Práctica 5: Método Monte-Carlo

Fabiola Vázquez

21 de octubre de 2020

---

1. Introducción

El objetivo de esta práctica [5] es aproximar el resultado de una integral utilizando el método Monte-Carlo, variando el tamaño de la muestra, al valor proporcionado por WolframAlpha [7]. El experimento se realiza con el software R versión 4.0.2 [3] en un cuaderno de Jupyter [1].

2. Experimento

Se quiere aproximar el resultado de la siguiente integral

$$\int_3^7 \frac{dx}{\exp(x) + \exp(-x)}. \tag{1}$$

Para ello se generan números pseudoaleatorios con distribución $g(x) = \frac{2}{\pi} f(x)$, donde $f(x) = \frac{1}{\exp(x)+\exp(-x)}$, haciendo uso de la función `r(AbscontDistribution(d = g))`. El código del experimento [6] está basado mayormente, en el código implementado por Schaeffer [4].

Como se quiere comparar el valor de la integral obtenida mediante la simulación con el valor proporcionado por Wolfram (denotado como `wolfram`), la función `decimalesexactos`, descrita en el código 1, nos regresa la cantidad de decimales en que son coinciden dichos valores.

En el experimento se varía el valor del parámetro `pedazo` de 1000 a 10000 en saltos de 1000, pero no se aproximaba tanto al valor de `wolfram`, por lo que se varía en otros rangos como de 50 a 500 y de 50000 a 95000, haciendo 50 réplicas con cada uno. El cuadro 1 muestra un fragmento de los datos recopilados en el experimento y la figura 1 muestra los gráficos de caja bigote de los valores de la integral obtenidos en el experimento. Como se puede apreciar en dicha figura, a mayor valor de `pedazo` más se aproxima al valor de `wolfram`.

Cuadro 1: Fragmento de los datos obtenidos en el experimento

Corrida	Pedazo	Integral	Precisión decimal
1	1000	0.0488361	5
105	3000	0.0488884	4
210	5000	0.0488486	4
320	7000	0.0488275	4
400	8000	0.0487473	3
700	65000	0.0489142	3
900	85000	0.0488143	4

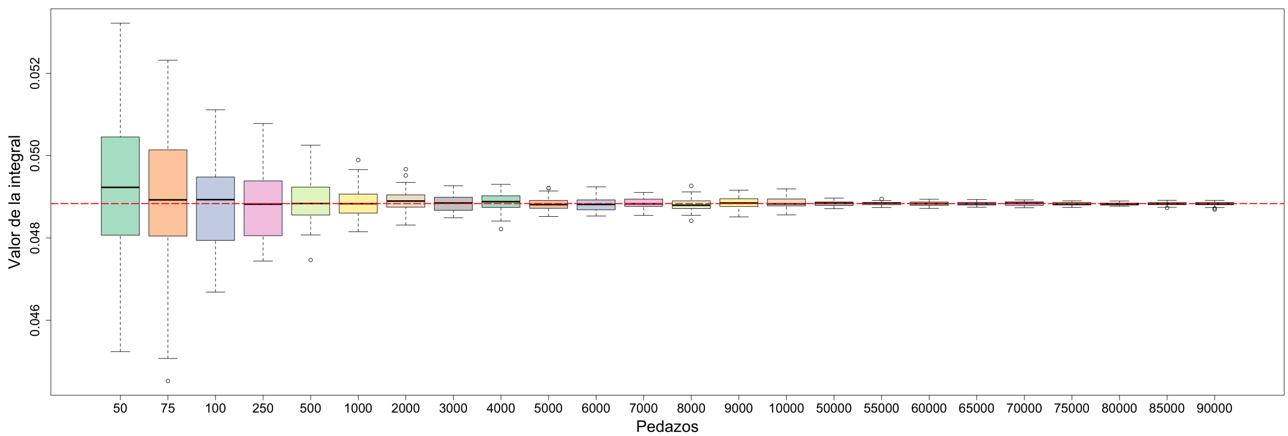

Figura 1: Gráficas de caja bigote de los valores de la integral obtenidos variando el tamaño de `pedazo`. En rojo tenemos el valor de `wolfram`.

En la figura 2 se muestran los decimales exactos obtenidos en las variaciones del tamaño de `pedazo`. Como se aprecia, entre mayor es el tamaño más decimales se aproximan al valor de `wolfram`. Aún cuando el tamaño de `pedazo` se varía en una gran número, no fue posible aproximarse al valor de `wolfram` a mas de cuatro decimales, y aumentar a un mayor valor no fue computacionalmente posible.

Código 1: Función `decimalesexactos`.

```
decimalesexactos <- function (x){
    for (i in 1:6){
        if (!(trunc(wolfram*10^i)==trunc(x*10^i))){
            return(i-1)
            break
        }
    }
    return(6)
}
```

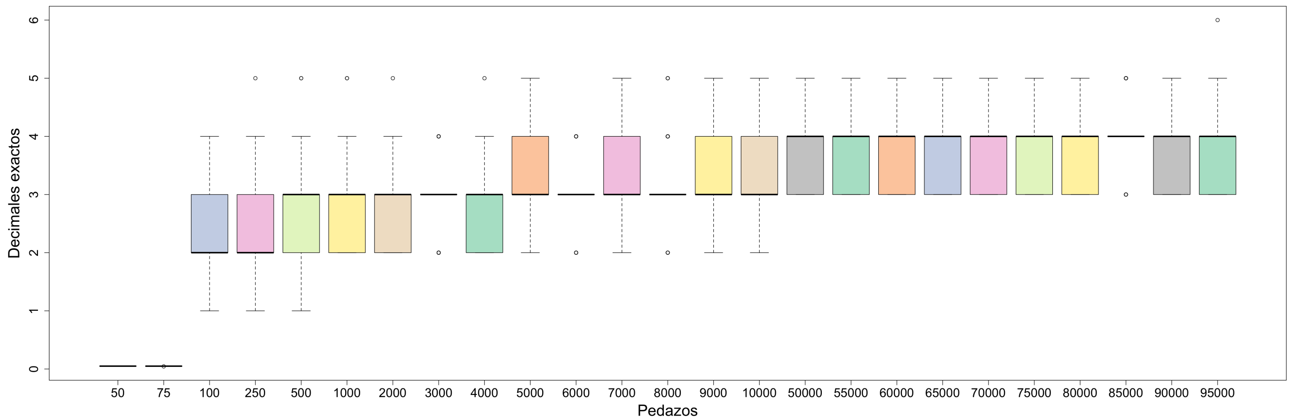

Figura 2: Gráficas de caja bigote de la cantidad de decimales en que coinciden los valores obtenidos de la integral con el valor de `wolfram`.

2.1. Pruebas de correlación

Se realiza la prueba de correlación para las columnas del tamaño de `pedazo` y la cantidad de decimales exactos. La prueba arroja un valor p de 2.2×10^{-16} el cual es menor que 0.05, por lo que se rechaza la hipótesis nula y se concluye que hay correlación entre estas variables. Es decir, el tamaño de `pedazo` afecta a la cantidad de decimales exactos. En particular, se estima el coeficiente de correlación en 0.5576 por lo que se concluye que a mayor tamaño de `pedazo`, más decimales exactos hay.

```
Pearson's product-moment correlation

data:  datos$Pedazo and datos$`Decimales Exactos`
t = 23.753, df = 1250, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5182761 0.5946934
sample estimates:
      cor
0.5576652
```

3. Reto 1

Para el reto 1, hay que automatizar y paralelizar la estimación del valor π de Kurt [2]. De manera similar que la tarea base, se varía el tamaño del parámetro `runs` en potencias de 10. En la figura 3 se aprecia que mientras es mayor el tamaño de `runs` más se aproxima al valor de `pi` = 3.141592

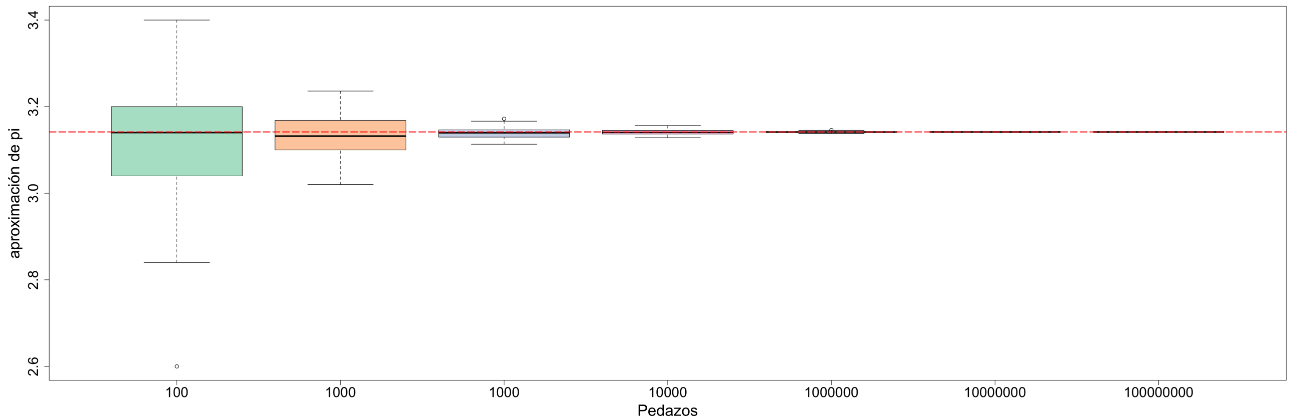

Figura 3: Gráficas de caja bigote de las aproximaciones de `pi` obtenidas variando el tamaño de `runs`.

y en la figura 4 se muestran la cantidad de decimales a los que se aproxima. Aún aumentando a gran tamaño `runs`, no se pudo aproximar a más que cuatro decimales.

Referencias

[1] Thomas Kluyver, Benjamin Ragan-Kelley, Pérez, et al. Jupyter notebooks—a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas: Proceedings of the 20th International Conference on Electronic Publishing*, page 87. IOS Press, 2016.

[2] Will Kurt. 6 neat tricks with monte carlo simulations – count bayesie; probably a probability blog. https://www.countbayesie.com/blog/2015/3/3/6-amazing-trick-with-monte-carlo-simulations.

[3] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.

[4] Elisa Schaeffer. Integral. https://github.com/satuelisa/Simulation/blob/master/MonteCarlo/integral.R.

[5] Elisa Schaeffer. Práctica 5: método Monte-Carlo. https://elisa.dyndns-web.com/teaching/comp/par/p5.html.

[6] Fabiola Vázquez. Método Monte-Carlo. https://github.com/fvzqa/Simulacion/blob/master/Tarea5/Tarea5.ipynb.

[7] WolframAlpha. https://www.wolframalpha.com/input/?i=integrate++%281%2F%28exp%28x%29%2Bexp%28-x%29%29%29+from+3+to+7.

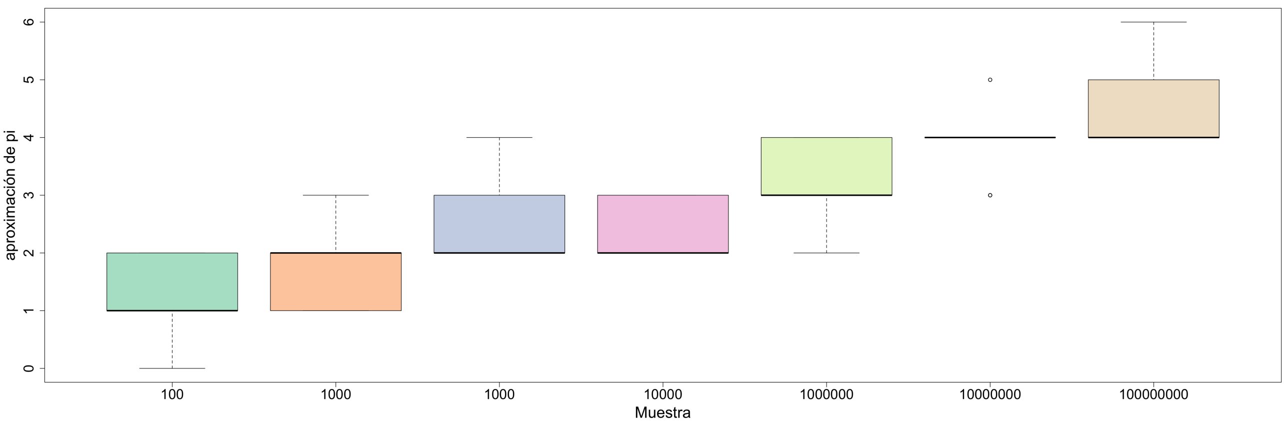

Figura 4: Gráficas de caja bigote de la cantidad de decimales que coincide la aproximación de π obtenida con el valor de `pi`.