Capítulo 3

Simulación de vectores aleatorios

En esta práctica vamos a considerar vectores aleatorios bidimensionales. Denotaremos por $X = (X_1, X_2)$ un vector aleatorio tomando valores en $\mathbb{R}^2$ y por F_X su función de distribución bidimensional. En el caso bidimensional, la posible dependencia entre las componentes cobra una vital importancia. Esa posible dependencia se recoge en una función llamada *cópula*, y la simulación de valores aleatorios depende por tanto de dicha función. Sin embargo, la Teoría de Cópulas queda fuera de los objetivos del curso y nos centraremos en dos casos particulares más sencillos de manejar:

Independencia: Si $X = (X_1, X_2)$ es un vector aleatorio cuyas componentes son independientes, se cumple que $F_X(t_1, t_2) = F_{X_1}(t_1)F_{X_2}(t_2)$ para todo t_1, t_2, donde F_{X_1} y F_{X_2} denotan las funciones de distribución marginales de X_1 y X_2, respectivamente.

Vectores aleatorios discretos con soporte finito: Vamos a suponer que los valores que toma el vector aleatorio $X = (X_1, X_2)$ es finito.

En ambos casos veremos que la simulación de valores aleatorios es sencilla.

3.1. Simulación para variables aleatorias independientes

Tenemos un vector aleatorio $X = (X_1, X_2)$ cuyas componentes son independientes. En tal caso, simular un valor aleatorio (x_1, x_2) es muy sencillo: basta generar un valor aleatorio x_1 que proviene de X_1, y un valor aleatorio x_2 que proviene de X_2.

Veamos algún ejemplo de este procedimiento.

Ejemplo 3.1. Consideremos un vector aleatorio $X = (X_1, X_2)$ donde las componentes son independientes, X_1 sigue una distribución binomial $\mathcal{U}(0, 1)$ y X_2 sigue una distribución exponencial de parámetro $\lambda = 1$, $exp(1)$. Vamos a generar 2000 valores. Para ello, dado que las componentes son independientes, basta con generar los valores para cada una de las componentes por separado. Podemos entonces hacer uso de las instrucciones vistas en la primera práctica:

```
n<-2000               # Tamaño de la muestra
x1<-runif(n,0,1)      # Generación de los valores aleatorios de x1
x2<-rexp(n,1)         # Generación de los valores aleatorios de x2
x<-cbind(x1,x2)       # Unir los datos en una matrix de dimensión nx2
```

Una vez simulados los datos, es posible realizar estimaciones como hemos visto en las prácticas anteriores. Por ejemplo, las siguientes instrucciones se puede utizan para estimar $P(X_1 > 1, X_2 \in (1,2))$, $P(X_1 \in (0.25, 0.5))$, $E(X_1)$ y $E(X_2)$:

```
sum(x[,1]>1 & x[,2]>1 & x[,1]<2)/n
sum(x[,1]>0.25 & x[,1]<0.5)/n
mean(x[,1])
mean(x[,2])
```

Ejemplo 3.2. Consideramos el vector aleatorio $X = (X_1, X_2)$ que toma los valores $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$, cada uno de ellos con probabilidad $\frac{1}{4}$. Se puede comprobar fácilmente que, para este vector aleatorio, sus componentes son independientes y además sus marginales X_1 y X_2 siguen ambas una distribución Bernoulli con parámetro 0.5. Por lo tanto, si queremos generar por ejemplo 100 valores aleatorios para X, tendríamos dos procedimientos: el primero de ellos consistiría en generar 200 valores aleatorios, en una matriz 100×2, de una distribución binomial con parámetros 1 y 0.5:

```
n<-100
x<-matrix(rbinom(2*n,1,0.5),ncol=2)
```

La segunda opción sería seguir los pasos descritos al inicio de la práctica, simulando los datos de X_1 y X_2 por separado para posteriormente usar el comando `cbind`:

```
n<-100
x1<-rbinom(n,1,0.5)
x2<-rbinom(n,1,0.5)
x<-cbind(x1,x2)
```

3.2. Simulación para variables discretas no necesariamente independientes

En este apartado veremos cómo generar valores aleatorios de un vector aleatorio discreto con soporte finito donde las componentes no son necesariamente independientes. En este caso, la distribución de probabilidad conjunta de $\vec{X} = (X_1, X_2)'$ se puede representar de la siguiente manera:

$X_1 \backslash X_2$	y_1	$\dots$	y_m
x_1	$p_{1,1}$	$\dots$	$p_{1,m}$
$\dots$	$\dots$	$\dots$	$\dots$
x_n	$p_{n,1}$	$\dots$	$p_{n,m}$

Para generar los valores aleatorios, podemos utilizar la siguiente notación:

$$z_1 = (x_1, y_1), \quad z_2 = (x_1, y_2), \quad z_m = (x_1, y_m), \quad \dots$$

En general, si denotamos por $z_{(k-1)m+l} = (x_k, y_l)$, podemos generar valores aleatorios de la variable aleatoria Z, tomando los valores $z_1, \dots, z_{nm}$ con probabilidades $P(Z = z_{(k-1)m+l}) = p_{k,l}$.

Ejemplo 3.3. Consideramos el vector aleatorio $X = (X_1, X_2)$ tomando los siguientes valores:

$$\begin{array}{c|ccc} X_1 \backslash X_2 & 1 & 2 & 3 \\ \hline 1 & 0.2 & 0 & 0.2 \\ 2 & 0.2 & 0.3 & 0.1 \end{array} \tag{3.1}$$

Una forma de llevar a cabo la simulación sería generando valores aleatorios de una distribución uniforme en el intervalo $(0, 1)$, de manera que si el valor obtenido está entre 0 y 0.2, se asignará el valor $(1, 1)$; si el valor obtenido está entre 0.2 y 0.4, se asignará el valor $(1, 3)$; si el valor obtenido está entre 0.4 y 0.6, se asignará el valor $(2, 1)$; si el valor obtenido está entre 0.6 y 0.9, se asignará el valor $(2, 2)$; y por último, si el valor obtenido está entre 0.9 y 1, se asignará el valor $(2, 3)$. Para ello, se puede utilizar el siguiente código aunque, evidentemente, también se podría hacer de muchas otras maneras:

```
n<-50
u<-runif(n,0,1)
x<-(u<=0.2)%*%t(c(1,1))+(u>0.2 & u<=0.4)%*%t(c(1,3))+
        (u>0.4 & u<=0.6)%*%t(c(2,1))+(u>0.6 & u<=0.9)%*%t(c(2,2))+
        (u>0.9)%*%t(c(2,3))
```

Nótese que en esta expresión, cada uno de los vectores lógicos se interpreta como un vector columna (dimensión $n \times 1$), mientras que $t(c(i, j))$ es un vector (columna) transpuesto, es decir, de dimensión 1×2). Por lo tanto, las multiplicaciones de matrices concuerdan en dimensión dando lugar a matrices de dimensión $n \times 2$. Recordemos que para multiplicar matrices es necesario utilizar $\% * \%$, puesto que $*$ indica el producto componente a componente.

Los datos anteriores se pueden representar gráficamente utilizando el paquete ggplot2:

```
library(ggplot2)   # Cargar el paquete ggplot
library(latex2exp) # Cargar el paquete latex2exp que permite
                   # escribir símbolos matemáticos en las etiquetas
ggplot(data=NULL, aes(x[,1],x[,2]))+
  geom_count(col="red")+
  scale_size(range=c(2,15))+      # Cambia la escala del ancho de las
                                  # burbujas
  xlab(TeX("Componente $x_1$"))+ # Añade una etiqueta al eje x
  ylab(TeX("Componente $x_2$"))  # Añade una etiqueta al eje y
```

El gráfico obtenido, representado en la Figura 3.1, dibuja una burbuja en cada uno de los pares obtenidos en la muestra, de forma que el radio de la burbuja es proporcional a la frecuencia con la que dicho par ha aparecido. Se puede ver que la burbuja más grande se corresponde con el par (2,2), que es el par que más veces ha aparecido en la muestra y que es el par con mayor probabilidad (0.3), como se puede ver en la Ecuación 3.1.

Una vez que tenemos generada la muestra aleatoria, podemos trabajar también con las estimaciones de las distribuciones marginales y/o condicionadas.

Ejemplo 3.4. Continuemos con el ejemplo anterior, y vamos a resolver las siguientes cuestiones:

(a) Da una estimación de la distribución de probabilidad marginal de X_1.

(b) Estima la media de la variable $X_1|(X_2 = 3)$.

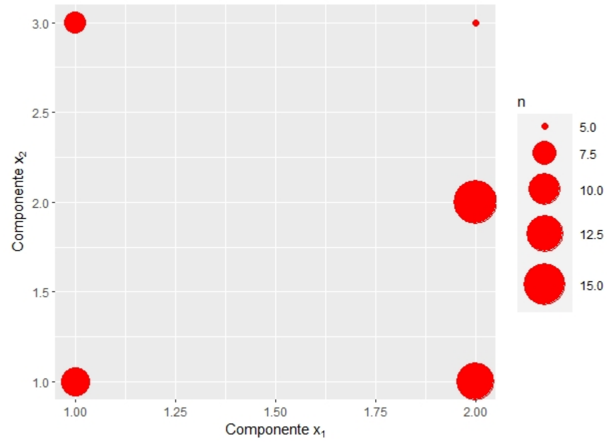

Figura 3.1: Gráfico de burbujas realizado con los datos simulados en el Ejemplo 3.3.

(c) Estima la probabilidad $P(X_2 = 3|X_1 = 1)$.

Para resolver el apartado (a), podemos utilizar los siguientes comandos:

```
x1<-x[,1]                          # Muestra estimada de la marginal X1
c(sum(x1==1)/n,sum(x1==2)/n)       # Estimación de las probabilidades
```

Para el apartado (b), hemos de calcular la estimación de la variable $X_1|X_2$. Para ello seleccionamos los elementos del vector `x` cuya segunda componente tome el valor 3. Eso lo podemos hacer con la siguiente instrucción:

```
y1<-x[x[,2]==3,1]
```

La interpretación de esta expresión es la siguiente: se extraen de la matriz `x` aquellos valores de la primera columna (primera componente, x_1) donde la segunda columna (segunda componente, x_2) toma el valor 3. A continuación, estimamos la media de la variable X_1 cuando $X_2 = 3$:

```
mean(y1)
```

Para el apartado (c), tenemos que coniderar la distribución condicionada de X_2 cuando $X_1 = 1$. Al igual que hicimos en el apartado anterior, primero calculamos la estimación de dichos valores:

```
y2<-x[x[,1]==1,2]
```

A continuación, estimamos la probabilidad de que $X_2 = 3$ cuando $X_1 = 1$:

```
sum(y2==3)/length(y2)
```

En esta última expresión, se divide entre la longitud del vector `y2` puesto que este vector no tiene, necesariamente, n datos, puesto que se extraen *algunos* de los valores de la segunda columna de la matriz `x`.

3.3. Ejercicios propuestos

Ejercicio 3.1. Considera el vector aleatorio $X = (X_1, X_2)$ cuya distribución de probabilidad viene dada por:

$X_1 \backslash X_2$	0	1	2	3
1	0.2	0	0	0.4
5	0.1	0.1	0.2	0

1. Simula $n = 5000$ valores aleatorios de $X = (X_1, X_2)$.

2. ¿Cuál es la estimación de $P(X = (5, 2))$ en la muestra obtenida?

3. Considera la distribución marginal de X_2. Estima su media, su desviación típica y dibuja su función de distribución empírica.

4. Considera la distribución condicionada $X_1|X_2 = 0$. Estima su media, su mediana y su desviación típica.

Ejercicio 3.2. Considera el vector aleatorio $X = (X_1, X_2)$ cuya distribución de probabilidad viene dada por:

$X_1 \backslash X_2$	0	5	10
-1	0.1	0	0.3
3	0.2	0	0.1
10	0	0.1	0.2

1. Simula $n = 10000$ valores aleatorios de $X = (X_1, X_2)$.

2. Considera la distribución marginal de X_1. Estima su media, su mediana y dibuja su función de distribución empírica.

3. Considera la distribución condicionada $X_1|X_2 = 10$. Estima su media, su mediana y dibuja su función de distribución empírica. Compara los valores obtenidos con los del apartado anterior.