Ejercicio 14.1

Augusto Souto & Federico Molina

29 de mayo de 2019

Índice

1. Ejercicio 14:

Partiendo de uno de los códigos elaborados para resolver el ejercicio 6.2, utilizar el método de muestreo estratificado para calcular la integral de la función $x_1 x_2^2 x_3^3 x_4^4 x_5^5$ sobre el hipercubo J^m de dimensión $m = 5$ en base a 10^6 iteraciones. Calcular media, desviación estándar y un intervalo de confianza de nivel 95 %. Comparar con los resultados obtenidos con el código del ejercicio 6.2.

Sugerencia: definir 5 estratos, en función del valor de x_5, tomando los siguientes intervalos: $[0, 075)$, $[075, 085)$, $[085, 090)$, $[090, 095)$, $[095, 1]$. Hacer dos experimentos, uno tomando $\frac{10^6}{5}$ iteraciones en cada estrato, otro tomando una cantidad de iteraciones proporcional a la probabilidad de cada estrato.

2. Resolución

En primer lugar, se simulan los puntos del hipercubo en cada uno de los estratos mediante una función, a la que llamaremos sorteo. Dicha función recibe como parámetros de entrada un vector con la cantidad de puntos a simular en cada estrato y una semilla que permite reproducir los resultados. Por otra parte, la función devuelve una lista con 5 matrices, las cuales representan los puntos simulados en cada estrato.

```r
sorteo<-function(n, semilla){

set.seed(semilla)

x_1_1<-runif(n[1], 0, 1) %>% as.matrix()
x_2_1<-runif(n[1], 0, 1) %>% as.matrix()
x_3_1<-runif(n[1], 0, 1) %>% as.matrix()
x_4_1<-runif(n[1], 0, 1) %>% as.matrix()
x_5_1<-runif(n[1], 0, 0.75) %>% as.matrix()

estrato_1<-cbind(x_1_1,x_2_1, x_3_1, x_4_1, x_5_1)


#zona 2
x_1_2<-runif(n[2], 0, 1) %>% as.matrix()
x_2_2<-runif(n[2], 0, 1) %>% as.matrix()
x_3_2<-runif(n[2], 0, 1) %>% as.matrix()
x_4_2<-runif(n[2], 0, 1) %>% as.matrix()
x_5_2<-runif(n[2], 0.75, 0.85) %>% as.matrix()

estrato_2<-cbind(x_1_2,x_2_2, x_3_2, x_4_2, x_5_2)

#zona 3
x_1_3<-runif(n[3], 0, 1) %>% as.matrix()
x_2_3<-runif(n[3], 0, 1) %>% as.matrix()
x_3_3<-runif(n[3], 0, 1) %>% as.matrix()
x_4_3<-runif(n[3], 0, 1) %>% as.matrix()
x_5_3<-runif(n[3], 0.85, 0.90) %>% as.matrix()

estrato_3<-cbind(x_1_3,x_2_3, x_3_3, x_4_3, x_5_3)

#zona 4
x_1_4<-runif(n[4], 0, 1) %>% as.matrix()
x_2_4<-runif(n[4], 0, 1) %>% as.matrix()
x_3_4<-runif(n[4], 0, 1) %>% as.matrix()
x_4_4<-runif(n[4], 0, 1) %>% as.matrix()
x_5_4<-runif(n[4], 0.90, 0.95) %>% as.matrix()
```

```r
estrato_4<-cbind(x_1_4,x_2_4, x_3_4, x_4_4, x_5_4)

#zona 5
x_1_5<-runif(n[5], 0, 1) %>% as.matrix()
x_2_5<-runif(n[5], 0, 1) %>% as.matrix()
x_3_5<-runif(n[5], 0, 1) %>% as.matrix()
x_4_5<-runif(n[5], 0, 1) %>% as.matrix()
x_5_5<-runif(n[5], 0.95, 1)    %>% as.matrix()

estrato_5<-cbind(x_1_5,x_2_5, x_3_5, x_4_5, x_5_5)

return(list(estrato_1, estrato_2, estrato_3, estrato_4, estrato_5))

}
```

Por otro lado, se utiliza primero un numero de iteraciones de $n_i = \frac{10^6}{6}$ para todos los estratos para luego probar con $n_i = p_i 10^6$ donde p_i es la masa de probabilidad que acumula cada estrato del hipercubo. Por lo tanto, los vectores de entrada en la función sorteo son

$$n = \begin{pmatrix} \frac{10^6}{6} & \frac{10^6}{6} & \frac{10^6}{6} & \frac{10^6}{6} & \frac{10^6}{6} & \frac{10^6}{6} \end{pmatrix}$$

y

$$n = \begin{pmatrix} p_1 10^6, & p_2 10^6, & p_3 10^6, & p_4 10^6, & p_5 10^6, & p_6 10^6 \end{pmatrix}$$

Los sorteos se hacen con los siguientes comandos:

```r
#DEFINO LOS PARAMETROS DE ENTRADA DE LA FUNCION PARA EL CASO N PROPORCIONAL AL ESTRATO
n<-10^6
probs<-c(0.75,0.10, 0.05, 0.05, 0.05, 0.05)
n_1<-probs[1]*n
n_2<-probs[2]*n
n_3<-probs[3]*n
n_4<-probs[4]*n
n_5<-probs[5]*n

n_ponderado<-c(n_1, n_2, n_3,n_4, n_5)

#HAGO EL SORTEO DE PUNTOS EN EL HIPERCUBO
matriz_1<-sorteo(n=n_ponderado, semilla=10)

#DEFINO N PARA EL CASO NO PROPORCIONAL
n_equiprobable<-rep(10^6,5) %>% c()

#HAGO EL SORTEO DE PUNTOS EN EL HIPERCUBO
matriz_2<-sorteo(n=n_equiprobable, semilla=10)
```

Como se observa, primero se simularon los puntos en el caso donde el numero de iteraciones es proporcional a la masa de probabilidad del estrato y luego se simularon los puntos en el caso donde el numero de iteraciones es igual para todos los estratos del hipercubo.

Luego, usamos la función que nos permite computar el valor estimado de la integral, la varianza muestral y los intervalos de confianza. Esta función recibe como parámetros de entrada el nivel de confianza, la matriz con los puntos del estrato y el numero de puntos a tomar para calcular las estimaciones.

```r
estimacion<-function( alfa=0.05, matriz, n){

   valor<-vector("numeric", length=n)
  #valores que toma la funci?n en los puntos sorteados dentro de la region#
```

```r
  for (i in 1:n) {
    valor[i]=(matriz[i,1]*matriz[i,2]^2*matriz[i,3]^3*matriz[i,4]^4*matriz[i,5]^5)
  }

  s<-vector("numeric", length=n)
  t<-vector("numeric", length=n)

  t[1]=0
  s[1]=valor[1]

  for (i in 2:n) {
    t[i]<-t[i-1]+(1-(1/i))*(valor[i] -( s[i-1] / (i-1)) )^2
    s[i]<-s[i-1]+valor[i]
  }

  int<-s[n]/n
  var_int<-t[n]/(n-1)
  var_est<-var_int/n
  inf<-int-qnorm(1-alfa/2)*(var_int/n)^(1/2)
  sup<-int+qnorm(1-alfa/2)*(var_int/n)^(1/2)

  return(c(int, var_int, inf, sup))
}
```

Por lo tanto, combinando las estimaciones de los estratos podemos obtener las estimaciones del método de mustreo estratificado. Para llevar a cabo esto, definimos la función "estimación hipercubo", la misma toma como parámetros de entrada al vector de probabilidades asignadas a cada estrato (construido a partir de la masa de probabilidad de cada estrato), a la matriz de puntos que se sorteo anteriormente y al vector que define el número de iteraciones usadas en cada estrato.

```r
estimacion_hipercubo<-function(probs, matriz, n){
for (i in 1:5) {
  if(i==1){
  int_tot<-probs[i]*estimacion(matriz = matriz[[i]], n=n[i])[1]
  var_int<-((probs[i]^2)*(estimacion(matriz =  matriz[[i]], n=n[i])[2])^2)/n[i]
  }else{
  int_tot=int_tot+probs[i]*estimacion(matriz =  matriz[[i]], n=n[i])[1]
  var_int=var_int+((probs[i]^2)*(estimacion(matriz =  matriz[[i]], n=n[i])[2])^2)/n[i]
  }

  }
  return(c(int_tot, var_int))
}
```

Como se ve, la función devuelve la estimación de la integral y la varianza muestral. Utilizando la última, se calculan los intervalos de confianza al 95 %.

```r
#ESTIMACIONES PARA EL CASO N PROPORCIONAL
caso_1<-estimacion_hipercubo( matriz=matriz_1, probs=probs, n=n_ponderado)

inf_1<-caso_1[1]-qnorm(1-0.05/2)*(caso_1[2]/n)^(1/2)
sup_1<-caso_1[1]+qnorm(1-0.05/2)*(caso_1[2]/n)^(1/2)

#ESTIMACIONES PARA EL CASO N NO PROPORCIONAL
caso_2<-estimacion_hipercubo( matriz=matriz_2, probs=probs, n=n_equiprobable)

inf_2<-caso_2[1]-qnorm(1-0.05/2)*(caso_2[2]/n)^(1/2)
```

```
sup_2<-caso_2[1]+qnorm(1-0.05/2)*(caso_2[2]/n)^(1/2)
```

Para el caso con "n" proporcional, se estimó integral con un valor de 0.0014004413 y a la varianza muestral en $4705588e-14$. Por otro lado, el intervalo de confianza està definido entre 0.0014004409 y 0.0014004417.

Para el caso no proporcional, la estimación de la integral es de 0.0013907521 y la varianza muestral estimada es $2363981e-15$. El intervalo de confianza està definido entre 0.0013907520 y 0.0013907522.

Estos resultados contrastan con los obtenidos mediante el calculo del ejercicio 6.2, donde la integral estimada es de 0.0013895204, con varianza muestral de $9398566088e-05$ e intervalo ubicado entre 0.0013705193 y 0.0014085215.

Como se observa, las varianzas se reducen de forma significativa tras aplicar el método de estratificación. El caso en el que se logra reducir más la varianza fue el que asigno las iteraciones de manera no proporcional.

Por otro lado, mientras la distancia del estimador del ejercicio anterior con el valor analítico de la integral fue de $6315313e-07$, en el caso del método con n proporcional la distancia fue de $115524e-05$ y en el caso no proporcional de $1863199e-06$. Por lo tanto, se reduce la varianza significativamente pero aumenta el sesgo del estimador.

3. Datos Sobre la Implementación del Ejercicio

Se utilizó el lenguaje R y el equipo utilizado fue un Intel core i3 M380, con 4 gb de memoria y 2.53 GHz.