Simulación ITAM
27 de octubre de 2020 2020-II

Tarea 4.

La fecha de entrega es el **26 de octubre de 2020**.

Lecturas

- Robert & Casella Caps. 3 y 4.
- Dagpunar Cap.5 (Técnicas de Reducción de varianza)

Problemas

1. Consideren el siguiente modelo de líneas de espera con un servidor. Los tiempos de interarribo, así como los tiempos de servicio son independientes $\mathcal{U}(0,1)$.

   Sea T_i el tiempo de interarribo entre los clientes $i-1$ e i y S_i el tiempo de servicio del cliente i. W_i es el tiempo de espera en fila del cliente i. La condición inicial es que el primer cliente en el sistema llega en el tiempo 0. Entonces:

   $$W_i = max\{0, W_{i-1} + S_{i-1} - T_i\}$$

   para $i = 2, 3, \ldots, 100$, donde $W_1 = 0$. Escriban un programa para simular 5000 realizaciones del tiempo total de espera en la fila, junto con 5000 realizaciones antitéticas.

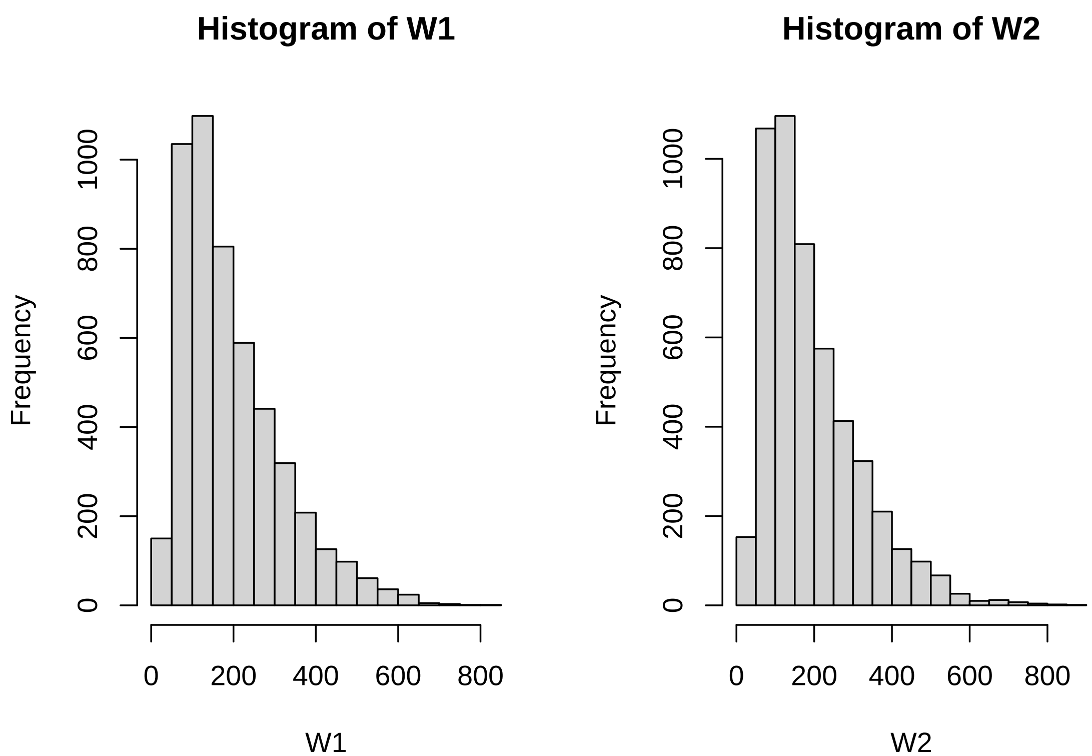
   a. Usando un estimador combinado de las realizaciones primarias y antitéticas, estimar la esperanza del tiempo de espera de los 100 clientes y su error estándar estimado. Estimar el porcentaje de reducción de varianza.
   b. Repetir el experimento cuando la duración del servicio es $\mathcal{U}(0,2)$. ¿Porqué se alcanza una reducción de varianza mucho mejor aquí que en (a)?

   Solución.

   La siguiente función genera dos vectores `W1` y `W2` correspondiente a la variable W_i con variables directas y antitéticas respectivamente. La escribí en forma de función para poder repetir la simulación en el inciso (b).

```r
simula <- function(b=1){
        W1 <- W2 <- NULL #guardar las muestras de las w[100]
        w1 <- w2 <- 0  #cada una de las realizaciones directa y antitética
        for(j in 1:5000){
                u <- runif(99,max=b)
                v <- runif(99,max=b)
                k <- ifelse(b==1,1,2) - u
                l <- ifelse(b==1,1,2) - v
                for(i in 2:100){
                        w1 <- append(w1,max(0,w1[i-1] + u[i-1]-v[i-1]))
                        w2 <- append(w2,max(0,w2[i-1] + k[i-1]-l[i-1]))
                }
                W1[j] <- sum(w1)
                W2[j] <- sum(w2)
                w1 <- w2 <- 0
        }
        return(cbind(W1,W2))
}

W <- simula()
W1 <- W[,1]
W2 <- W[,2]
par(mfrow = c(1,2))
hist(W1)
hist(W2)
```



Para el inciso (a), basta promediar `W1` y `W2` y podemos comparar la reducción de varianza, por ejemplo, con respecto a `W1` o a `W2`

```r
W <- (W1 + W2)/2
c(mean(W), mean(W1), mean(W2)) #medias
```

```
[1] 193.3285 193.8621 192.7949
```

```r
c(sd(W), sd(W1), sd(W2))#desviaciones estándar
```

```
[1]  51.56629 121.10646 122.09407
```

```
#reducción de varianza respecto a W1:
100*(var(W1)- var(W))/var(W1)
```

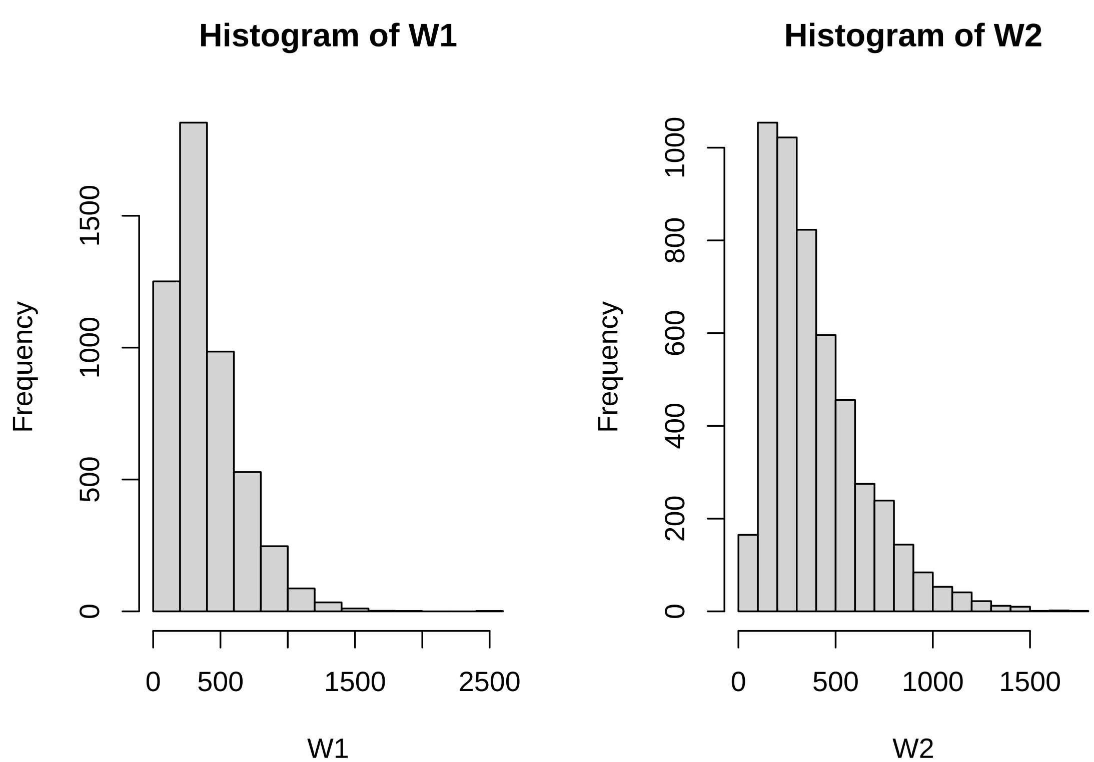
```
[1] 81.87002
```

```
#reducción de varianza respecto a W2:
100*(var(W2)- var(W))/var(W2)
```

```
[1] 82.16214
```

Para el inciso (b), repetimos el código pero cambiando el límite a 2. La reducción de varianza debe ser mayor porque la covarianza aumenta.

```
W <- simula(b=2)
W1 <-W[,1]
W2 <-W[,2]
par(mfrow=c(1,2))
hist(W1)
hist(W2)
```




```
W <- (W1 + W2)/2
c(mean(W), mean(W1), mean(W2)) #medias
```

```
[1] 390.6381 389.4418 391.8343
```

```
c(sd(W), sd(W1), sd(W2))#desviaciones estándar

[1] 106.7977 250.8540 248.6775

#reducción de varianza respecto a W1:
100*(var(W1)- var(W))/var(W1)

[1] 81.87484

#reducción de varianza respecto a W2:
100*(var(W2)- var(W))/var(W2)

[1] 81.55617
```

$\square$

2. Cinco elementos, numerados del 1 al 5 se acomodan inicialmente en un orden aleatorio (esto es, el orden inicial es una permutación aleatoria de los números {1,2,3,4,5}) En cada estado del proceso, uno de los elementos es seleccionado con cierta probabilidad y puesto en el frente de la lista. Por ejemplo, si el orden presente es {2,3,4,1,5} y el elemento 1 se elige, entonces el nuevo orden es {1,2,3,4,5}. Supongan que cada selección es independiente y se elige al elemento i con probabilidad p_i, donde las $p_i's$ son $(\frac{1}{15}, \frac{2}{15}, \frac{3}{15}, \frac{4}{15}, \frac{5}{15})$. Sea L_j la variable que denota la posición del j-ésimo elemento seleccionado, y sea $L = \sum_{j=1}^{100} L_j$. Queremos usar simulación para estimar $E(L)$.

   (a) Expliquen cómo realizarían la simulación para estimar $E(L)$.

   (b) Calculen $E(N_i)$ donde N_i es el número de veces que el elemento i es elegido en 100 selecciones.

   (c) Sea $Y = \sum_{i=1}^{5} iN_i$ ¿Cómo se correlaciona Y con L?

   (d) Estimar L usando Y como variable de control.

Solución.

a. Partiendo de una permutación $\sigma_0 = \sigma(\{1, 2, 3, 4, 5\})$ inicial, podemos utilizar el siguiente algoritmo. Para $j = 1, \ldots, 100$:

   a) Selecciona $i \sim (\frac{1}{15}, \frac{2}{15}, \frac{3}{15}, \frac{4}{15}, \frac{5}{15})$

   b) Calcula $L_j =$ posición del elemento seleccionado

   c) genera la nueva permutación $\sigma(i, X_1, X_2, X_3, X_4)$.

   d) Incrementa $j := j + 1$

   Implementando el algoritmo anterior, obtenemos:

```r
sigma0 <- c(1,2,3,4,5)  #permutación inicial
n <- 100
L <- I <- NULL
permutaciones <- matrix(rep(0,500),nrow=100,5)
for(j in 1:n){
i <- sample(1:5,size=1,replace = F,prob = (1:5)/15)
I[j] <- i
L[j] <- match(i,sigma0)
permutaciones[j,] <- c(i,sigma0[-match(i,sigma0)])
sigma0 <- permutaciones[j,]
}
Lhat <- mean(L)
Lhat

[1] 2.55
```

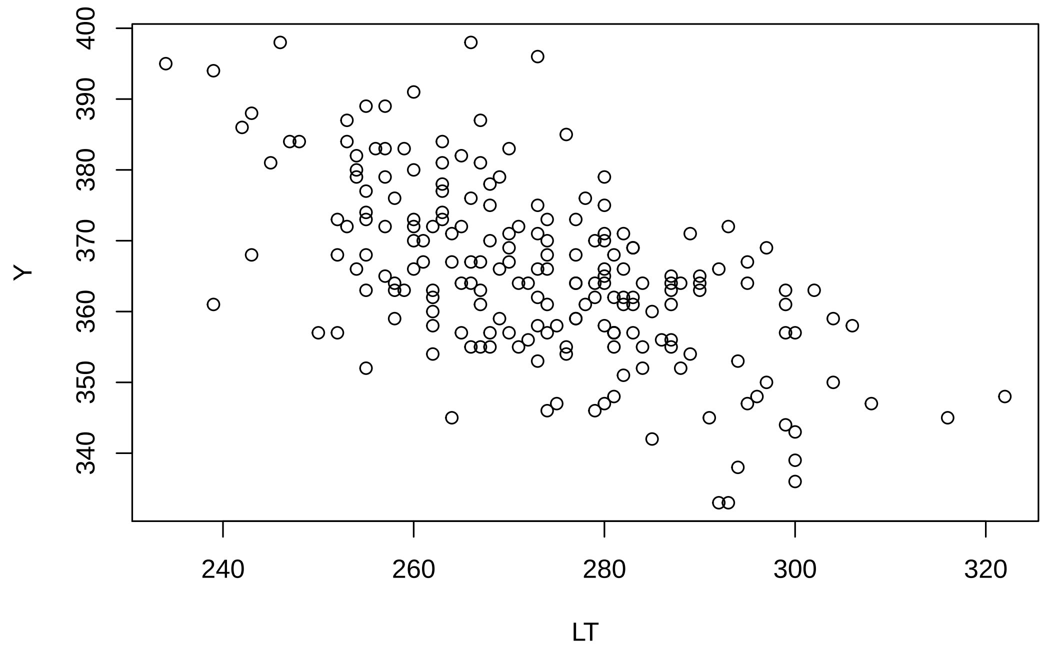
b. Notemos que $E(N_i) = Np_i$. Entonces es una cantidad fija. Si $N = 100$, entonces el vector de valores esperados es $(E(N_1), E(N_2), E(N_3), E(N_4), E(N_5) = 100(1/15, 2/15, 3/15, 4/15, 5/15) = (6.667, 13.333, 20, 26.667, 33.333)$.

c. Notemos que $E(Y/N) = \sum_{i=1}^{5} ip_i$. Entonces Y/N es el estimador del número seleccionado promedio. Ahora bien, Y es la suma promedio de los números seleccionados. Mientras mayor sea la probabilidad de selección de un número grande, será más probable que este ocupe la primera posición, haciendo L pequeña. Entonces parece que la correlación entre Y y L es negativa.

d. Con la consideración de (c), podemos llevar a cabo las simulaciones y generar parejas de puntos (Y, L) para corroborar lo propuesto.

```r
simula4 <- function(sigma0=1:5,n=100){
L <- I <- NULL
permutaciones <- matrix(rep(0,500),nrow=100,5)
for(j in 1:n){
i <- sample(1:5,size=1,replace = F,prob = (1:5)/15)
I[j] <- i
L[j] <- match(i,sigma0)
permutaciones[j,] <- c(i,sigma0[-match(i,sigma0)])
sigma0 <- permutaciones[j,]
}
return(c(LT=sum(L),Y=sum(as.vector(table(I))*(1:5))))
}

datos <- NULL
for(i in 1:200) datos <- rbind(datos,simula4())
plot(datos)
```




```
head(datos)

      LT   Y
[1,] 259 383
[2,] 262 362
[3,] 316 345
[4,] 252 357
[5,] 271 372
[6,] 255 374
```

Entonces, es posible usar Y como variable de control y reducir la varianza, ya que conocemos exactamente $E(Y) = \frac{100}{15}\sum_{i=1}^{5} i^2 = 366.6666667$.

Entonces el nuevo estimador será $\tilde{\theta} = L + \hat{\beta}(Y - 366.67)$, donde $\hat{\beta}$ es la pendiente en la línea de regresión estimada y estimamos el nuevo estimador:

```
summary(lm(LT~Y,data=as.data.frame(datos)))


Call:
lm(formula = LT ~ Y, data = as.data.frame(datos))

Residuals:
    Min      1Q  Median      3Q     Max
-37.504  -8.862  -0.164   7.399  35.639

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 550.20668   26.17915   21.02   <2e-16 ***
Y            -0.75818    0.07161  -10.59   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.65 on 198 degrees of freedom
Multiple R-squared:  0.3615,Adjusted R-squared:  0.3582
F-statistic: 112.1 on 1 and 198 DF,  p-value: < 2.2e-16

betahat <- coefficients(lm(LT~Y,data=as.data.frame(datos)))[2]
datos <- NULL
for(i in 1:1000) datos <- rbind(datos,simula4())
```

```
Warning in as.vector(table(I)) * (1:5): longitud de objeto mayor no es múltiplo de la longitud de uno menor

datos <- as.data.frame(datos)
theta <- mean(datos$LT)
tildetheta <- mean(datos$LT - betahat*(datos$Y-366.67))
c(theta,tildetheta)

[1] 273.8580 273.6487

#intervalo de confianza para theta
theta +c(-1,1)*qnorm(.975)*sd(datos$LT)/sqrt(1000)

[1] 272.9237 274.7923

#intervalo de confianza para tildetheta
tildetheta + c(-1,1)*qnorm(.975)*sd(datos$LT - betahat*(datos$Y-366.67))/sqrt(1000)

[1] 272.8873 274.4102
```

□

3. Sean X y Y dos independientes exponenciales con medias 1 y 2 respectivamente y supongan que queremos estimar $P(X+Y>4)$. ¿Cómo utilizarían condicionamiento para reducir la varianza del estimador? Digan si considerarían condicionar en X o en Y y porqué.

   Solución.

   Podemos expresar $\theta = P(X+Y>4) = E(I(X+Y>4))$ como un valor esperado. Podemos utilizar el teorema de probabilidad total para encontrar la versión condicionada:

$$
\begin{aligned}
E(I(X+Y>4)) = E(E(I(X+Y>4)|Y)) &= \int_0^\infty P(X+y>4|y) f_y(y)dy \\
&= \int_0^4 P(X>4-y|y) f_y(y)dy \\
&= \int_0^4 P(X>4-y) f_y(y)dy \\
&= E_Y(I(0<y<4)(1-F_x(4-y))) \\
&= E_Y\left(I(0<y<4)e^{-(4-y)}\right)
\end{aligned}
$$

   Entonces podemos estimar θ con $\hat{\theta} = \frac{\sum_{i=1}^n I(0<y_i<4)\exp(-(4-y_i))}{n}$ muestreando de una exponencial con media 2. Por simetría, el mismo ejercicio es para la otra condición. La que conviene para condicionar es la que tiene menor varianza condicional, ya que la varianza de este estimador depende de la media condicional.

□

4. Supongan que queremos estimar $\theta = \int_0^1 e^{x^2}\,dx$. Muestren que generar un número aleatorio u y usar el estimador $e^{u^2}(1 + e^{1-2u})/2$ es mejor que generar dos números aleatorios u_1 y u_2 y usar $(e^{u_1^2} + e^{u_2^2})/2$.

Solución.

Se proponen dos algoritmos: el algoritmo 1, sugiere calcular θ como el promedio de dos variables antitéticas:

$$\begin{aligned}\theta &= \frac{e^{u^2}(1+e^{1-2u})}{2} = \frac{e^{u^2}+e^{1-2u+u^2}}{2}\\ &= \frac{e^{u^2}+e^{(1-u)^2}}{2}\end{aligned}$$

Como la función $h(u) = e^{u^2}$ es monótona, la covarianza $Cov(h(u), h(1-u))$es negativa.

El algoritmo 2, usa dos diferentes números u_1 y u_2 independientes y hace el mismo promedio, pero en este caso, $Cov(h(u_1), h(u_2)) = 0$ por ser independientes, por lo que no se alcanza reducción de varianza.

Numéricamente, se puede mostrar definiendo dos funciones:

```
algoritmo1 <- function(n){
u <- runif(n)
theta <- (exp(u^2)+exp((1-u)^2))/2
return(list(thetahat= mean(theta),sdtheta = sd(theta)))
}

algoritmo2 <- function(n){
u1 <- runif(n); u2<- runif(n)
theta <- (exp(u1^2)+exp((u2)^2))/2
return(list(thetahat= mean(theta),sdtheta = sd(theta)))
}

algoritmo1(10000)


$thetahat
[1] 1.459695

$sdtheta
[1] 0.1653463


algoritmo2(10000)


$thetahat
[1] 1.463826

$sdtheta
[1] 0.3357278
```

□

5. Explicar cómo se pueden usar variables antitéticas en la simulación de la integral

$$\theta = \int_0^1 \int_0^1 e^{(x+y)^2}\,dx\,dy$$

¿Es claro en este caso que usando variables antitéticas es más eficiente que generando nuevos pares de variables aleatorias? Dar una justificación a su respuesta.

Solución.

Esta integral doble se puede resolver usando variables antitéticas como se describe en el siguiente algoritmo: Para $i = 1, \ldots, n$

1 Genera $U_{i1} \sim \mathcal{U}(0,1)$ y $U_{i2} \sim \mathcal{U}(0,1)$.

2 Calcula $\theta_{1i} = h(U_{i1}, U_{i2})$ y $\theta_{2i} = h(1 - U_{i1}, 1 - U_{i2})$, donde $h(x,y) = e^{(x+y)^2}$.

3 Define $\theta_i = \frac{\theta_{i1}+\theta_{i2}}{2}$

4 Obtener $\hat{\theta}_n = \frac{1}{n}\sum_{i=1}^{n} \theta_i$

De nuevo, usando el hecho de que la función $h(x,y) = e^{(x+y)^2}$ es monótona en cada uno de sus componentes, entonces $Cov(h(U_1,U_2), h(1-U_1, 1-U_2)) \leq 0$.

Numericamente, podemos hacer la siguiente función, que implementa el algoritmo propuesto.

```
algoritmo <- function(n){
                                u1 <- runif(n)
                                u2 <- runif(n)
                                th1 <- exp((u1+u2)^2)
                                th2 <- exp((2-u1-u2)^2)
                                theta <- (th1+th2)/2
                                return(list(thetahat=mean(theta),sdthetahat=sd(theta)))
                        }
algoritmo(10000)


$thetahat
[1] 4.888057

$sdthetahat
[1] 3.404378
```

□

6. En ciertas situaciones una variable aleatoria X con media conocida, se simula para obtener una estimación de $P(X \leq a)$ para alguna constante dada a. El estimador simple de una simulación para una corrida es $I = I(X \leq a)$.

   - Verificar que I y X están correlacionadas negativamente.
   - Por el inciso anterior, un intento natural de reducir la varianza es usar X como variable de control (esto es usar $Y_c = I - c(X - E(X))$. En este caso, determinar el porcentaje de reducción de varianza de Y_c sobre I es es posible (usando la mejor c si X es $\mathcal{U}(0,1)$.
   - Repetir el inciso anterior si X es exponencial con media 1.

Solución.

- Tomando por ejemplo $a = -1$, $X \sim \mathcal{N}(-2, 4)$

```
a <- -1
X <- rnorm(100,mean=-2,sd=2)
cor(ifelse(X<a,1,0),X)

[1] -0.805235
```

- Utilizando $a = 0.3$ (Olvidé dar un valor), caso Uniforme:

```
k <- 100
a <- 0.3

#piloto
x <- runif(k)
betaopt <- -lm(ifelse(x<a,1,0)~x)$coeff[2]

#the real stuff
x <- runif(1000)
y <- ifelse(x<a,1,0)
vc <- y  + betaopt*(x-0.5)
c(mean(y),sd(y)) #media y desviación estándar de I

[1] 0.2870000 0.4525879

c(mean(vc),sd(vc)) #media y desviación estándar de vc

[1] 0.2885355 0.2917832

100*(sd(y)-sd(vc))/sd(y) #Porcentaje de reducción de varianza

[1] 35.53006
```

- Para el caso de $X \sim$ exponencial, vemos que también alcanzamos reducción de varianza

```
k <- 100
a <- 0.3

#piloto
x <- rexp(k,1)
betaopt <- -lm(ifelse(x<a,1,0)~x)$coeff[2]

#the real stuff
x <- rexp(1000,1)
y <- ifelse(x<a,1,0)
vc <- y  + betaopt*(x-1)
c(mean(y),sd(y)) #media y desviación estándar de I

[1] 0.2640000 0.4410198

c(mean(vc),sd(vc)) #media y desviación estándar de vc

[1] 0.2677588 0.3808822

100*(sd(y)-sd(vc))/sd(y) #Porcentaje de reducción de varianza

[1] 13.63605
```

□

7. El número de reclamos en una aseguradora que se harán la próxima semana depende de un factor ambiental U. Si el valor de este factor es $U = u$, entonces el número de reclamos tendrá distribución Poisson con media $\frac{15}{0.5+u}$. Suponiendo que $U \sim \mathcal{U}(0, 1)$, sea p la probabilidad de que habrá al menos 20 reclamos la siguiente semana.

   - Explicar como obtener una simulación cruda de p.

- Desarrollar un estimador de simulación eficiente usando esperanza condicional junto con una variable de control
- Desarrollar un estimador de simulación eficiente usando esperanza condicional y variables antitéticas.
- Escriban un programa para determinar las varianzas de los incisos anteriores.

Solución.

- Para obtener una simulación cruda de p, notemos que podemos escribir $p = P(X > 20) = 1 - P(X \leq 20)$ y entonces:

$$\begin{aligned} P(X \leq 20) &= \int_0^1 P(X \leq 20|u)du \\ &= \int_0^1 \sum_{i=0}^{20} e^{-\frac{15}{0.5+u}} \frac{(15/(0.5+u))^i}{i!} du \\ &= \sum_{i=0}^{20} \frac{1}{i!} \int_0^1 e^{-\frac{15}{0.5+u}} (15/(0.5+u))^i du \end{aligned}$$

  Entonces podemos estimar cada integral $g(i) = \int_0^1 e^{-\frac{15}{0.5+u}} (15/(0.5+u))^i du$ para cada $i = 0, 1, \ldots, 20$ usando las propias uniformes y evaluando la integral.

```
g <- NULL
for (i in 0:20){
u <-runif(10000)
y <- exp(-15/(0.5+u))*(15/(0.5+u))^i
g[i+1] <- mean(y)
}
p <- 1 - sum(g*1/factorial(0:20))
p

[1] 0.2521201
```

- Podemos usar la dependencia entre u y $X|u$ para crear la variable de control.

```
k <- 100
xu <- NULL
#piloto
u <- runif(k)
for(i in 1:length(u)) xu[i] <-rpois(1,15/(0.5+u[i]))

betaopt <- -lm(xu ~ u)$coeff[2]
u <- runif(10000)
#the real stuff
for(i in 1:length(u)) x[i] <-rpois(1,15/(0.5+u[i]))
vc <- x + betaopt*(u-0.5)
p <- 1- sum(vc<=20)/1000
p

[1] -7.034
```

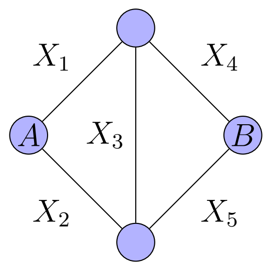
- Por último, para el caso de variables antitéticas,

```
g <- NULL
for (i in 0:20){
u1 <-runif(5000)
y1 <- exp(-15/(0.5+u))*(15/(0.5+u))^i
y2 <- exp(-15/(0.5+(1-u)))*(15/(0.5+(1-u)))^i
g[i+1] <- (mean(y1)+mean(y2))/2
}
p <- 1 - sum(g*1/factorial(0:20))
p

[1] 0.2505657
```

□

8. Consideren la siguiente gráfica, representando una red puente:



Supongan que queremos estimar la longitud esperada l de la ruta más corta entre los nodos A y B, donde las longitudes de los arcos son variables aleatorias $X_1, \ldots, X_5$. Entonces tenemos que $l = E(H(\mathbf{X})$, donde

$$H(\mathbf{X}) = \text{mín}\{X_1 + X_4, X_1 + X_3 + X_5, X_2 + X_3 + X_4, X_2 + X_5\}$$

Noten que $H(\mathbf{x})$ es no decreciente en cada componente del vector $\mathbf{x}$. Supongan que las longitudes son independientes y $X_i \sim \mathcal{U}(0, a_i)$, con $\mathbf{a} = (1, 2, 3, 1, 2)$. Escribiendo $X_i = a_i U_i$ se puede restablecer el problema como la estimación de $l = \mathrm{E}(h(\mathbf{U}))$ con $h(\mathbf{U}) = H(a_1 U_1, \ldots, a_5 U_5)$.

- Obtener un estimador crudo de MonteCarlo para l.
- Obtener un estimador usando variabes antitéticas
- Obtener un estimador usando variables de control.
- Obtener un estimador usando condicionamiento.

En todos los casos anteriores, calcular la reducción de varianza obtenida y determinar el mejor método.

Solución.

- Por simulación directa:

```
a <- c(1,2,3,1,2)
H <- function(x)min(x[1]+x[4],x[1]+x[3]+x[5],x[2]+x[3]+x[4],x[2]+x[5])
h <- function(x)H(a*x)

E <- NULL
n <- 10000
for(i in 1:n) E[i]<-h(runif(5))
c(mean(E),sd(E)/sqrt(n))

[1] 0.933679176 0.003923846
```

- Para variables antitéticas:

```
E1 <- E2 <- NULL
for(i in 1:n/2){
u <- runif(5)
E1[i] <- h(u)
E2[i] <- h(1-u)
}
E <- (E1+E2)/2
c(mean(E),sd(E)/sqrt(n))

[1] 0.933413805 0.001320706
```

- Para variable de control, podemos considerar por ejemplo $Y = \text{mín}\{X_1 + X_4, X_2 + X_5\}$. Esta variable es particularmente conveniente para los parámetros dados, ya que con alta probabilidad la ruta más corta tendrá longitud igual a Y. Con un poco de cálculos, podemos obtener que $E(Y) = 15/16 = 0.9375$. Entonces

```
Hc <- function(x)min(x[1]+x[4],x[2]+x[5])
hc <- function(x)Hc(a*x)

u <- matrix(runif(n*5),nrow=n,ncol=5)
Y <- apply(u,1,h)
Yc <- apply(u,1,hc)
a <- cor(Y,Yc)
E <- mean(Y-a*(Yc-15/16))
c(E,sd(Y-a*(Yc-15/16))/sqrt(n))

[1] 0.9294531982 0.0005273668
```

- Para condicionamiento, definimos $Z_1 = \text{mín}\{X_4, X_3 + X_5\}$, $Z_2 = \text{mín}\{X_5, X_3 + X_4\}$, entonces $Y_1 = X_1 + Z_1$, $Y_2 = X_2 + Z_2$ y $Y = H(X)$ se puede escribir como $Y = \text{mín}\{Y_1, Y_2\}$ donde condicional a (Z_1, Z_2), (Y_1, Y_2) es uniforme en el rectángulo $[z_1, z_1 + 1] \times [z_2, z_2 + 2]$. La esperanza condicional de Y dado (Z_1, Z_2) se puede evaluar exactamente. Aquí sólo dejaré la idea marcada.

□

9. Sea S la suma de los resultados de lanzar 100 veces un dado honesto. Usen la desigualdad de Chebyshev para acotar $P(S \geq 380)$.

Solución.

La desigualdad de Chebyshev es la siguiente:

$$P(|X - \mu| \geq \epsilon) \leq \frac{\sigma^2}{\epsilon^2}$$

En este ejercicio, S_n es la suma de los resultados de lanzar un dado honesto, por lo que $S_n = \sum_{i=1}^{n} X_i$ donde $P(X_i = j) = 1/6$ para $j = 1, 2, \ldots, 6$, es decir, es una uniforme discreta en $\{1, 2, \ldots, 6\}$. Como $\mathrm{E}(X_i) = 3.5$ y $\mathrm{Var}(X_i) = 35/12 = 2.91667$, entonces $\mathrm{E}(S_n) = 3.5n$ y $\mathrm{Var}(S_n) = 2.92n$. Aplicando la desigualdad de Chebyshev con estas características, obtenemos:

$$P(|S_{100} - 350| \geq \epsilon) \leq \frac{292}{\epsilon^2}$$

Por lo que

$$P(S_{100} > 380) = P(S_{100} - 350 \geq 30) = P(|S_{100} - 350| \geq 30) \leq \frac{292}{30^2}$$

```
292/30^2

[1] 0.3244444
```

Así que $P(S_{100} > 380) \leq 0.3244444$.

□

10. Estimar usando MC crudo: $\int_{-\infty}^{\infty} \log(x^2) e^{-x^2}\, dx$ y aplicar dos técnicas de reducción de varianza a esta integral. Calcular la reducción lograda con cada método.

Solución.

Podemos observar que:

$$\begin{aligned}\int_{-\infty}^{\infty} \log(x^2) e^{-x^2}\, dx &= \int_{-\infty}^{\infty} \log(x^2) \frac{\sqrt{\pi}}{\sqrt{2\pi(1/2)}} e^{-x^2/(2(1/2))}\, dx \\ &= \mathrm{E}(\sqrt{\pi}\log(X^2))\end{aligned}$$

donde X es una normal con media 0 y varianza $1/2$.

Aplicando MC crudo:

```
N <- 10000
X <- rnorm(N,sd=1/sqrt(2))
theta <- sqrt(pi)*log(X^2)
thetahat <- mean(theta)
thetahat

[1] -3.465263

var1 <- var(theta)
var1

[1] 15.33496
```

- Aplicando variables antitéticas, tenemos que generar dos variables correlacionadas negativamente:

```
library(MASS)
M <- matrix(c(1/2,-0.99/2,-0.99/2,1/2),2,2)
X <- mvrnorm(N/2,rep(0,2),M)
cor(X)

           [,1]       [,2]
[1,]  1.0000000 -0.9895052
[2,] -0.9895052  1.0000000

theta1 <- sqrt(pi)*log(X[,1]^2)
theta2 <- sqrt(pi)*log(X[,2]^2)
theta <- (theta1 + theta2)/2
thetahat1 <- mean(theta)
thetahat1

[1] -3.483833

var2 <- var(theta)
var2

[1] 14.21368
```

EL porcentaje de reducción de varianza es:

```
100*(var1-var2)/var1

[1] 7.311929
```

- Utilizando muestreo de importancia, podemos ver que por simetría, $\int_{-\infty}^{\infty} \log(x^2)e^{-x^2}dx = 2\int_0^{\infty} \log(x^2)e^{-x^2}dx$. Podemos transformar tomando $Z = X^2$ y se obtiene:

$$2\int_0^{\infty} \log(x^2)e^{-x^2}dx = 2\int_0^{\infty} \log(z)e^{-z}\frac{dz}{2\sqrt{z}} = \int_0^{\infty} \frac{\log(z)}{\sqrt{z}}e^{-z}dz = \mathrm{E}\left(\frac{\log(z)}{\sqrt{z}}\right)$$

donde $Z \sim \exp(1)$.

```
h <- function(z)log(z)/sqrt(z)
z <- rexp(N,1)
thetahat <- mean(h(z))
thetahat

[1] -3.119699

var3 <- var(h(z))
var3

[1] 334.2219
```

EL porcentaje de reducción de varianza es:

```
100*(var1-var3)/var1

[1] -2079.476
```

En este último caso vemos que la elección de la función de importancia, no es la adecuada, pues no hay reducción de varianza. Una segunda opción es tomar

$$\int_0^{\infty} \log(z)e^{-z}\frac{dz}{\sqrt{z}} = \int_0^{\infty} \log(z)z^{-1/2}e^{-z}dz = \Gamma(1/2)\mathrm{E}\left(\log(z)\right)$$

muestreando de una $\mathcal{G}\left(1/2, 1\right)$.

```r
z <- rgamma(N,1/2,1)
thetahat <- mean(log(z)*gamma(1/2))
thetahat
```

```
[1] -3.477792
```

```r
var4 <- var(log(z)*gamma(1/2))
var4
```

```
[1] 15.08321
```

```r
# % reducción de varianza
100*(var1-var4)/var1
```

```
[1] 1.641725
```

Esta densidad de importancia es mejor que la anterior.

□