Algoritmo de Box-Muller

Si (X,Y) son variables aleatorias independientes con distribución normal estándar $N(0,1)$ entonces el vector (R,Θ) dado por

$$R = \sqrt{X^2 + Y^2} \quad \text{y} \quad \Theta = \arctan(X/Y),$$

tiene coordenadas independientes con Θ uniforme en $(0,2\pi)$ y R^2 tiene distribución exponencial Exp(1/2). De modo que podemos simular pares de variables normales independientes:

- Genera variables independientes U,W con $U \sim Unif(0,1)$ y $W \sim Exp(1/2)$.
- Define

$$X = \sqrt{W}\cos(2\pi U),$$
$$Y = \sqrt{W}\sin(2\pi U).$$

  entonces X,Y son independientes con distribución normal estándar.

1. La densidad de un vector de variables normales $Y = (Y_1,Y_2,\ldots,Y_n)$ está determinado por el vector de medias $\mu = (E(Y_1),\ldots,E(Y_n))$ y la matriz de covarianzas $\Sigma = (\Sigma_{ij})$ con $\Sigma_{ij} = \Sigma ji = Cov(Y_i,Y_j)$.
2. Una matriz simétrica A es diagonalizable y además tiene una raíz cuadrada B. Es decir, $A = Q\Lambda Q^{-1}$ y $B^T B = A$, donde Q tiene una base de eigenvectores de la función Ax y Λ es una matriz diagonal con los eigenvalores correspondientes.
   Más aún $Q^{-1} = Q^T$. Y de ese modo, $B = Q\Lambda^{1/2}$.

1. Si $X = (X_1,X_2,\ldots,X_n)$ es un vector de variables independientes normales estándar. Y definimos el vector $Y = (Y_1,Y_2,\ldots,Y_n)$ a partir de combinaciones lineares de X, es decir
$$Y = MX + b;$$
   entonces el vector Y es de variables normales con vector de medias $\mu = b$ y matriz de covarianza $\Sigma = MM^T$.

1 Multinormales como transformación de variables normales independientes

Para generar un vector Y de Normales Multivariadas con media μ y covarianza Σ dada:

1. Encuentra los eigenvectores y eigenvalores de Σ. Sea Q la matriz de eigenvectores y Λ la matriz diagonal de eigenvalores.
2. Genera un vector multinormal X con entradas independientes y estándar.
3. Define $M = Q\Lambda^{1/2}$ y regresa $Y = MX + \mu$.

2 Ejercicios

1. Genera y grafica muestras del vector aleatorio (X,Y) con densidad:

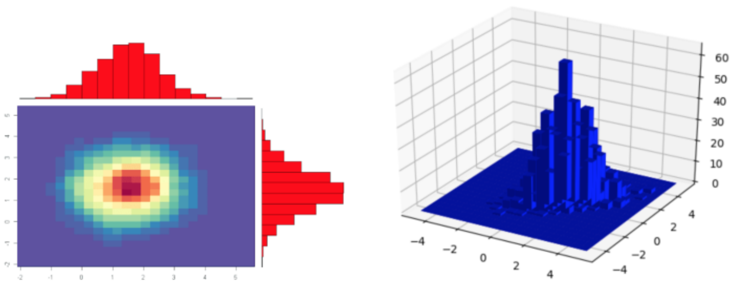
$$f(x,y) = \frac{e^{-\frac{x}{y}}e^{-y}}{y}$$

   Explica brevemente el algoritmo que usaste; recuerda que para explicar no es suficiente presentar el código.

2. Simula variables aleatorias normales multivariadas con

$$\mu = (0,0), \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ y } \mu = (1,0), \Sigma = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}$$

   Deberán agregar dos figuras para cada caso, como:



   Las figuras no deben ser idénticas a las anteriores, es parte de su práctica investigar como generarlas.

Nota: No olviden anexar el código en formato .r, poner el número de alumno en Moodle, y si desean poner su nombre que sea empezando por el apellido paterno pues así esta en la lista.

Ejercicio 1

Para explicar nuestro algoritmo primero tenemos que verificar si son variables independientes o dependientes para conocer el método a utilizar. Podemos hacerlo sacando las marginales de X y Y y verificar que:

$$f_{X,Y}(x,y) = f_X(x)f_Y(y)$$

Es decir, si el producto de las marginales nos da la densidad conjunta significa que son variables independientes. Otro forma de saber, es verificando si los factores se pueden seperar en terminos de x y y, si no se puede separar significa que no son independientes. Para nuestra distribución, vemos que no podemos separar los términos por lo tanto son variables dependientes, una vez sabiendo eso, partiremos con el método para simular variables dependientes:

```
Algoritmo
1.- Obtener la marginal de Y
2.- Conocer su distribucion y generar variables de Y
3.- Obtener X dado Y
```

Para obtener la marginal es:

$$f_y(y) = \int_0^\infty \frac{e^{-\frac{x}{y}}e^{-y}}{y} = e^{-y}$$

Vemos que la marginal de Y tiene una distribución exponencial con $\lambda = 1$, es decir, $exp(1)$.

Ahora debemos sacar X dado Y:

$$f(X|Y=y) = \frac{\frac{e^{-\frac{x}{y}}e^{-y}}{y}}{e^{-y}} = \frac{1}{y}e^{-\frac{x}{y}}$$

Vemos que tiene una distribución exponencial con $\lambda = \frac{1}{y}$.

Una vez obtenido X y Y procederemos a simularlas con R, para eso utilizaremos primer el metodo de la transformada inversa para generar variables aleatorias X y Y. Sabemos que la funcion de districion exponencial es:

$$F(X) = 1 - e^{-\lambda x}$$

Su función inversa seria:

$$\begin{aligned}
F(X) &= 1 - e^{-\lambda x} \\
u &= 1 - e^{-\lambda x} \\
u - 1 &= -e^{-\lambda x} \\
-u + 1 &= e^{-\lambda x} \\
ln(1-u) &= -\lambda x \\
-\frac{ln(1-u)}{\lambda} &= x \\
x &= F^{-1}(u,\lambda)
\end{aligned}$$

Una vez conociendo la inversa podremos generar variables exponeniales con el metodo de la transformada inversa:

```
#' Funcion inversa de la distribucion exponencial
#'
#' @param u - la probabilidad acumulada.
#' @param lambda parametro de forma.
#'
#' @return una variable aleatoria.
inversa_distribucion_exp <- function(u, lambda) {
  return((-ln(1-u)/lambda))
}

#' Generador de variables aleatorias exponencial.
#'
#' @param n - numero de variables.
#' @param lamda - parmetro de forma.
#'
#' @return vector de variables aleatorias.
generacion_va_exp <- function(n, lambda) {
  va = c()
  for (i in 1:n) {
    u = runif(1)
    va[i] = inversa_distribucion_exp(u, lambda)
  }
  return(va)
}
```

Uniendo las ideas ahora procederemos a generar variables aleatorias y graficarlas:

```
#'  Grfica  las variables aleatorias X y Y
#'
#' @param n - numero de variables aleatorias.
graficar_variables_X_Y <- function(n) {
  X <- generacion_va_exp(n, 1)
  Y <- generacion_va_exp(n, 1/y)
  plot(X, Y)
}

graficar_variables_X_Y(500)
```

Ejercicio 2

Para simular variables aleatorias normales multivariadas primero definimos nuestra matriz de covarianza y el vector de medias:

```
sigma1 <- rbind(c(1,0), c(0,1))
u1 <- c(0,0)
Y <- simular_va_norm_multivariadas(1000, sigma1, u1)
```

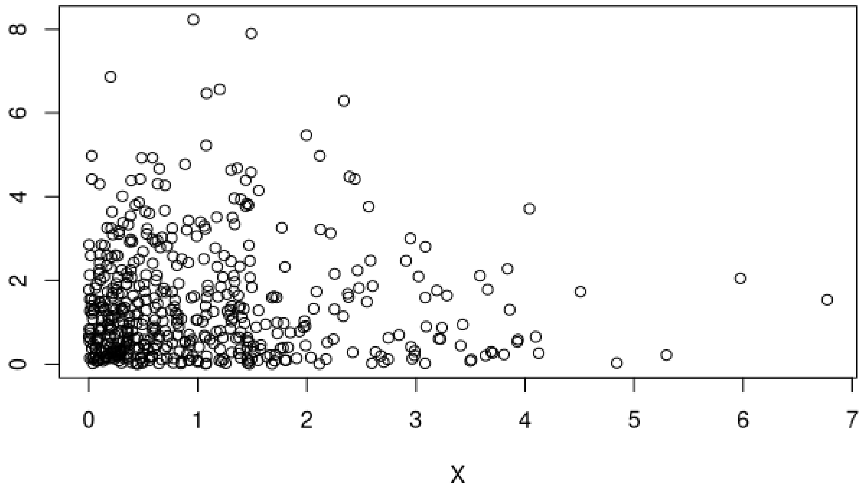
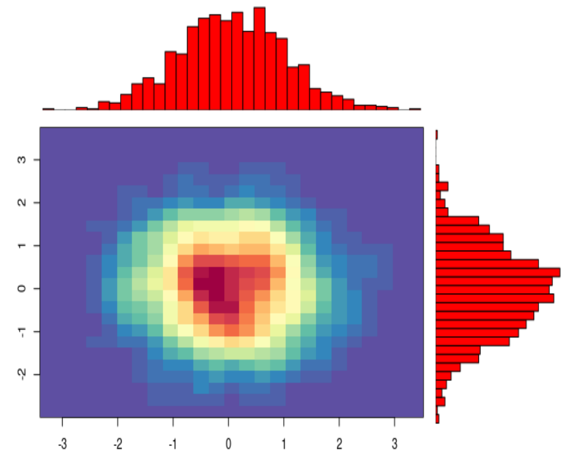


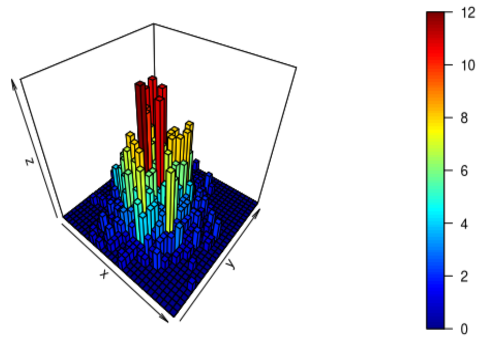
FIG. 1 – *Gráfica las variables aleatorias X y Y*

Para la simlulación seguimos los procedimientos del método Multinormales como transformación de variables normales independentes, encontramos los eigenvectores y eigenvalores de Σ, generemos un vector multinormal X con entradas independientes:

```r
#' Simula variables aleatorias normales multivariadas
#'
#' @param n - numero de simulaciones.
#' @param sigma - matriz de covarianza.
#' @param u - vector de medias.
#'
#' @return - el vector Y es de variables normales.
simular_va_norm_multivariadas <- function(n, sigma, u) {
  va_X <- c()
  va <- c()
  eigenvec_val <- eigen(sigma)
  Q <- eigenvec_val$vectors
  values <- eigenvec_val$values
  A <- rbind(c(values[1],0), c(0,values[2]))
  M = Q %*% A^0.5

  for (i in 1:n) {
    X = rnorm(ncol(M),0,1)
    Y = M %*% X + u
    va <- cbind(va, Y)
  }
  va <- t(va)
  return(va)
}
```





Para hacer las gráficas tuvimos que buscar en diferentes sitios web y libros de R, para hacerla tenemos que hacer un dataframe con nuestras variables aleatorias y darle formato por cada aspecto:

```
# Defimos un data frame para hacer nuestra grafica
df <-  data.frame(Y[,1], Y[,2])
h1 <- hist(df[,1], breaks=25, plot=F)
h2 <- hist(df[,2], breaks=25, plot=F)
top <- max(h1$counts, h2$counts)
k <- kde2d(df[,1], df[,2], n=25)

# Generamos las imagenes y juntamos los histogramas
oldpar <- par()
par(mar=c(3,3,1,1))
layout(matrix(c(2,0,1,3),2,2,byrow=T),c(3,1), c(1,3))
image(k, col = r)
par(mar=c(0,2,1,0))
barplot(h1$counts, axes=F, ylim=c(0, top), space=0, col='red')
par(mar=c(2,0,0.5,1))
barplot(h2$counts, axes=F, xlim=c(0, top), space=0, col='red', horiz=T)

hist3D(z=z, border="black")
```

Para la segunda matriz de covarianzas hicimos el mismo procedimiento para simular variables multinormales.

Referencias

– https://everydayanalytics.ca/2014/09/5-ways-to-do-2d-histograms-in-r.html
– https://stackoverflow.com/questions/30563340/how-to-make-3d-histogram-in-r