# Práctica 5: método Monte Carlo

C. A. Estrada

21 de octubre de 2020

## 1. Objetivo

1. Determinar el tamaño de muestra que se requiere por cada lugar decimal de precisión del estimado obtenido para el integral para por lo menos desde uno hasta siete decimales [1], comparando con el resultado obtenido en Wolfram Alpha.

2. Aplicar un método Monte Carlo para la estimación de la pintura necesaria para un mural y comparar el resultado con un conteo exacto.

## 2. Determinación de tamaños de muestra para el integral.

### 2.1. Metodología

Para efectos de esta práctica, se utiliza el paquete estadístico R versión 4.0.2 [2]. Se pretende calcular el valor de la integral (1) para la función (2) empleando el método Monte Carlo, y comparar el valor obtenido por Wolfram Alpha de 0.048834 [3].

$$\int_3^7 f(x)d(x) \tag{1}$$

$$f(x) = \frac{1}{exp(x) + exp(-x)} \tag{2}$$

Se genera el código para calcular el valor del integral conforme a lo ya preestablecido [1], empleando números de muestra de 10, 100, 1,000, 10,000, 100,000 y 500,000, realizando diez réplicas para cada una, y posteriormente se calcula el porcentaje de error obtenido respecto al valor de 0.048834. Finalmente, se grafican los resultados en un solo diagrama de caja-bigote.

### 2.2. Resultados y discusión

En la figura 1 se muestran los resultados obtenidos por la aplicación del método Monte Carlo para la estimación del integral. Se observa que conforme se aumenta la cantidad de muestra el resultado va acercándose más al valor obtenido en Wolfram Alpha [3], ya que el porcentaje de error va reduciéndose y se observa que la mediana se encuentra más cerca del valor real (línea roja).

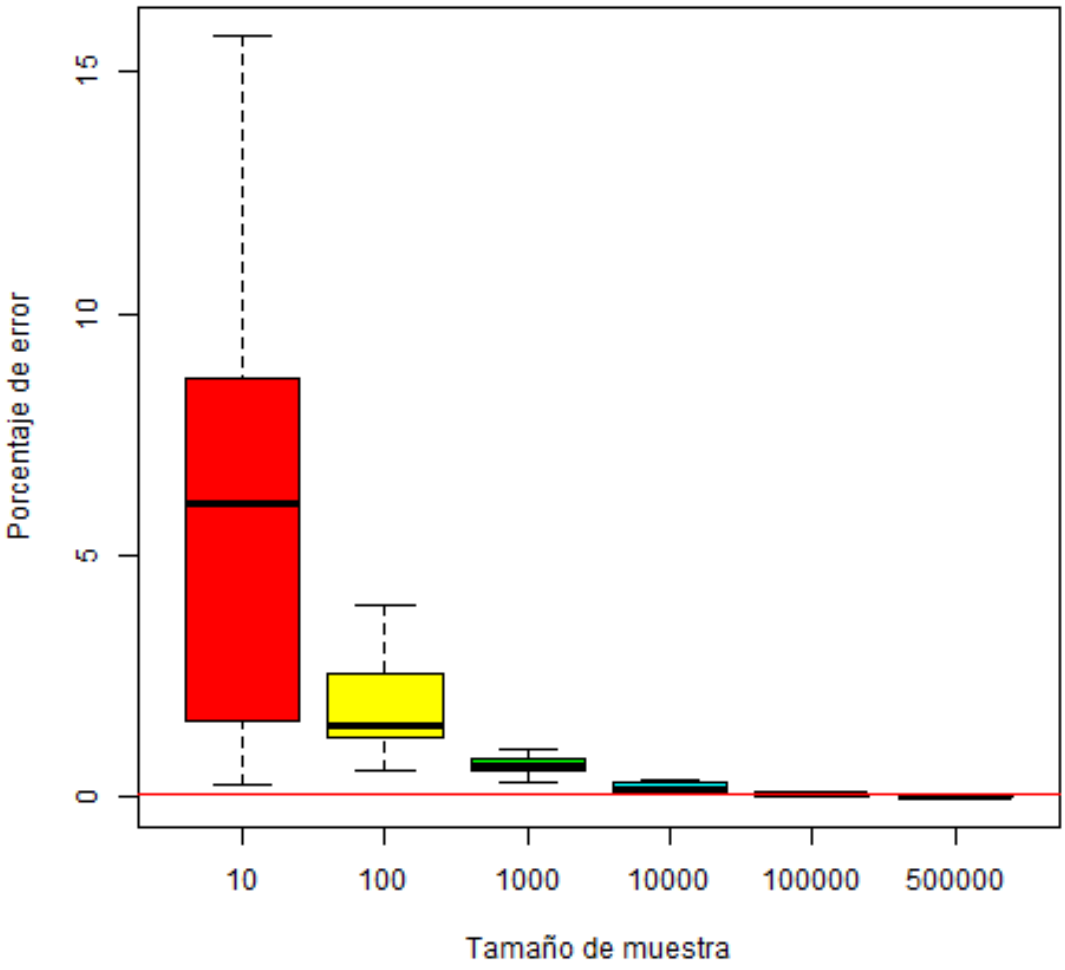


Figura 1: Porcentajes de error obtenidos conforme aumenta la muestra. La línea roja corresponde al valor de 0.048834 para la integral.

Cuando el tamaño de muestra es mínimo, el porcentaje de error es el más alto, presentando la mayor cantidad de variación en los resultados, mientras que en los tamaños de muestra mayores (100,000 y 500,000), las cajas del diagrama correspondientes a estos valores se presentan, prácticamente, como una línea sobre el valor de 0.048834.

### 2.3. Conclusión

El aumento del tamaño de muestra para la estimación del integral utilizando el método Monte Carlo genera resultados más cercanos al valor real del integral.

## 3. Estimación de la cantidad de pintura.

### 3.1. Metodología

Se emplea el paquete estadístico R versión 4.0.2 [2] para la generación del código, y a su vez, el fundamento de una simulación Monte Carlo [4]. Se elige una imagen para un mural, en este caso el logotipo de Coca Cola [5], en donde se toma en cuenta que cada pixel corresponde a $10cm^2$ del mural, y que cada litro de pintura rinde $10m^2$. Para la cuenta exacta de pixeles se emplea el paquete "Countcolors" en R [6], el cual toma en cuenta rangos definidos de color de pixeles, y además genera una imagen en donde sustituye los pixeles a contar por algún otro color para confirmar que corresponda a la región de interés.

```
white.center = c(1,1,1)
red.center=c(1,0,0)
blanco=coca.white\$pixel.count
rojo=coca.red\$pixel.count
pixeles=c(blanco,rojo)
pintura=pixeles*0.001
```

Para estimación por el método Monte Carlo se realiza un muestreo al azar de 10,000 datos a partir de la matriz que corresponde a los pixeles azules de la imagen empleada, debido a que en esta matriz se puede diferenciar el color blanco del rojo por sus valores numéricos, en donde el color rojo debe de ser cercano o igual a cero, mientras que el blanco se acerca al uno. Se realizan 1,000 repeticiones del experimento, para obtener la media de pixeles correspondientes al color rojo, y con ella la estimación de Monte Carlo. La diferencia de este resultado con el total de pixeles de la imagen (160,000) corresponde a la estimación de pixeles de color blanco.

```
runs=1000
for(r in 1:length(runs)){
  for(s in 1:10000){
    blue = coca[,,3]
    x =sample (blue, runs[r])
    y=sum(x < 0.5)
    print(y)
  }
}
pix = datos [,3]
pixm = mean(pix)
```

(a) Original. (b) Pixeles rojos. (c) Pixeles blancos.

Figura 2: Imagen original del logo de Coca Cola y las sustituciones de color de pixeles objetivo generadas con countcolors en R.

```
montecarlorojo = pixm*(160000/runs)
mcblanco=160000-montecarlorojo
pixelesmc=c(mcblanco, montecarlorojo)
pinturamc=pixelesmc*0.001
```

Una vez obtenidos los pixeles correspondientes a cada color, se calcula la cantidad de pintura necesaria para un mural de $160m^2$ de la imagen.

## 3.2. Resultados y discusión

En el cuadro 1 se presenta el número de pixeles de color rojo y blanco y su equivalente el litros de pintura para la estimación con números aleatorios y la cuenta exacta. En la figura 2 se muestra la figura original que se emplea en el experimento [5], así como las figuras generadas con el paquete countcolors de R, que sustituye los pixeles rojos por color azul y los pixeles blancos por color magenta, como confirmación de los pixeles contados para cada color.

Cuadro 1: Estimación de pixeles y litros de pintura por cada color.

| **Color** | **Monte Carlo** | | **Cuenta exacta** | |
|---|---|---|---|---|
| | Pixeles | Pintura | Pixeles | Pintura |
| **Rojo** | 149,075 | 149.01 L | 148,524 | 148.52 L |
| **Blanco** | 10,925 | 10.93 L | 11,336 | 11.34 L |

Se observa que en la aproximación realizada por Monte Carlo se obtuvo una leve diferencia respecto al conteo exacto de pixeles, debido a los parámetros del algoritmo de countcolors para su cuantificación, del cual, por la imagen que se genera, se observa que sí abarcó las regiones correctas de color. Por otra parte, para la estimación se tomó en cuenta que en la matriz en donde se realiza el muestreo algunos valores que corresponden a rojo no son exactamente igual a cero ($x<0.5$), así que el uso de este parámetro es lo que ocasiona la diferencia entre ambos resultados. No obstante, para fines prácticos, sí da una idea aproximada de la cantidad de pintura que se tendría que emplear.

### 3.3. Conclusión

La implementación del método Monte Carlo para la estimación de pintura sí da una idea aproximada de la cantidad de pintura que se tendría que emplear.

## Referencias


[1] E. Schaeffer. Práctica 5: método monte-carlo, 2020. URL https://elisa.dyndns-web.com/teaching/comp/par/p5.html.

[2] The R Foundation. The R Project for Statistical Computing, 2020. URL https://www.r-project.org/.

[3] Wolfram Alpha, 2020. URL https://www.wolframalpha.com/.

[4] W. Kurt. Monte Carlo Simulations in R, 2015. URL https://www.countbayesie.com/blog/2015/3/3/monte-carlo-simulations.

[5] Coca Cola Company. Coca Cola Logo, 2010. URL https://upload.wikimedia.org/wikipedia/commons/c/cb/Coca-Cola-Logo.jpg.

[6] H. Weller. Introduction to countcolors package, 2019. URL https://cran.r-project.org/web/packages/countcolors/vignettes/Introduction.html.