Capítulo 4

Simulación de distribuciones normales bivariantes

En esta práctica vamos a analizar en detalle la distribución normal bivariante. A lo largo de esta práctica aprenderemos a:

Simulación: Veremos cómo podemos simular datos que provienen de una distribución normal bivariante. Para ello, previamente enunciaremos alguna propiedades necesarias de esta distribución.

Representación gráfica de su densidad: Veremos cómo realizar representaciones gráficas de las densidades de una normal bivariante, viendo cómo interpretar la gráfica obtenida.

4.1. Distribución normal bivariante

Partimos de un vector aleatorio bidimiensional $\vec{X} = (X_1, X_2)'$. Decimos que X sigue una distribución normal bivariante con vector de medias $\vec{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$ y matriz de varianzas-covarianzas $\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{pmatrix}$, y se denota por:

$$\vec{X} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \equiv \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{pmatrix} \right), \tag{4.1}$$

si su función de densidad conjunta viene dada por:

$$f(x_1, x_2) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu})'\Sigma^{-1}(\vec{x} - \vec{\mu}) \right), \tag{4.2}$$

donde $\vec{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, $|\Sigma|$ y Σ^{-1} denotan el determinante y la inversa de la matrix Σ, respectivamente, y $(\vec{x} - \vec{\mu})'$ denota el vector transpuesto de $(\vec{x} - \vec{\mu})$, que viene dado por:

$$(\vec{x} - \vec{\mu})' = (x_1 - \mu_1, x_2 - \mu_2)\,.$$

En el vector de medias $\vec{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$, μ_1 and μ_2 denotan las medias de X_1 y X_2, respectivamente. Asimismo, en la matriz de varianzas-covarianzas Σ, σ_1 y σ_2 denotan las desviaciones típicas de X_1 y X_2, respectivamente, mientras que ρ denota el coeficiente de correlación lineal entre X_1 y X_2 ($\rho \in [-1, 1]$). Además, si (X_1, X_2) sigue una distribución normal bivariante como en (4.1), las marginales de X_1 y X_2 siguen distribuciones normales (univariantes):

$$X_1 \equiv \mathcal{N}(\mu_1, \sigma_1), \qquad X_2 \equiv \mathcal{N}(\mu_2, \sigma_2).$$

4.2. Simulación de valores aleatorios de una distribución normal bivariante

A continuación vamos a explicar dos métodos diferentes que permiten simular valores aleatorios que provienen de una distribución normal bivariante. La primera de ellas será mediante el uso del resultado antes enunciado, mientras que la segunda de ellas será mediante el uso del comando `mvnrnd`.

Existe un paquete en `R`, el paquete `MASS`, donde se ha implementado el comando `mvrnorm` que permite simular los datos buscados directamente. Solamente hemos de indicar el vector de medias y la matriz de varianzas-covarianzas, así como el número de datos deseados.

Ejemplo 4.1. Vamos a generar $n = 1000$ valores aleatorios que provienen de una distribución normal bivariante con vector de medias $\mu = \begin{pmatrix} 5 \\ 1 \end{pmatrix}$ y matrix de varianzas-covarianzas $\Sigma = \begin{pmatrix} 4 & 3 \\ 3 & 9 \end{pmatrix}$. Eso significa que $\sigma_1^2 = 4$, $\sigma_2^2 = 9$ y $\rho = 0.5$. En primer lugar, cargamos el paquete necesario:

```
library("MASS")
```

A continuación, basta indicar el vector de medias, la matrix de varianzas-covarianzas, así como el número de datos buscados:

```
n<-1000;
mu<-c(5,1)
sigma<-matrix(c(4,3,3,9),nrow=2)
x<-mvrnorm(n,mu,sigma)
```

La matrix `x` tiene dimensiones $n \times 2$, es decir, cada fila hace referencia a cada par aleatorio de $\vec{X}$, mientras que cada columna hace referencia a cada una de sus componentes: X_1 y X_2.

Los datos obtenidos se pueden representar mediante el paquete `ggplot2`, ya utilizado en prácticas anteriores. En primer lugar, podemos representar la nube de puntos de estos datos:

```
library(ggplot2)
library(latex2exp)
ggplot(data=NULL, aes(x[,1],x[,2]))+
  geom_point(size=1)+
  xlab(TeX("Componente $x_1$"))+
  ylab(TeX("Componente $x_2$"))
```

De esta forma, se obtiene la nube de puntos de los datos simulados. A continuación, podemos añadir las curvas de nivel de la función de densidad estimada a partir de la muestra:

```
ggplot(data=NULL, aes(x[,1],x[,2]))+
  geom_point(size=1)+
  geom_density_2d(size=1.2)+
  xlab(TeX("Componente $x_1$"))+
  ylab(TeX("Componente $x_2$"))
```

Como se puede ver (Figura 4.1, arriba a la izquierda), todas las curvas de nivel se representan con el mismo color y la misma intensidad. Haciendo un pequeño cambio en la última instrucción, podemos cambiar la tonalidad de la curva de nivel en función de lo cercana o lejana que se encuentre al centro de la nube de puntos (Figura 4.1, arriba a la derecha):

```
ggplot(data=NULL, aes(x[,1],x[,2]))+
  geom_point(size=1)+
  geom_density_2d(size=1.2, aes(col=..level..))+
  xlab(TeX("Componente $x_1$"))+
  ylab(TeX("Componente $x_2$"))
```

Si bien por defecto `R` pinta las curvas de nivel en (distintas tonalidades de) azul, podemos pedir que lo pinte en otra gama de colores distinta (Figura 4.1, abajo a la izquierda). Por ejemplo:

```
ggplot(data=NULL, aes(x[,1],x[,2]))+
  geom_point(size=1)+
  geom_density_2d(size=1.2, aes(col=..level..))+
  scale_color_viridis_c()+
  xlab(TeX("Componente $x_1$"))+
  ylab(TeX("Componente $x_2$"))
```

Por último, podemos rellenar el fondo del gráfico con un color que cambie en función de lo cercana o lejana que esté dicha región al centro de la nube de puntos (Figura 4.1, abajo a la derecha):

```
ggplot(data=NULL, aes(x[,1],x[,2]))+
  geom_point(size=1)+
  geom_density_2d(size=1.2, aes(col=..level..))+
  scale_color_viridis_c()+
  geom_density2d_filled(alpha=0.4)+
  xlab(TeX("Componente $x_1$"))+
  ylab(TeX("Componente $x_2$"))
```

Los gráficos obtenidos se han representado en la Figura 4.1.

4.3. Representación gráfica de la función de densidad

Utilizando las opciones de gráficos 3D, podemos hacer una representación gráfica de la función de densidad de una función de densidad de una distribución normal bivariante. Para ello, vamos a utilizar dos comandos: `seq` y `outer`. El comando `seq` tiene tres argumentos, los dos primeros indican un intervalo, y el tercero de ellos indica el número de puntos que se van a tomar dentro del intervalo, de manera que los puntos elegido sean equidistance. Por ejemplo, para dividir el intervalo (0,10) en 5 partes, podemos usar la instrucción:

```
seq(0,10,length=5)
[1]  0.0  2.5  5.0  7.5 10.0
```

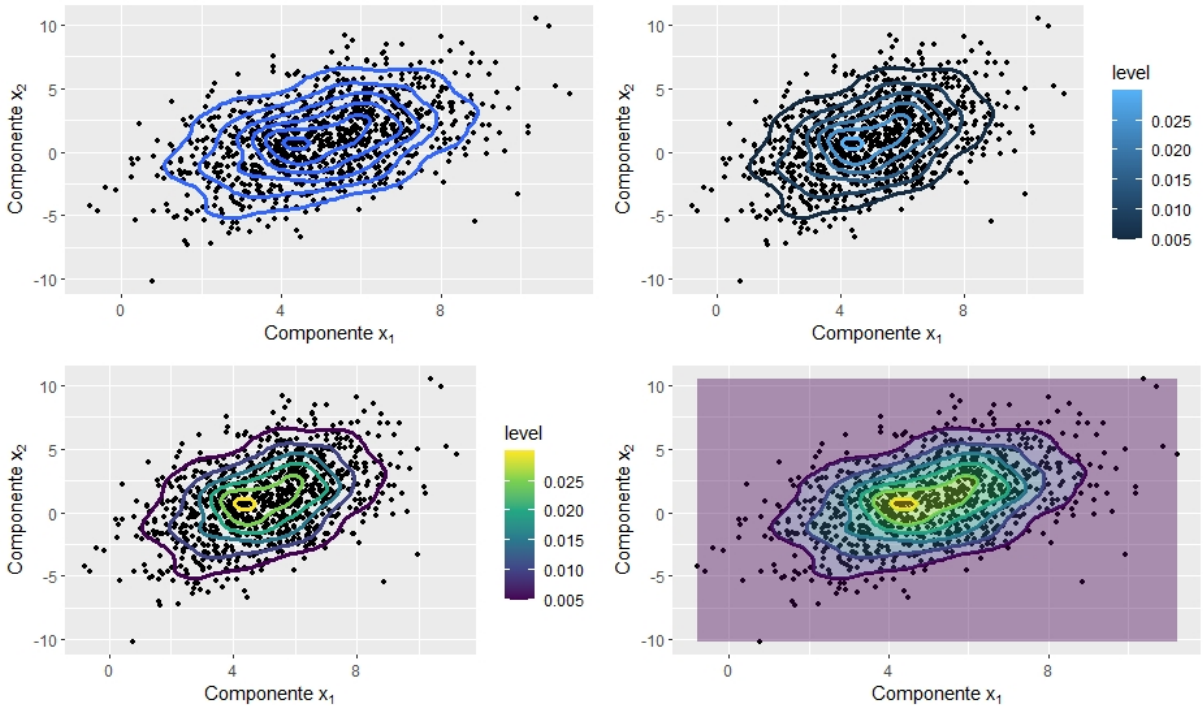


Figura 4.1: Representaciones gráficas de la nube de puntos simulados en el Ejemplo 4.1 junto con las estimaciones de las curvas de nivel.

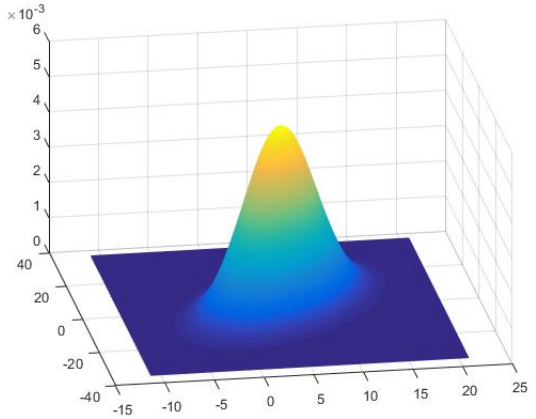
Por otra parte, la instrucción `outer` tiene tres argumentos de entrada. Los dos primeros son dos vectores ordenados, de forma que este comando creará una matriz con el producto cartesiano, al que se aplicará el tercer argumento de entrada, que será una función. Por ejemplo, si tomamos $a = (1, 2, 3)$ y $b(10, 20, 30)$, y la función elegida es la suma, el comando `outer` nos da lo siguiente:

```
fd<-function(s,t){s+t}
fd<-Vectorize(fd)
a<-c(1,2,3)
b<-c(10,20,30)
outer(a,b,fd)
     [,1] [,2] [,3]
[1,]   11   21   31
[2,]   12   22   32
[3,]   13   23   33
```

Para realizar la representación gráfica de la densidad de una distribución normal dada por (4.1), definiremos aplicaremos el comando `seq` sobre intervalos centrados en μ_1 y μ_2, respectivamente, con una amplitud suficiente. A continuación, definimos la red de puntos que produce el comando `outer`, y por último, calculamos el valor de la función de densidad, dada en la ecuación (4.1) para cada uno de los puntos de la red de puntos.

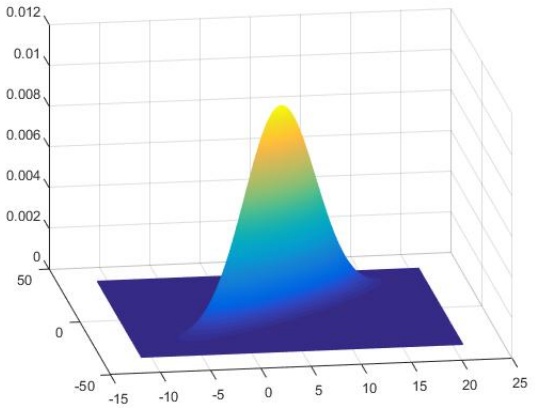
Ejemplo 4.2. Consideramos la misma distribución normal bivariante del Ejemplo 4.1, cuyo vector de medias era $\vec{\mu} = \begin{pmatrix} 5 \\ 1 \end{pmatrix}$ y su matrix de varianzas-covarianzas $\Sigma = \begin{pmatrix} 4 & 3 \\ 3 & 9 \end{pmatrix}$. Seguimos a continuación los pasos explicados anteriormente:

```
mu<-c(5,1)
sigma<-matrix(c(4,3,3,9),nrow=2)
is<-solve(sigma)
fd<-function(s,t) {x<-matrix(c(s,t),2,1);
    exp(-0.5*t(x-mu)%*%is%*%(x-mu))/(2*pi*sqrt(det(sigma)))}
fd<-Vectorize(fd)
x<-seq(-3,3,length=100)*sqrt(sigma[1,1])+5
y<-seq(-3,3,length=100)*sqrt(sigma[2,2])+1
z<-outer(x,y,fd)
persp(x,y,z,theta=30,phi=30)
```



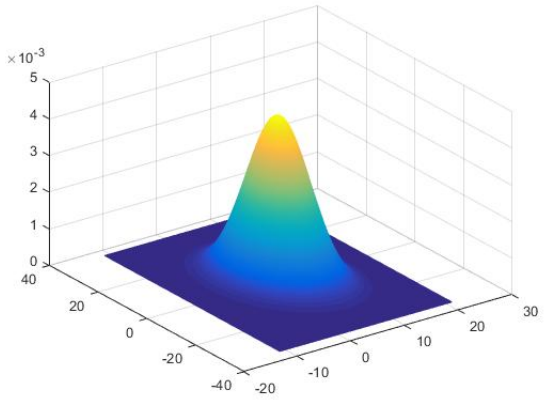
Como podemos observar en la gráfica, la zona central es la que toma valores más altos, tomando el valor más alto en el punto de medias: $(5,1)'$. Además, vemos que la curva se abre más en la segunda componente que en la primera puesto que $\sigma_2 > \sigma_1$. Por último, dado que el coeficiente de correlación es positivo ($\rho = 0.5$), la "cresta" se va moviendo en una región creciente, en el sentido de que a medida que aumenta (disminuye) una componente, también aumenta (disminuye) la otra componente.

Si aumentamos el coeficiente de correlación hasta $\rho = 0.9$, la matriz de varianzas-covarianzas sería $\Sigma = \begin{pmatrix} 4 & 5.4 \\ 5.4 & 9 \end{pmatrix}$. En las órdenes anteriores, solamente tendríamos que sustituir la definición de `sigma`, y obtendríamos la siguiente gráfica:



Como podemos ver en esta segunda gráfica, como el coeficiente de correlación es más próximo a 1, la "cresta" es más afilada.

Por último, si el coeficiente de correlación fuese negativo, por ejemplo $\rho = -0.25$, la matriz de varianzas-covarianzas sería $\Sigma = \begin{pmatrix} 4 & -0.75 \\ 0.75 & 9 \end{pmatrix}$. Modificando las instrucciones anteriores, podríamos hacer la siguiente representación gráfica:



En este caso vemos que la cresta sigue una forma decreciente, en el sentido de que cuando aumenta una componente, disminuye la otra. Además, la cresta no es muy afilada porque el coeficiente de correlación no es muy distinto de 1.

Concluimos por tanto que a la hora de analizar las densidades de una distribución normal bivariante, hemos de tener en cuenta lo siguiente:

1. La dirección de la cresta: si ambas componentes crecen en la misma dirección a lo largo de la cresta, el coeficiente de correlación será positivo; en caso contrario, ρ sería negativo.
2. En cuál de ambas direcciones (x_1 o x_2) la curva es más amplia, lo que indicará qué componente tiene una mayor varianza.
3. Por último, cuanto más afilada esté la cresta, más cercano está $|\rho|$ a 1.

4.4. Ejercicios propuestos

Ejercicio 4.1. Queremos simular valores aleatorios de una distribución normal bivariante donde $\mu_1 = \mu_2 = 0$, $\sigma_1 = \sigma_2 = 1$ y ambas componentes son independientes. Explica cómo simular valores aleatorios de dos formas distintas:

1. Siguiendo los pasos dados en esta práctica.
2. Utilizando solamente las instrucción de la primera práctica.

Ejercicio 4.2. Considera una distribución normal bivariante con vector de medias $\vec{\mu} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$ y matriz de varianzas-covarianzas $\Sigma = \begin{pmatrix} 4 & 2 \\ 2 & 3 \end{pmatrix}$.

1. Dibuja la densidad conjunta de ambas variables y explica la gráfica.
2. Simula $n = 10000$ valores aleatorios.

3. Utiliza los valores simulados para estimar la probabilidad $P(X_1 > 2, X_2 \leq 3.5)$.

Ejercicio 4.3. Dibuja las funciones de densidad de una distribución normal bivariante en los siguientes casos:

1. $\vec{\mu} = \begin{pmatrix} 0 \\ 5 \end{pmatrix}$ y $\Sigma = \begin{pmatrix} 1 & 4.5 \\ 4.5 & 25 \end{pmatrix}$.
2. $\vec{\mu} = \begin{pmatrix} 0 \\ 5 \end{pmatrix}$ y $\Sigma = \begin{pmatrix} 1 & -4.5 \\ -4.5 & 25 \end{pmatrix}$.
3. $\vec{\mu} = \begin{pmatrix} 0 \\ 5 \end{pmatrix}$ y $\Sigma = \begin{pmatrix} 1 & 1 \\ 1 & 25 \end{pmatrix}$.

Compara las tres densidades explicando para cada una de ellas los factores más relevantes.