Simulación
21 de mayo de 2018
2018-I

ITAM
Jorge de la Vega

Tarea 4.

La fecha de entrega es el **30 de abril de 2018**.

Lectura

- Salmon, F. (2009) Recipe for Disaster: The Formula That Killed Wall Street, Wired
- Arturo Ederly: Cópulas y variables aleatorias, una introducción
- Robert & Casella Caps. 3 y 4.
- Dagpunar Cap.5

Problemas

1. Obtener una muestra aleatoria de 5,000 observaciones del vector $X = (X_1, X_2, X_3, X_4)$ donde $X_1 \sim \mathcal{N}(4, 9)$, $X_2 \sim$ **Bernoulli** (0.6), $X_3 \sim Beta(2, 3)$ y $X_4 \sim Gamma(3, 2)$. Considerar la siguiente estructura de correlación entre las variables: $cor(X_1, X_3) = -0.7$, $cor(X_2, X_4) = 0.4$, $cor(X_1, X_4) = 0.5$ y $cor(X_2, X_3) = 0.2$. En otro caso consideramos que las variables son independientes. Hacer los histogramas de las funciones marginales y corroborar que tienen la distribución considerada.

   Solución.

   Ya que contamos con una matriz de correlación, podemos utilizar una cópula Gaussiana para generar la muestra solicitada. Como los datos ya están dados en términos de correlación, la matriz Σ está dada por

$$\Sigma = \begin{pmatrix} 1 & 0 & -0.7 & 0.5 \\ 0 & 1 & 0.2 & 0.4 \\ -0.7 & 0.2 & 1 & 0 \\ 0.5 & 0.4 & 0 & 1 \end{pmatrix}$$

   Entonces el procedimiento es:

   a) Obtener un vector $Z \sim N(0, \Sigma)$

   b) obtener un vector $U \sim (\Phi(Z_1), \Phi(Z_2), \Phi(Z_3), \Phi(Z_4))$

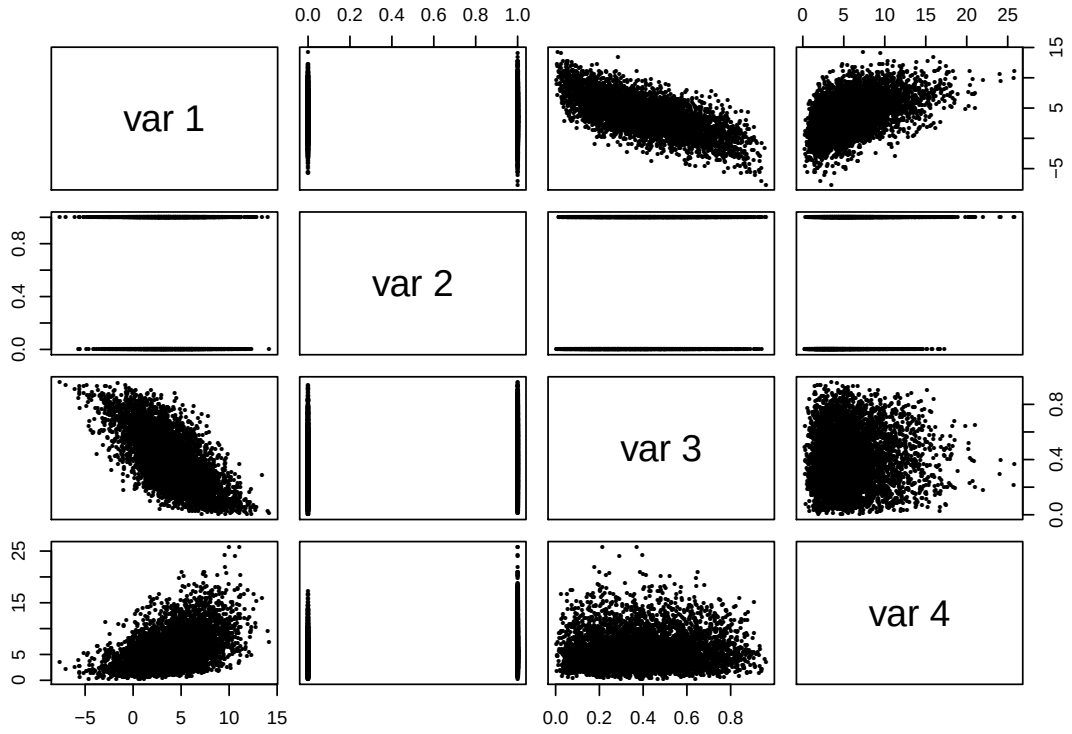
c) Obtener el vector $W \sim (F_1^{-1}(U_1), F_2^{-1}(U_2), F_3^{-1}(U_2), F_4^{-1}(U_4))$

Un ejercicio similar a este vimos en uno de los laboratorios.

```
library(copula)
cn4 <- normalCopula(c(0,-0.7,0.5,0.2,0.4,0),dim=4,dispstr = "un")
set.seed(100) #fija una semilla
U <- rCopula(5000,cn4) #Genera la muestra aleatoria
W <- cbind(qnorm(U[,1],mean=4,sd=3),
qbinom(U[,2],size=1,prob=0.6),
qbeta(U[,3],2,3),
qgamma(U[,4],shape=3,scale=2))
head(W)
```

```
         [,1] [,2]      [,3]      [,4]
[1,] 3.555827    1 0.4295056  7.964706
[2,] 5.618034    1 0.2758444  8.101240
[3,] 1.838498    0 0.4750418  4.665784
[4,] 3.274686    1 0.4470294  5.595060
[5,] 6.085712    1 0.2766029 15.324181
[6,] 3.190979    1 0.5095977  8.232991
```

```
pairs(W,pch=16,cex=0.5)
```

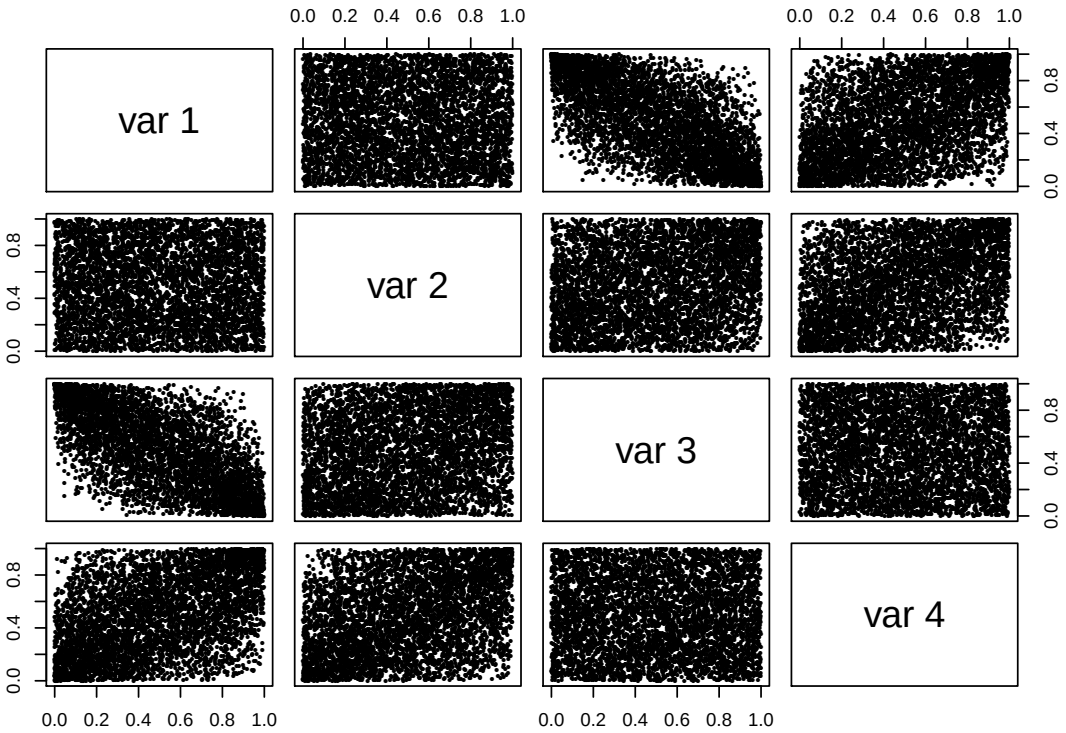



```
round(cor(W),2)
```

```
      [,1]  [,2]  [,3] [,4]
[1,]  1.00 -0.01 -0.71 0.48
[2,] -0.01  1.00  0.18 0.29
[3,] -0.71  0.18  1.00 0.00
[4,]  0.48  0.29  0.00 1.00
```

Los datos simulados de la cópula son los siguientes:

```
pairs(U, pch=16, cex=0.5)
```

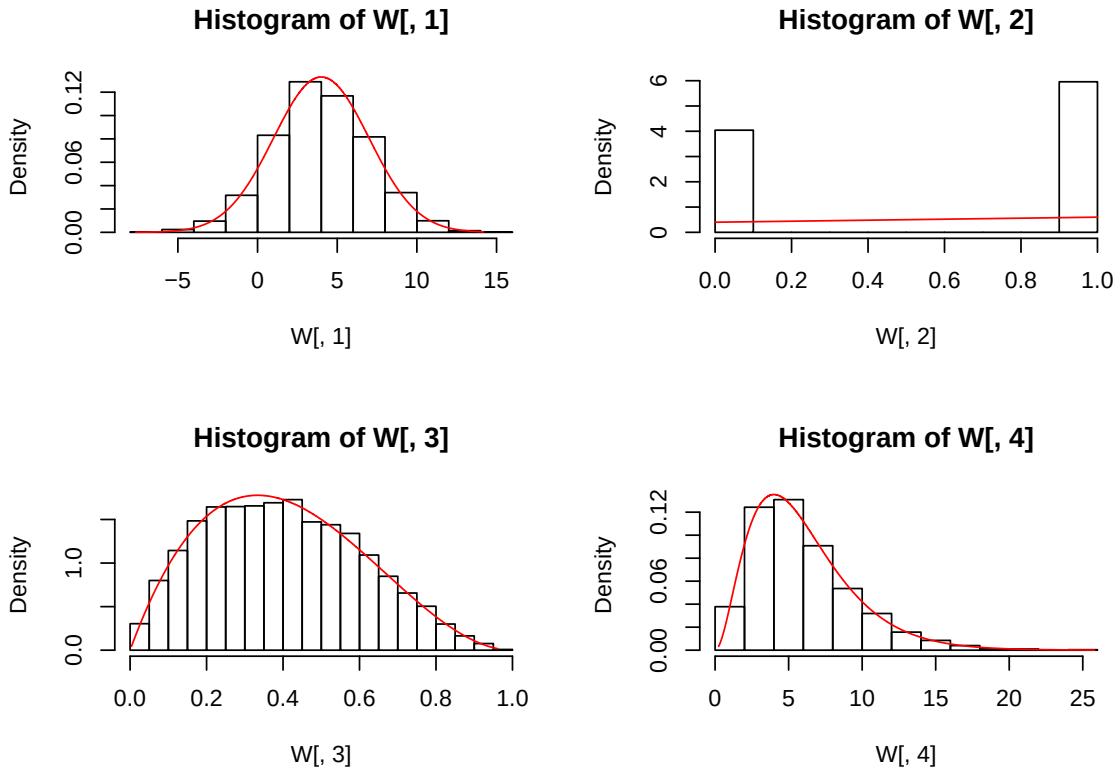



```
round(cor(U,method = "kendall"),2)
```

```
      [,1]  [,2]  [,3] [,4]
[1,]  1.00 -0.01 -0.51 0.33
[2,] -0.01  1.00  0.13 0.25
[3,] -0.51  0.13  1.00 0.00
[4,]  0.33  0.25  0.00 1.00
```

Ahora procedemos al paso 4, para generar nuestro vector y hacemos los histogramas para ver si tienen el comportamiento deseado, y se muestra un ejemplo de los valores generados:

```
#Grafica los histogramas y agrega densidades con las distribuciones deseadas para ver
#la aproximación
par(mfrow=c(2,2))
hist(W[,1],probability = T); points(sort(W[,1]),dnorm(sort(W[,1]),4,3),type="l",col="red")
hist(W[,2],probability = T); points(sort(W[,2]),dbinom(sort(W[,2]),1,0.6),type="l",col="red")
hist(W[,3],probability = T); points(sort(W[,3]),dbeta(sort(W[,3]),2,3),type="l",col="red")
hist(W[,4],probability = T); points(sort(W[,4]),dgamma(sort(W[,4]),shape=3,scale=2),type="l",col="red")
```




```
cor(W,method="kendall")
```

```
             [,1]        [,2]         [,3]         [,4]
[1,]  1.00000000 -0.01048706 -0.505808202  0.331958872
[2,] -0.01048706  1.00000000  0.142987645  0.251824288
[3,] -0.50580820  0.14298764  1.000000000 -0.001945189
[4,]  0.33195887  0.25182429 -0.001945189  1.000000000
```

□

2. La τ de Kendall entre X y Y es 0.55. Tanto X como Y son positivas. ¿Cuál es la τ entre X y $1/Y$? ¿Cuál es la τ de $1/X$ y $1/Y$?

   Solución.

   Hay que recordar que la τ de Kendall es una estadística basada en los rangos de la variable aleatoria (expresados en términos de las concordancias y discordancias de las observaciones) y que es invariante ante transformaciones monótonas. Entonces, partiendo de que las variables son positivas, y como $1/Y$ es monótona decreciente, el valor de la τ se mantiene, pero cambia el signo. En el caso en el que se cambian ambas variables, el valor de la estadística queda el mismo. Lo podemos comprobar con una pequeña simulación

```
X <- runif(100)
Y <- X + runif(100)
cor(X,Y,method="kendal")
```

```
[1] 0.52
```

```
cor(X,1/Y,method="kendall")
```

```
[1] -0.52
```

```
cor(1/X,1/Y,method="kendall")
```

```
[1] 0.52
```

□

3. Mostrar que cuando $\theta \to \infty$, $C^{Fr}(u_1, u_2) \to min\{u_1, u_2\}$, donde C^{Fr} es la cópula de Frank.

   Solución.

   Cuando $\theta \to \infty$, $e^{-\theta} - 1 \approx -1$, por lo que

$$\begin{aligned} C^{Fr}(u,v) &\approx -\frac{1}{\theta}\log[1-(e^{-\theta u}-1)((e^{-\theta v}-1)] \\ &= -\frac{1}{\theta}\log[1-(e^{-\theta(u+v)}-e^{-\theta u}-e^{-\theta v}+1)] \\ &= -\frac{1}{\theta}\log[-e^{-\theta(u+v)}+e^{-\theta u}+e^{-\theta v}] \\ &= -\frac{1}{\theta}\log[e^{-\theta u}(-e^{-\theta v}+1+e^{-\theta(v-u)})] \quad \text{si } u<v \\ &= u-\frac{1}{\theta}\log[1-e^{-\theta v}+e^{-\theta(v-u)}] \\ &\to u \quad \text{cuando } \theta \to \infty \end{aligned}$$

   Y el límite es simétrico, por lo que $\theta \to \infty$, $C^{Fr}(u,v) \to min\{u,v\}$.

□

4. - Construyan un vector de 100 números crecientes y espaciados regularmente entre 0.1 y 20. Llámenlo `SIG2` . Ahora construyan otro vector de longitud 21 empezando en -1 y terminando en 1. Llámenlo `RHO`.
   - Para cada entrada σ^2 de `SIG2` y cada entrada de `RHO`:
     - Generar una muestra de tamaño $N = 500$ de una distribución bivariada normal $Z = (X, Y)$ donde $X \sim \mathcal{N}(0,1)$ y $Y \sim \mathcal{N}(0, \sigma^2)$ y el coeficiente de correlación de X y Y es ρ. Z es una matriz de dimensiones 500×2.

- Crear una matriz de 500×2, llámenlo `EXPZ`, con las exponenciales de las entradas de Z. ¿Qué distribución tienen estas variables transformadas?
- Calculen el coeficiente de correlación, $\tilde{\rho}$ de las columnas de `EXPZ`. Grafiquen los puntos $(\sigma^2, \tilde{\rho})$ y comenten sobre lo que obtuvieron.

Solución.

Para el primer inciso,

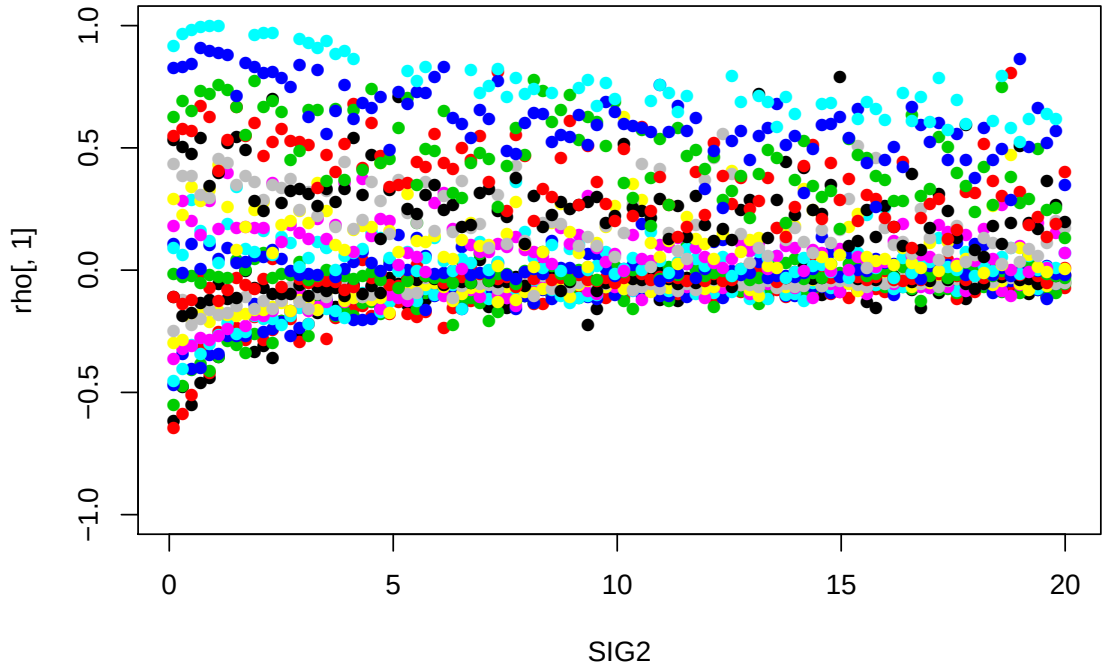
```
SIG2 <- seq(0.1,20,length=100)
RHO <- seq(-1,1,length=21)
```

Para el segundo inciso, generamos las normales utilizando el método de Box-Müller, que de construye con la siguiente función (pueden usar cualquier función que genere normales). Z tiene una distribución lognormal.

```
normalBM <- function(n){
#genera una muestra de pares de normales independientes de tamaño n.
u1 <- runif(n)
u2 <- runif(n)
R  <- sqrt(-2*log(u1))
z1 <- R*cos(2*pi*u2)
z2 <- R*sin(2*pi*u2)
return(cbind(z1,z2))
}
```

Una vez obtenidos los pares de variables normales, multiplicamos por la matriz B tal que $BB' = \Sigma$, donde $\Sigma = \begin{pmatrix} 1, \rho\sigma \\ \rho\sigma, \sigma^2 \end{pmatrix}$.

```
rho <- matrix(numeric(),nrow=100,ncol=21)

suppressWarnings(
for(i in SIG2){
for(j in RHO){
Sigma <- matrix(c(1,sqrt(i)*j,sqrt(i)*j,i),nrow=2,byrow=T)
e <- eigen(Sigma)
B <- e$vectors %*% diag(sqrt(e$values)) %*% t(e$vectors)
ZEXP <- exp(normalBM(500) %*% B)
rho[match(i,SIG2),match(j,RHO)] <- cor(ZEXP[,1],ZEXP[,2])
}
}
) #Algunos puntos no están definidos, para no listar todos los casos.

plot(SIG2,rho[,1],pch=16,col=1,ylim=c(-1,1))

for(i in 2:21)points(SIG2,rho[,i],pch=16,col=i)
```



Lo que se puede observar de la gráfica, es que la transformación no es lineal para los valores de la correlación.

□

5. Consideren la cópula de Clayton. Mostrar que converge a la cópula de comonotonicidad cuando $\theta \to \infty$. [Hint: usen la regla de l'Hópital considerando que la cópula de Clayton se puede escribir como $exp\{log(u_1^{-\theta} + u_2^{-\theta} - 1)/\theta\}$ para θ positivo.]

Solución.

Sea $u < v$ para $u, v \in (0,1)$. Entonces $log(u) > log(v)$ y por lo tanto $\theta log(v) - \theta log(u) < 0$. Siguiendo el hint, notemos que

$$\begin{aligned} log\left(u^{-\theta} + v^{-\theta} - 1\right) &= log\left(e^{-\theta log(u)}(1 + e^{\theta(log(v)-log(u))} + e^{\theta log(u)})\right) \\ &= -\theta log(u) + log\left(1 + e^{\theta(log(v)-log(u))} + e^{\theta log(u)}\right) \end{aligned}$$

Así que

$$\begin{aligned} \exp\left(\frac{1}{\theta} log\left(u^{-\theta} + v^{-\theta} - 1\right)\right) &= \exp(-log(u)) \exp\left(\frac{1}{\theta} log\left(1 + e^{\theta(logv - logu)} + e^{\theta logu}\right)\right) \\ &= u \exp\left(\frac{-log(1 + e^{\theta k} + e^{\theta m})}{\theta}\right) \end{aligned}$$

donde $k = log(u) - log(v) < 0$ y $m = log(u) < 0$ Si aplicamos l'Hôpital a este cociente, tenemos

$$lim_{\theta\to\infty}\frac{-log(1+e^{\theta k}+e^{\theta m})}{\theta} = lim_{\theta\to\infty}\frac{ke^{\theta k}+me^{\theta m}}{1+e^{\theta k}+e^{\theta m}} = 0$$

Así que $u\exp\left(\frac{-log(1+e^{\theta k}+e^{\theta m})}{\theta}\right) \to ue^0 = u$. Como el resultado es simétrico en u y en v, se tiene que

$$lim_{\theta\to\infty}C(u,v)^C = min\{u,v\}$$

□

6. Supongan que tienen dos vectores de datos $(x_1,\ldots,x_n)$ y $(y_1,\ldots,y_n)$. Entonces la cópula empírica es la función $C:[0,1]\times[0,1]\to[0,1]$ definida por

$$C(u,v) = \frac{1}{n}\sum_{j=1}^{n} I\left(\frac{r_j}{n+1}\leq u, \frac{s_j}{n+1}\leq v\right)$$

donde $(r_1,\ldots,r_n)$ y $(s_1,\ldots,s_n)$ denotan los vectores de rangos de x y y respectivamente.

Escriban una función *vectorizada* (esto es, que se pueda vauar en vectores) llamada `empCopula` que tome cuatro argumentos `u, v, xVec` y `yVec`. Pueden suponer que los valores `u,v` están en $[0,1]$ y que `xVec` y `yVec` son vectores numéricos que tienen la misma longitud (no vacios).

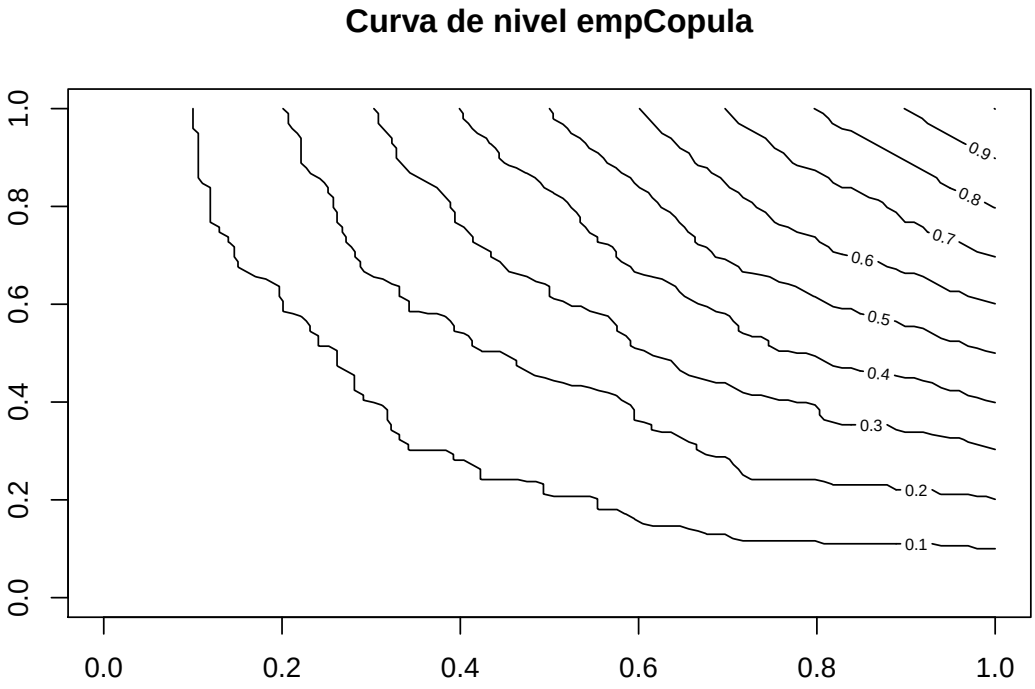
Solución.

La función que se pide es la siguiente. Es muy simple pero no está *vectorizada*, por lo que no puedo aplicar la función `outer` para poder generar un grid y hacer una gráfica.

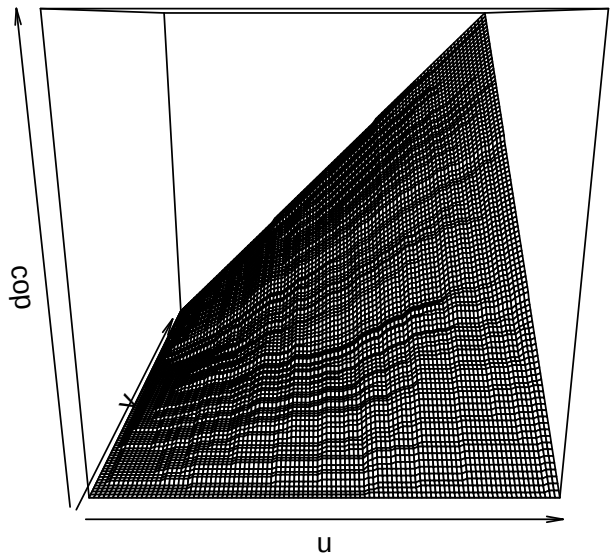
```
empCopula <- function(u,v,xVec,yVec){
#esta función calcula la cópua empírica de un par de vectores aleatorios
#que tienen la misma longitud.
n <- length(xVec)
rx <- rank(xVec)/(n+1)
ry <- rank(yVec)/(n+1)
return(mean((rx<=u) & (ry<=v)))
}
```

Para hacer la gráfica, genero manualmente el grid para la función

```
k <- 100
u <- v <- seq(0, 1, length = k)
xVec <- runif(200)
yVec <- rnorm(200)
cop <- matrix(numeric(),nrow=k,ncol=k)

#genera un grid para graficar:
for(uu in u)
for(vv in v)
cop[which(uu==u),which(vv==v)] <- empCopula(uu,vv,xVec,yVec)


contour(u, v, cop, main = "Curva de nivel empCopula" )
```




```
persp(u,v,cop, main= "empCopula")
```



□

7. la cópula Farlie-Gumbel-Morgenstern es $C(u,v) = uv[1+\alpha(1-u)(1-v)]$ para $|\alpha| \leq 1$. Mostrar que la densidad conjunta correspondiente $\partial^2 C(u,v)/\partial u \partial v$ es no negativa. Mostrar que C tiene marginales uniformes en $(0,1)$. Encontrar el coeficiente de correlación de Spearman y la tau de Kendall.

   Solución.

   Este ejercicio tiene tres partes:

   - Para obtener la densidad conjunta, derivamos con respecto a cada una de las variables:

$$\begin{aligned} \frac{\partial C}{\partial u} &= v + \alpha v(1-v)[1-2u] \\ \frac{\partial^2 C}{\partial u \partial v} &= 1 + \alpha[1-2u][1-2v] \end{aligned}$$

     Como $1-2u \leq 0$ y $1-2v \leq 0$, para $0 \leq u, v \leq 1$ entonces el producto toma el valor mínimo en $u = 1$ y $v = 0$ o $u = 0$ y $v = 1$. En ese caso, para que el producto sea no negativo basta que $\alpha \leq 1$, lo cual siempre se cumple.

- Para ver que C tiene marginales uniformes, basta con evaluar C en los siguientes valores $C(1,v)$ y $C(u,1)$. Como la función es simétrica en los dos valores, basta hacer $C(1,v) = v + \alpha v(0)(1-v) = v$. Por lo tanto, las marginales son uniformes.
- Para esta parte, tenemos que resolver las ecuaciones:

$$\tau = 4\int_0^1\int_0^1 C(u,v)dC(u,v) - 1$$

para la τ de Kendall y

$$\rho_S = 12\int_0^1\int_0^1 C(u,v)dudv - 3$$

para la ρ de Spearman.
Haciendo primero la fórmula de la τ de Kendall, tenemos que:

$$\tau = 4\int_0^1\int_0^1 (uv + \alpha uv(1-u)(1-v)(1+\alpha(1-2u)(1-2v))dudv - 1$$

Haciendo el producto, cada una de las integrales dobles es fácil de resolver todas son polinomiales y simétricas. Por ejemplo:

$$\int_0^1\int_0^1 uvdudv = 1/4$$

$$\int_0^1\int_0^1 \alpha uv(1-2u)(1-2v)dudv = \alpha/36$$

$$\int_0^1\int_0^1 \alpha uv(1-u)(1-v)dudv = \alpha/36$$

$$\int_0^1\int_0^1 \alpha^2 uv(1-u)(1-v)(1-2u)(1-2v)dudv = 0$$

Entonces $\tau = 4(1/4 + \alpha/18) - 1 = 1 + 2\alpha/9 - 1 = 2\alpha/9$. Para el segundo caso,

$$\rho_S = 12\int_0^1\int_0^1 uv(1+\alpha(1-u)(1-v))dudv - 3$$

Expandiendo los términos, nos quedan de nuevo integrales que ya hicimos, por lo que

$$\rho_S = 12(1/4 + \alpha/36) - 3 = 3 + 12\alpha/36 - 3 = \alpha/3$$

□

8. Este es un ejercicio de calibración de las cópulas utilizando correlaciones de rangos. Supongan que una muestra produce un estimado de la τ de Kendall de 0.2. ¿Qué parámetro debe usarse para

   i. la cópula normal,

   ii. la cópula de Gumbel,

   iii. la cópula de Calyton?

   Solución.

   i. La correlación en la cópula normal – y de hecho en cualquier elíptica – debe ser establecida igual a $\rho = \sin(0.1\pi) = 0.309$.

   ii. En la cópula de Gumbel se establece $\delta = (1 - 0.2)^{-1} = 1.25$

   iii. En la cópula de Clayton establecemos $\alpha = 0.4(1 - 0.2)^{-1} = 0.5$.

   □

9. Usen la función `normalCopula` del paquete `copula` para crear una cópula gaussiana bidimensional con un parámetro de 0.9. Luego creen otra cópula gaussiana con parámetro de 0.2 y describan la estructura de ambas cópulas (diferencias y semejanzas).

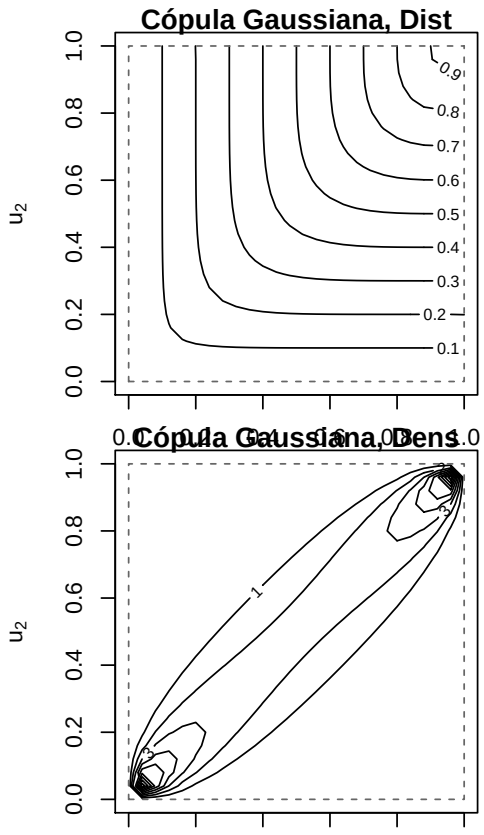
   Solución.

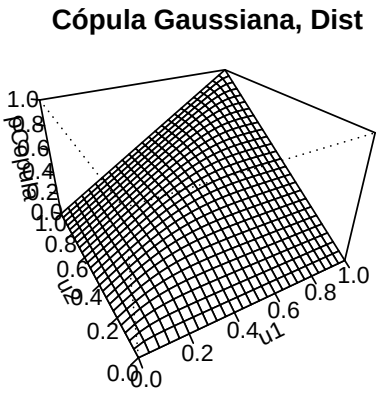
   Las siguientes gráficas ayudan a entender las diferencias entre las diferentes cópulas. Claramente la cópula con el coeficiente de correlación más alto es la que relaciona linealmente mejor a las dos variables. La dependencia lineal en la cópula gaussiana se relaciona directamente con el parámetro.

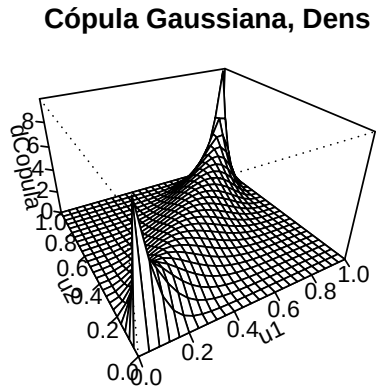
```
library(copula)
normal_0.9 <- normalCopula(param = 0.9, dim = 2)
str(normal_0.9) #despliega las características de la cópula


Formal class 'normalCopula' [package "copula"] with 8 slots
  ..@ dispstr     : chr "ex"
  ..@ getRho      :function (obj)
  ..@ dimension   : int 2
  ..@ parameters  : num 0.9
  ..@ param.names : chr "rho.1"
  ..@ param.lowbnd: num -1
  ..@ param.upbnd : num 1
  ..@ fullname    : chr "<deprecated slot>"


normal_0.2 <- normalCopula(param = 0.2, dim = 2)
str(normal_0.2 ) #despliega las características de la cópula
```

```
Formal class 'normalCopula' [package "copula"] with 8 slots
  ..@ dispstr     : chr "ex"
  ..@ getRho      :function (obj)
  ..@ dimension   : int 2
  ..@ parameters  : num 0.2
  ..@ param.names : chr "rho.1"
  ..@ param.lowbnd: num -1
  ..@ param.upbnd : num 1
  ..@ fullname    : chr "<deprecated slot>"
```

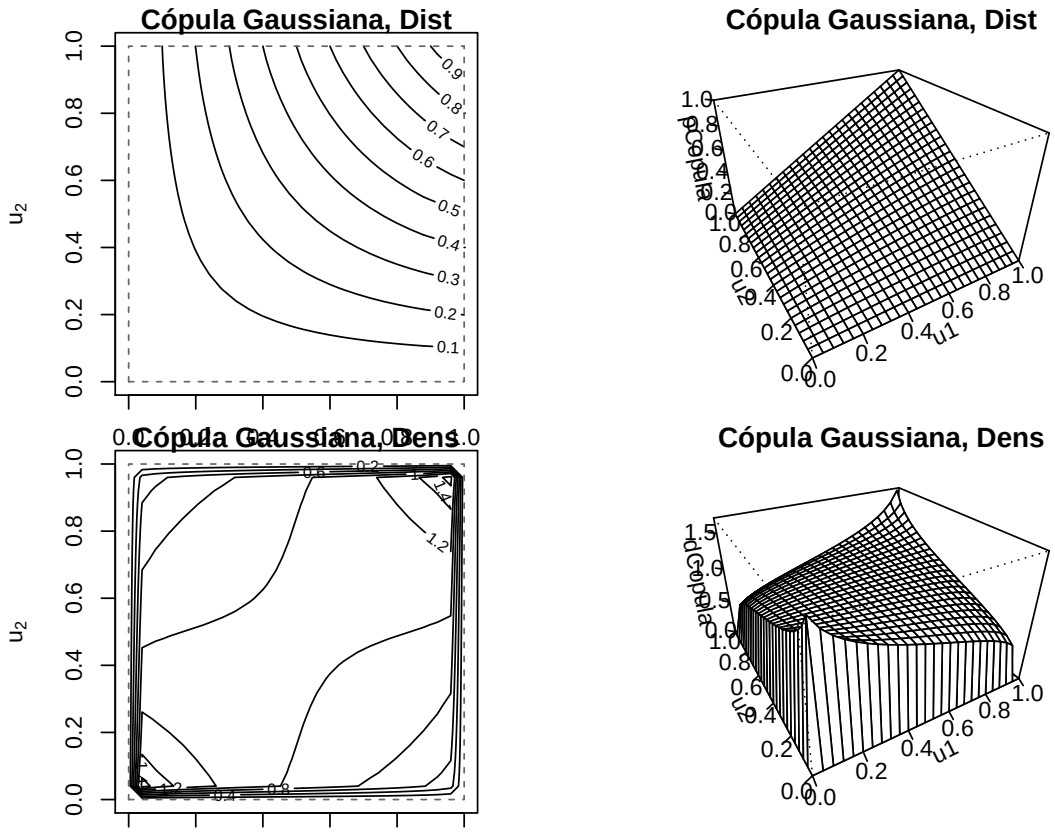
```
par(mar=c(1,2,1,1))
par(mfrow = c(2,2), pty="s")

contour(normal_0.9, pCopula, main = "Cópula Gaussiana, Dist" )
persp(normal_0.9, pCopula, main = "Cópula Gaussiana, Dist")
contour(normal_0.9, dCopula, main = "Cópula Gaussiana, Dens" )
persp(normal_0.9, dCopula, main = "Cópula Gaussiana, Dens")
```








```
contour(normal_0.2, pCopula, main = "Cópula Gaussiana, Dist" )
persp(normal_0.2, pCopula, main = "Cópula Gaussiana, Dist")
contour(normal_0.2, dCopula, main = "Cópula Gaussiana, Dens" )
persp(normal_0.2, dCopula, main = "Cópula Gaussiana, Dens")
```

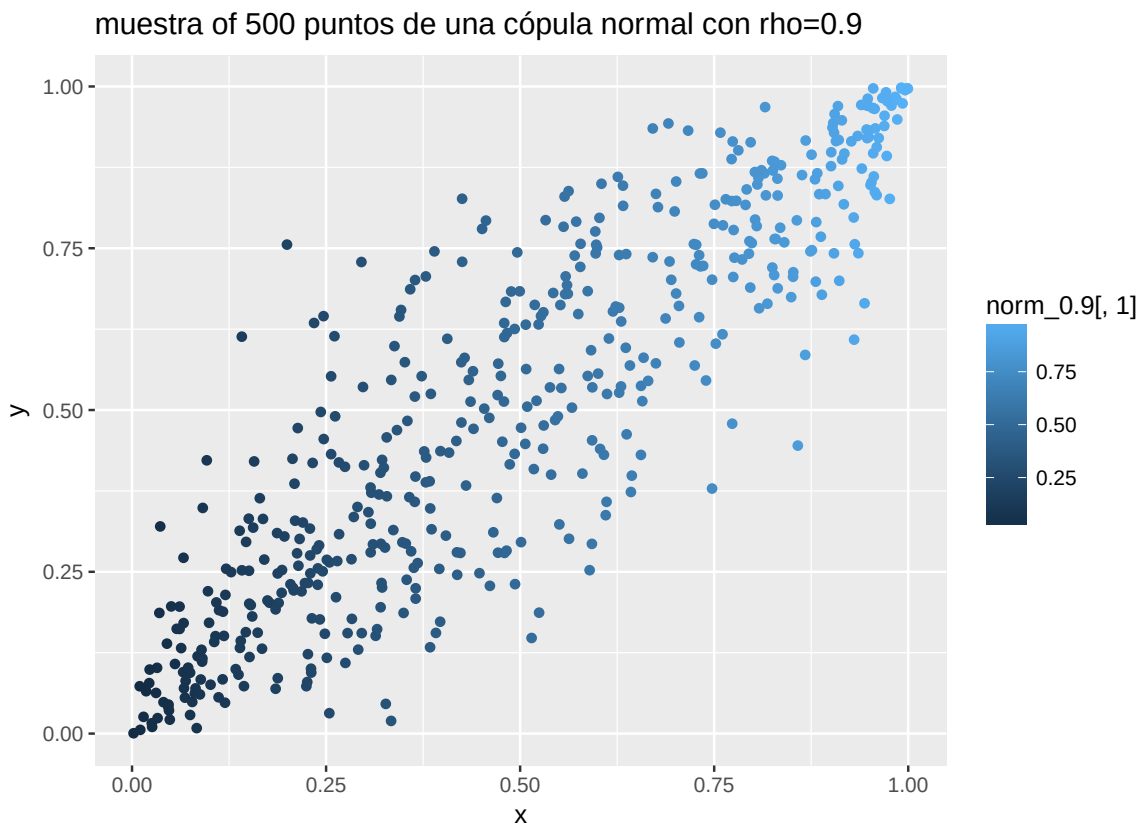


□

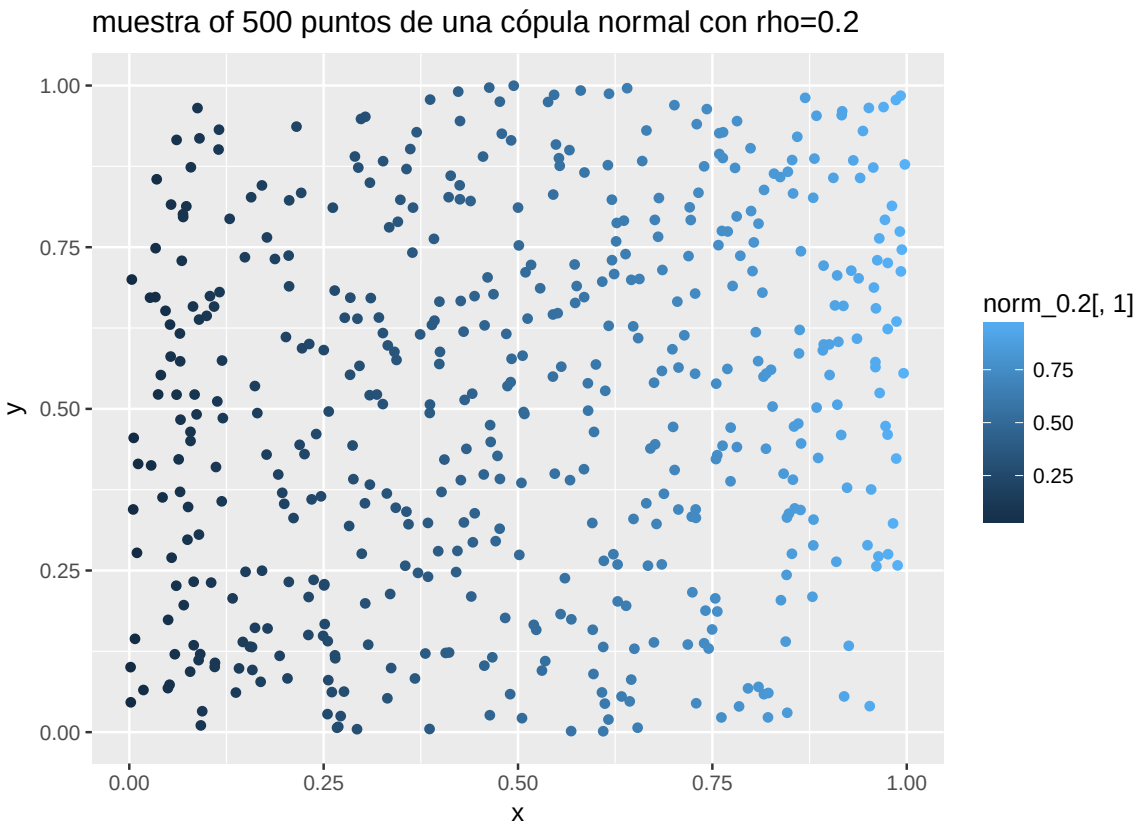
10. Usen la función `rCopula` del paquete `copula` para generar muestras de 500 puntos cuya distribución son las cópulas del ejercicio 8 anterior. Hagan una gráfica de las dos muestras. Teniendo en mente que una cópula determina la estructura de dependencia de una distribución multivariada conjunta, mirando estas gráficas, ¿pueden decir cuál de estas dos cópulas debe usarse para simular una distribución con una fuerte dependencia entre las marginales?

Solución.

```
norm_0.9 <- rCopula(500, normal_0.9) #muestra de la normal con rho=0.9
norm_0.2 <- rCopula(500, normal_0.2) #muestra de la normal con rho=0.2
library(ggplot2)
plot.norm_0.9 <- qplot(norm_0.9[,1], norm_0.9[,2], colour =     norm_0.9[,1],
                                    main="muestra of 500 puntos de una cópula normal con rho=0.9", xlab = "x", ylab = "y")
plot.norm_0.9
```




```
plot.norm_0.2 <- qplot(norm_0.2[,1], norm_0.2[,2], colour = norm_0.2[,1],
                       main="muestra of 500 puntos de una cópula normal con rho=0.2", xlab = "x", ylab = "y")
plot.norm_0.2
```



Claramente la cópula con parámetro 0.9 es la que relaciona densidades marginales con mayor dependencia.

□

11. 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 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, independientemente, 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 utilizarían 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) Desarrollen un estudio para 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:

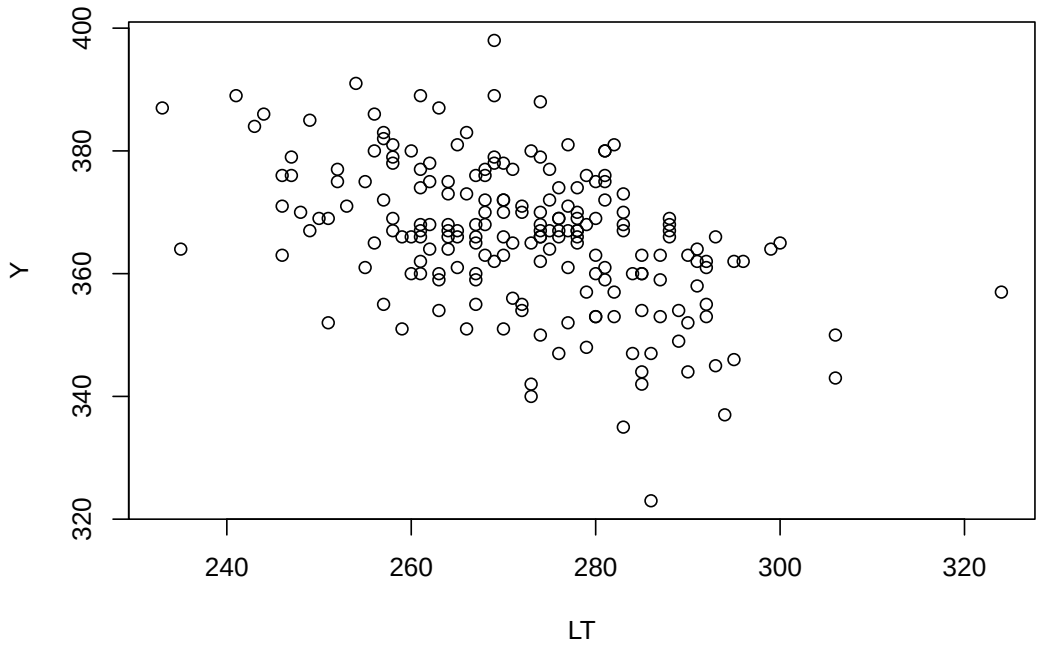
```
sigma0 <- c(1,2,3,4,5)  #permutación inicial
n <- 100
L <- I <- NULL
permutaciones <- matrix(rep(0,500),nrow=100,5)
for(j in 1:n){
i <- sample(1:5,size=1,replace = F,prob = (1:5)/15)
I[j] <- i
L[j] <- match(i,sigma0)
permutaciones[j,] <- c(i,sigma0[-match(i,sigma0)])
sigma0 <- permutaciones[j,]
}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


[1] 2.94
```

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.

```
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())

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

plot(datos)
```




```
head(datos)
```

```
      LT   Y
[1,] 290 363
[2,] 277 371
[3,] 258 381
[4,] 274 350
[5,] 246 363
[6,] 292 355
```

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 
-38.228  -8.190   1.372   8.603  46.748 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 482.50329   27.77774  17.370  < 2e-16 ***
Y            -0.57493    0.07582  -7.583 1.28e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.47 on 198 degrees of freedom
Multiple R-squared:  0.225,Adjusted R-squared:  0.2211 
F-statistic:  57.5 on 1 and 198 DF,  p-value: 1.284e-12
```

```
betahat <- coefficients(lm(LT~Y,data=as.data.frame(datos)))[2]
datos <- NULL
```

```
for(i in 1:1000)  datos <- rbind(datos,simula4())
datos <- as.data.frame(datos)
theta <- mean(datos$LT)
tildetheta <- mean(datos$LT - betahat*(datos$Y-366.67))
c(theta,tildetheta)

[1] 274.0860 274.1027

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

[1] 273.128 275.044

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

[1] 273.3313 274.8740
```

□

12. 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.

□

13. 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] -0.7329629
```

- 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] 0.2970000 0.4571652

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

[1] 0.3007997 0.2715124

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

[1] 40.60957
```

- 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] 0.2800000 0.4492236

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

[1] 0.2770949 0.3787295

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

[1] 15.69242
```

□

14. 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] 0.2529833
```

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

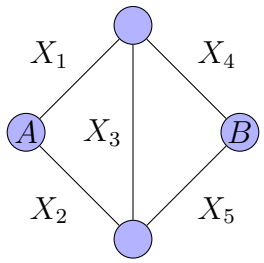
```
k <- 100
xu <- NULL
#piloto
u <- runif(k)
for(i in 1:length(u)) xu[i] <-rpois(1,15/(0.5+u[i]))

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] -7.086
```

- 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] 0.2506456
```

□

15. 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 $l = E[h(\mathbf{U})]$ con $h(\mathbf{U}) = H(a_1 U_1, \ldots, a_5 U_5)$.

- Obtener un estimador crudo de Monte Carlo 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] 0.926076947 0.003945846
```

- 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] 0.928925130 0.001363283
```

- Para variable de control, podemos considerar por ejemplo $Y = \min\{X_1 + X_4, X_2 + X_5\}$. 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] 0.929813622 0.000524424
```

- Para condicionamiento, definimos $Z_1 = \min\{X_4, X_3 + X_5\}$, $Z_2 = \min\{X_5, X_3 + X_4\}$, entonces $Y_1 = X_1 + Z_1$, $Y_2 = X_2 + Z_2$ y $Y = H(X)$ se puede escribir como $Y = \min\{Y_1, Y_2\}$ 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.

□