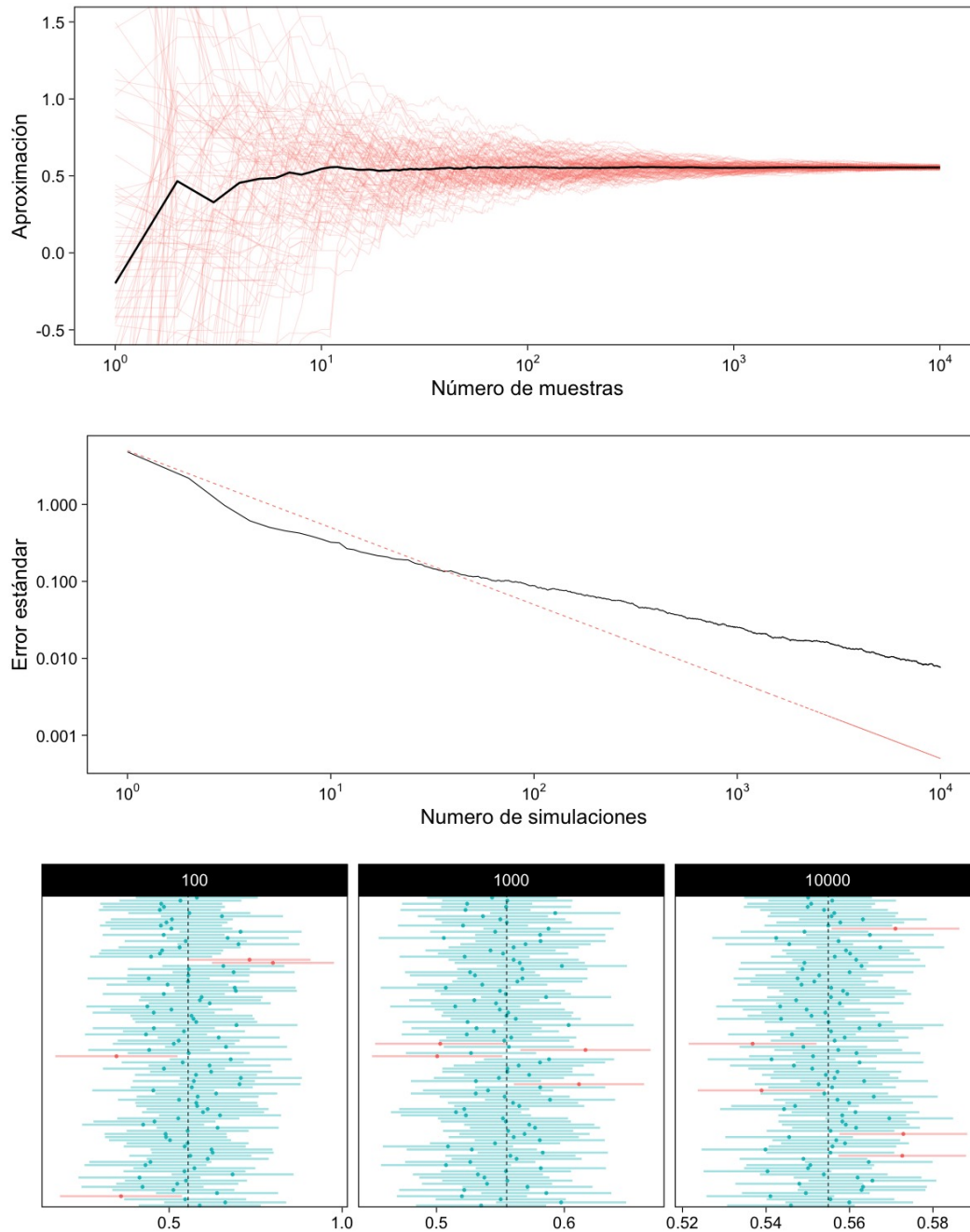
```

1 set.seed(108727); nsamples <- 10**4; nexp <- 100
2 f <- function(theta){ pi * exp(-0.5 * (1 - theta)**2) }
3 h <- function(theta){ theta * f(theta) }
4 theta <- rcauchy(nexp * nsamples)
5 theta <- array(theta, c(nexp, nsamples))
6 f_theta <- f(theta)
7 h_theta <- h(theta)

1 num_estimador_mc <- apply(h_theta, 1, cummean) # ojo, transpone el resultado
2 denom_estimador_mc <- apply(f_theta, 1, cummean) # ojo, transpone el resultado
3 estimador_mc <- num_estimador_mc / denom_estimador_mc
4 media_mc <- apply(estimador_mc, 1, mean)
5 error_mc <- apply(estimador_mc, 1, sd)

```





```
1 sessionInfo()
```

```
1 R version 4.3.1 (2023-06-16)
2 Platform: x86_64-apple-darwin20 (64-bit)
3 Running under: macOS Ventura 13.5.1
4
5 Matrix products: default
6 BLAS: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/
  libRblas.0.dylib
7 LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/
  libRlapack.dylib; LAPACK version 3.11.0
```

```

8
9 locale:
10 [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
11
12 time zone: America/Mexico_City
13 tzcode source: internal
14
15 attached base packages:
16 [1] stats      graphics  grDevices datasets  utils      methods
17 [7] base
18
19 other attached packages:
20 [1] scales_1.2.1      patchwork_1.1.2  lubridate_1.9.2  forcats_1.0.0
21 [5] stringr_1.5.0     dplyr_1.1.2      purrr_1.0.1      readr_2.1.4
22 [9] tidyr_1.3.0       tibble_3.2.1     ggplot2_3.4.2    tidyverse_2.0.0
23
24 loaded via a namespace (and not attached):
25 [1] gtable_0.3.3      compiler_4.3.1    renv_1.0.0
26 [4] crayon_1.5.2      tidyselect_1.2.0  R6_2.5.1
27 [7] labeling_0.4.2    generics_0.1.3    isoband_0.2.7
28 [10] MASS_7.3-60       munsell_0.5.0     pillar_1.9.0
29 [13] RColorBrewer_1.1-3 tzdb_0.4.0        rlang_1.1.1
30 [16] utf8_1.2.3        stringi_1.7.12    timechange_0.2.0
31 [19] cli_3.6.1         withr_2.5.0       magrittr_2.0.3
32 [22] grid_4.3.1        hms_1.1.3         lifecycle_1.0.3
33 [25] vctrs_0.6.3       glue_1.6.2        farver_2.1.1
34 [28] fansi_1.0.4       colorspace_2.1-0  tools_4.3.1
35 [31] pkgconfig_2.0.3

```

## Referencias

- [1] S. Reich and C. Cotter. *Probabilistic Forecasting and Bayesian Data Assimilation*. Cambridge University Press, Cambridge, 2015. [1](#), [4](#)
- [2] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Science & Business Media, mar 2013. [1](#)
- [3] D. Sanz-Alonso, A. M. Stuart, and A. Taeb. Inverse Problems and Data Assimilation. *arXiv:1810.06191 [stat]*, jul 2019. [1](#)