Análisis discriminante lineal (LDA) y Análisis discriminante cuadrático (QDA)

Joaquín Amat Rodrigo *j.amatrodrigo@gmail.com*

Septiembre, 2016

Índice

Análisis discriminante lineal

Idea intuitiva

El Análisis Discriminante Lineal o *Linear Discrimiant Analysis (LDA)* es un método de clasificación de variables cualitativas en el que dos o más grupos son conocidos *a priori* y nuevas observaciones se clasifican en uno de ellos en función de sus características. Haciendo uso del teorema de Bayes, *LDA* estima la probabilidad de que una observación, dado un determinado valor de los predictores, pertenezca a cada una de las clases de la variable cualitativa, $P(Y = k|X = x)$. Finalmente se asigna la observación a la clase k para la que la probabilidad predicha es mayor.

Es una alternativa a la regresión logística cuando la variable cualitativa tiene más de dos niveles. Si bien existen extensiones de la regresión logística para múltiples clases, el *LDA* presenta una serie de ventajas.

- Si las clases están bien separadas, los parámetros estimados en el modelo de regresión logística son inestables. El método de *LDA* no sufre este problema.
- Si el número de observaciones es bajo y la distribución de los predictores es aproximadamente normal en cada una de las clases, *LDA* es más estable que la regresión logística.

Cuando se trata de un problema de clasificación con solo dos niveles, ambos métodos suelen llegar a resultados similares.

El proceso de un análisis discriminante puede resumirse en 6 pasos:

- Disponer de un conjunto de datos de entrenamiento (*training data*) en el que se conoce a que grupo pertenece cada observación.
- Calcular las probabilidades previas (*prior probabilities*): la proporción esperada de observaciones que pertenecen a cada grupo.
- Determinar si la varianza o matriz de covarianzas es homogénea en todos los grupos. De esto dependerá que se emplee *LDA* o *QDA*.
- Estimar los parámetros necesarios para las funciones de probabilidad condicional, verificando que se cumplen las condiciones para hacerlo.
- Calcular el resultado de la función discriminante. El resultado de esta determina a que grupo se asigna cada observación.
- Utilizar validación cruzada (*cross-validation*) para estimar las probabilidades de clasificaciones erróneas.

Teorema de Bayes para clasificación

Considérense dos eventos A y B, el teorema de Bayes establece que la probabilidad de que B ocurra habiendo ocurrido A $(B|A)$ es igual a la probabilidad de que A y B ocurran al mismo tiempo (AB) dividida entre la probabilidad de que ocurra A.

$$P(B|A) = \frac{P(AB)}{P(A)}$$

Supóngase que se desea clasificar una nueva observación en una de las K clases de una variable cualitativa Y, siendo $K \geq 2$, a partir de un solo predictor X. Se dispone de las siguientes definiciones:

- Se define como *overall*, *prior probability* o probabilidad previa (π_k) la probabilidad de que una observación aleatoria pertenezca a la clase k.
- Se define $f_k(X) \equiv P(X = x|Y = k)$ como la función de densidad de probabilidad condicional de X para una observación que pertenece a la clase k. Cuanto mayor sea $f_k(X)$ mayor la probabilidad de que una observación de la clase k adquiera una valor de $X \approx x$.
- Se define como *posterior probability* o probabilidad posterior $P(Y = k|X = x)$ la probabilidad de que una observación pertenezca a la clase k siendo x el valor del predictor.

Aplicando del teorema de Bayes se pueden conocer la *posterior probability* para cada clase:

$$P(pertenecer\ a\ la\ clase\ k \mid valor\ x\ observado) = \frac{P(pertenecer\ a\ la\ clase\ k\ y\ observar\ x)}{P(observar\ x)}$$

Si se introducen los términos, definidos anteriormente, dentro la ecuación se obtiene:

$$P(Y = k|X = x) = \frac{\pi_k P(X = x|Y = k)}{\sum_{j=1}^{K} \pi_j\, P(X = x|Y = j)} = \frac{\pi_k f_k(x)}{\sum_{j=1}^{K} \pi_j\, f_j(x)}$$

La clasificación con menor error (clasificación de Bayes) se conseguirá asignando la observación a aquel grupo que maximice la *posterior probability*. Dado que el denominador $\sum_{j=1}^{K} \pi_j f_j(x)$ es igual para todas las clases, la norma de clasificación es equivalente a decir que se asignará cada observación a aquel grupo para el que $\pi_k f_k(x)$ sea mayor.

Para que la clasificación basada en Bayes sea posible, se necesita conocer la probabilidad poblacional de que una observación cualquiera pertenezca a cada clase (π_k) y la probabilidad poblacional de que una observación que pertenece a la clase k adquiera el valor x en el predictor, $(f_k(X) \equiv P(X = x|Y = k))$. En la práctica, raramente se dispone de esta información, por lo que los parámetros tienen que ser estimados a partir de la muestra. Como consecuencia, el clasificador *LDA* obtenido se aproxima al clasificador de Bayes pero no es igual.

Estimación de π_k y $f_k(X)$

La capacidad del *LDA* para clasificar correctamente las observaciones depende de cómo de buenas sean las estimaciones de π_k y $f_k(X)$, cuando más cercanas al valor real, más se aproximará el clasificador *LDA* al clasificador de Bayes. En el caso de la *prior probability* (π_k) la estimación suele ser sencilla, si se quiere conocer la probabilidad de que una observación cualquiera pertenezca a la clase k se divide el número de observaciones de esa clase entre el número total de observaciones $\hat{\pi}_k = \frac{n_k}{N}$.

La estimación de $f_k(X)$ no es tan directa y para conseguirla se requiere de ciertas asunciones. Si se considera que $f_k(X)$ se distribuye de forma normal en las K clases, entonces se puede estimar su valor a partir de la ecuación:

$$f_k(X) = P(Y = k|X = x) = \frac{1}{\sqrt{2\pi\sigma_k}}\ exp(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2)$$

Donde μ_k y σ_k^2 son la media y la varianza para la clase k.

Si además se asume que la varianza es constante en todos los grupos $\sigma_1^2 = \sigma_2^2 \ldots = \sigma_k^2 = \sigma^2$, entonces el sumatorio $\sum_{j=1}^{K} \pi_j f_j(x)$ se simplifica en gran medida permitiendo calcular la *posterior probability* según la ecuación:

$$P(Y = k|X = x) = \frac{\pi_k \frac{1}{\sqrt{2\pi\sigma}}\ exp(-\frac{1}{2\sigma^2}(x - \mu_k)^2)}{\sum_{j=1}^{K} \pi_j \frac{1}{\sqrt{2\pi\sigma}}\ exp(-\frac{1}{2\sigma^2}(x - \mu_j)^2)}$$

Esta ecuación se simplifica aún más mediante una transformación logarítmica de sus dos términos:

$$\hat{\delta}_k(x) = log(P(Y = k|X = x)) = x\ \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + log(\pi_k)$$

El término lineal en el nombre *Análisis discriminante lineal* se debe al hecho de que la función discriminatoria es lineal respecto de *X*.

En la práctica, a pesar de tener una certeza considerable de que *X* se distribuye de forma normal dentro de cada clase, los valores $\mu_1 \dots \mu_k$, $\pi_1 \dots \pi_k$ y σ^2 se desconocen, por lo que tienen que ser estimados a partir de las observaciones. En las estimaciones empleadas en *LDA* son:

$$\hat{\mu_k} = \frac{1}{n_k} \sum_{i:y_1} x_i$$

$$\hat{\sigma_k} = \frac{1}{N - K} \sum_{k=1}^{K} \sum_{i:y_1} (x_i - \hat{\mu}_k)^2$$

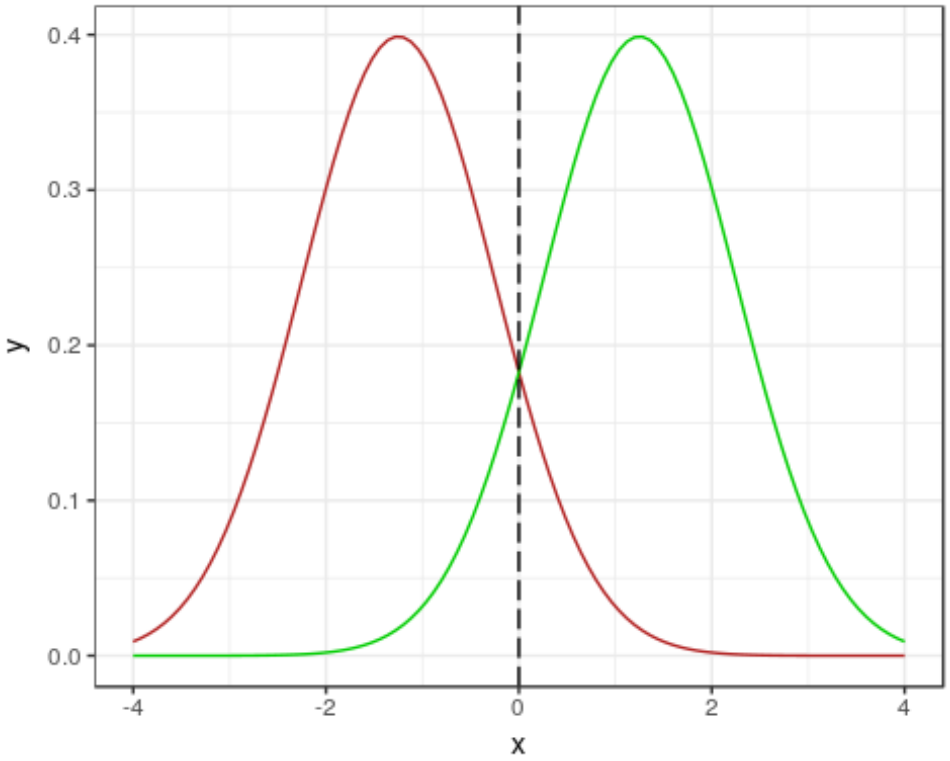
$$\hat{\pi}_k = \frac{n_k}{N}$$

$\hat{\mu_k}$ es la media de las observaciones del grupo k, $\hat{\sigma_k}$ es la media ponderada de las varianzas muestrales de las *K* clases y $\hat{\pi}_k$ la proporción de observaciones de la clase k respecto al tamaño total de la muestra.

La clasificación de Bayes consiste en asignar cada observación $X = x$ a aquella clase para la que $P(Y = k|X = x)$ sea mayor. En el caso particular de una variable cualitativa *Y* con solo dos niveles, se puede expresar la regla de clasificación como un ratio entre las dos *posterior probabilities*. Se asignará la observación a la clase 1 si $\frac{P(Y=1|X=x)}{P(Y=2|X=x)}$>1, y a la clase 2 si es menor. En este caso particular el límite de decisión de Bayes viene dado por $x = \frac{\mu_1 + \mu_2}{2}$.

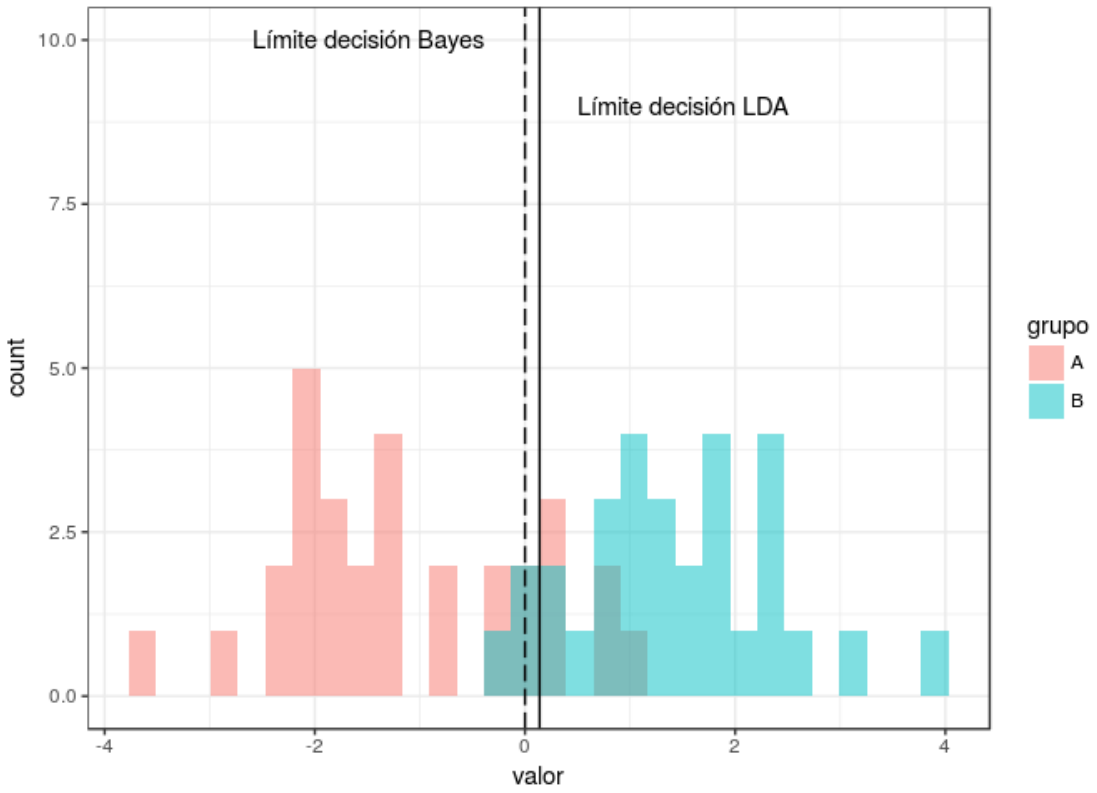
La siguiente imagen muestra dos grupos distribuidos de forma normal con medias μ_1 = -1.25, μ_2 = 1.25 y varianzas $\sigma^2{}_1 = \sigma^2{}_2 = 1$. Dado que se conoce el valor real de las medias y varianzas poblacionales (esto en la realidad no suele ocurrir), se puede calcular el límite de decisión de Bayes $x = \frac{-1.25+1.25}{2}$= 0 (linea discontinua).

```
library(ggplot2)
ggplot(data.frame(x = c(-4, 4)), aes(x)) + stat_function(fun = dnorm, args =
list(mean = -1.25, sd = 1), color = "firebrick") + stat_function(fun = dnorm, args
= list(mean = 1.25, sd = 1), color = "green3") + geom_vline(xintercept = 0,
linetype = "longdash") + theme_bw()
```



Si en lugar de conocer la verdadera distribución poblacional de cada grupo solo se dispone de muestras, escenario que suele ocurrir en los casos reales, el límite de decisión *LDA* se aproxima al verdadero límite de decisión de Bayes pero no es exacto. Cuanto más representativas sean las muestras mejor la aproximación.

```r
set.seed(6911)
library(ggplot2)
grupo_a <- rnorm(n = 30, mean = -1.25, sd = 1)
grupo_b <- rnorm(n = 30, mean = 1.25, sd = 1)
datos <- data.frame(valor = c(grupo_a, grupo_b), grupo = rep(c("A", "B"), each =
30))

ggplot(data = datos, aes(x = valor, fill = grupo)) +
  geom_histogram(alpha = 0.5, position = "identity") +
  geom_vline(xintercept = 0, linetype = "longdash") +
  geom_vline(xintercept = (mean(grupo_a) + mean(grupo_b))/2) +
  geom_text(aes(+1.2, 9, label = "Límite decisión LDA")) +
  geom_text(aes(-1.2, 10, label = "Límite decisión Bayes")) +
  theme_bw()
```



Extensión del LDA para múltiples predictores

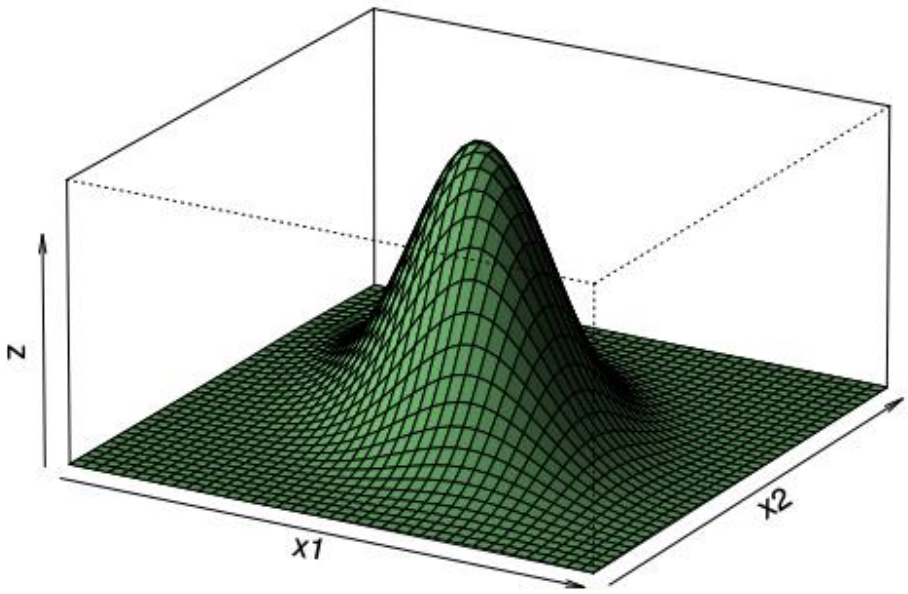
Los conceptos anteriormente descritos empleando un único predictor pueden generalizarse para introducir múltiples predictores en el modelo. La diferencia reside en que X, en lugar de ser un único valor, es un vector formado por el valor de p predictores $X = (X_1, X_2, \ldots, X_3)$ y que, en lugar de proceder de una distribución normal, procede de una distribución normal multivariante.

Un vector sigue una distribución k-normal multivariante si cada uno de los elementos individuales que lo forman sigue una distribución normal y lo mismo para toda combinación lineal de sus k elementos. Las siguientes imágenes muestran representaciones gráficas de distribuciones normales multivariante de 2 elementos (distribución normal bivariante).

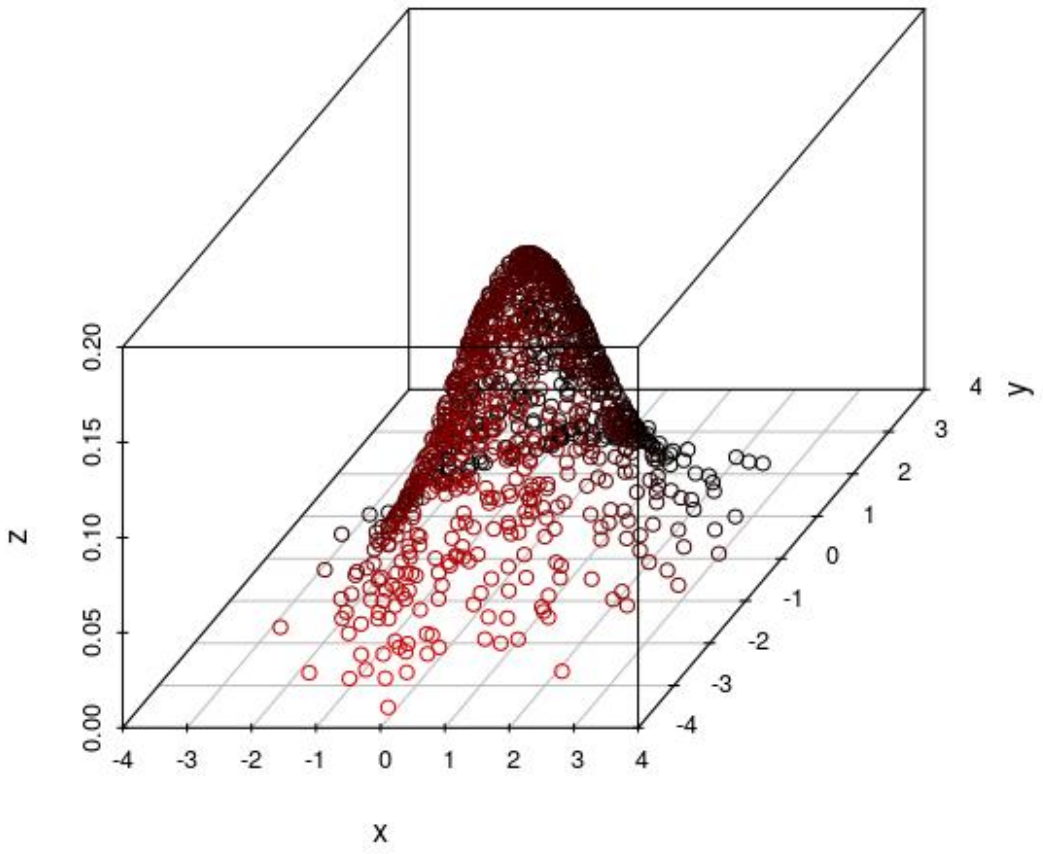
```
#Código obtenido de http://www.columbia.edu/~cjd11/charles_dimaggio/DIRE/styled-
4/styled-11/code-5/
# R code to create bivariate figure
mu1 <- 0 # set mean x1
mu2 <- 0 # set mean x2
s11 <- 10 # set variance x1
s22 <- 10 # set variance x2
s12 <- 15 # set covariance x1 and x2
rho <- 0.5 # set correlation coefficient  x1 and x2
x1 <- seq(-10,10,length = 41) # generate vector  x1
x2 <- x1 # copy x1 to x2


f <- function(x1,x2) # multivariate function
{
  term1 <- 1/(2*pi*sqrt(s11*s22*(1-rho^2)))
  term2 <- -1/(2*(1-rho^2))
  term3 <- (x1-mu1)^2/s11
  term4 <- (x2-mu2)^2/s22
  term5 <- -2*rho*((x1-mu1)*(x2-mu2))/(sqrt(s11)*sqrt(s22))
  term1*exp(term2*(term3+term4-term5))
}

z <- outer(x1,x2,f) # calculate density values


persp(x1, x2, z, # 3-D plot
  main = "Distribución multivariante con dos predictores",
  col ="lightgreen",
  theta = 30, phi = 20,
  r = 50,
  d = 0.1,
  expand = 0.5,
```

```
  ltheta = 90, lphi = 180,
  shade = 0.75,
  ticktype = "simple",
  nticks = 5)
```

Distribución multivariante con dos predictores




```
#Otra forma de representar una distribución bivariante
library(mvtnorm)
library(scatterplot3d)

sigma.zero <- matrix(c(1,0,0,1), ncol=2)
x1000 <- rmvnorm(n=1000, mean=c(0,0), sigma=sigma.zero)
scatterplot3d(x1000[,1], x1000[,2], dmvnorm(x1000, mean=c(0,0), sigma=sigma.zero),
highlight=TRUE, xlab = "x", ylab = "y", zlab = "z", )
```



Para indicar que una variable aleatoria p-dimensional X sigue una distribución normal multivariante se emplea la terminología $X \sim N(\mu, \sum)$. Donde μ es el vector promedio de X y $\sum$ es la covarianza de X, que al ser un vector con p elementos, es una matriz $p\ x\ p$ con la covarianza de cada par de predictores. La ecuación que define la función de densidad de una distribución normal multivariante es:

$$f(x) = \frac{1}{(2\pi)^{p/2}|\sum|^{1/2}}\ exp\ (-\frac{1}{2}(x-\mu)^T \sum^{-1} (x-\mu))$$

Si se sigue el mismo procedimiento que el mostrado para *LDA* con un solo predictor, pero esta vez con la ecuación de multivariante normal, y se asume que la matriz de covarianzas ∑ es igual para las *K* clases, se obtiene que el clasificador de Bayes es:

$$\hat{\delta}_k(x) = log(P(Y = k|X = x)) = x^T \sum^{-1} \mu_k - \frac{1}{2}\mu_k^T \sum^{-1} \mu_k + log(\pi_k)$$

Cuando los parámetros poblacionales se desconocen, no se puede calcular el límite de decisión de Bayes exacto, por lo que se recurre a la estimación de $\mu_1 \ldots, \mu_k$, $\pi_1 \ldots \pi_k$ y ∑ para obtener los límites de decisión de *LDA*.

Condiciones de LDA

Las condiciones que se deben cumplir para que un Análisis Discriminante Lineal sea válido son:

- Cada predictor que forma parte del modelo se distribuye de forma normal en cada una de las clases de la variable respuesta. En el caso de múltiples predictores, las observaciones siguen una distribución normal multivariante en todas las clases.
- La varianza del predictor es igual en todas las clases de la variable respuesta. En el caso de múltiples predictores, la matriz de covarianza es igual en todas las clases. Si esto no se cumple se recurre a Análisis Discriminante Cuadrático (*QDA*).

Cuando la condición de normalidad no se cumple, el LDA pierde precisión pero aun así puede llegar a clasificaciones relativamente buenas. *Using discriminant analysis for multi-class classification: an experimental investigation (Tao Li, Shenghuo Zhu, Mitsunori Ogihara)*.

Dos aproximaciones a LDA: Bayes y Fisher

Existen varios enfoques posibles para realizar un *LDA*. La aproximación descrita anteriormente está basada en el clasificador de Bayes, y utiliza todas las variables originales para calcular las probabilidades posteriores de que una observación pertenezca a cada grupo.

Antes de que el clasificador de Bayes fuese introducido en el *LDA*, Fisher propuso una aproximación en la que el espacio p-dimensional (donde p es el número de predictores originales) se reduce a un subespacio de menos dimensiones formado por las combinaciones lineales de los predictores que mejor explican la separación de las clases. Una vez encontradas dichas combinaciones se realiza la clasificación en este subespacio. Fisher definió como subespacio optimo a aquel que maximiza la distancia entre grupos en términos de varianza. Los términos de *discriminante lineal de Fisher* y *LDA* son a menudo usados para expresar la misma idea, aunque el artículo original de Fisher realmente describe un discriminante ligeramente diferente, que no hace algunas de las suposiciones del *LDA* como una distribución normal de las clases o covarianzas iguales entre clases.

La aproximación de Fisher se puede ver como un proceso con dos partes:

- Reducción de dimensionalidad: Se pasa de p variables predictoras originales a k combinaciones lineales de dichos predictores (variables discriminantes) que permiten explicar la separación de los grupos pero con menos dimensiones ($k < p$).
- Clasificación de las observaciones empleando las variables discriminantes.

Los resultados de clasificación obtenidos mediante el método de Fisher son iguales a los obtenidos por el método de Bayes cuando:

- En el método de Bayes se asume que la matriz de covarianzas en igual en todos los grupos y se emplea como estimación la *pooled within-class covariance matrix*.
- En el método de Fisher, todos los discriminantes lineales se utilizan para la clasificación. El número máximo de discriminantes obtenido tras la reducción de dimensionalidad es *número grupos*-1.

Bayes Optimality in Linear Discriminant Analysis Onur C. Hamsici and Aleix M. Martinez

Generalizing Fisher's linear discriminant analysis via the SIR approach, Chapter 14

http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema1dm.pdf

http://www2.stat.unibo.it/montanari/Didattica/Multivariate/Discriminant_analysis.pdf

Precisión del LDA

Una vez que las normas de clasificación se han establecido, se tiene que evaluar como de buena es la clasificación resultante. En otras palabras, evaluar el porcentaje de aciertos en las clasificaciones.

Las matrices de confusión son una de las mejores formas de evaluar la capacidad de acierto que tiene un modelo *LDA*. Muestran el número de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. El método *LDA* busca los límites de decisión que más se aproximan al clasificador de Bayes, que por definición tiene el menor ratio de error total de entre todos los clasificadores (si se cumple la condición de normalidad). Por lo tanto, el *LDA* intenta conseguir el menor número de clasificaciones erróneas posibles, pero no diferencia entre falsos positivos o falsos negativos. Si se quiere intentar reducir el número de errores de clasificación en una dirección determinada (por ejemplo, menos falsos negativos) se puede modificar el límite de decisión, aunque como consecuencia aumentará el número de falsos positivos.

Cuando para evaluar el error de clasificación se emplean las mismas observaciones con las que se ha creado el modelo, se obtiene lo que se denomina el *trainning error*. Si bien esta es una forma sencilla de estimar la precisión en la clasificación, tiende a ser excesivamente optimista. Es más adecuado evaluar el modelo empleando observaciones nuevas que el modelo no ha visto, obteniendo así el *test error*. En el capítulo *Validación de modelos de regresión* se describen diferentes estrategias para estimar el *test error*.

Ejemplo datos insectos

Un equipo de biólogos quiere generar un modelo estadístico que permita identificar a que especie (a o b) pertenece un determinado insecto. Para ello se han medido tres variables (longitud de las patas, diámetro del abdomen y diámetro del órgano sexual) en 10 individuos de cada una de las dos especies.

Obtención de los datos de entrenamiento

```
input <- ("
especie pata abdomen organo_sexual
a 191 131 53
a 185 134 50
a 200 137 52
a 173 127 50
a 171 128 49
a 160 118 47
a 188 134 54
a 186 129 51
a 174 131 52
a 163 115 47
b 186 107 49
b 211 122 49
b 201 144 47
b 242 131 54
b 184 108 43
b 211 118 51
b 217 122 49
b 223 127 51
b 208 125 50
b 199 124 46
")
datos <- read.table(textConnection(input), header = TRUE)
```

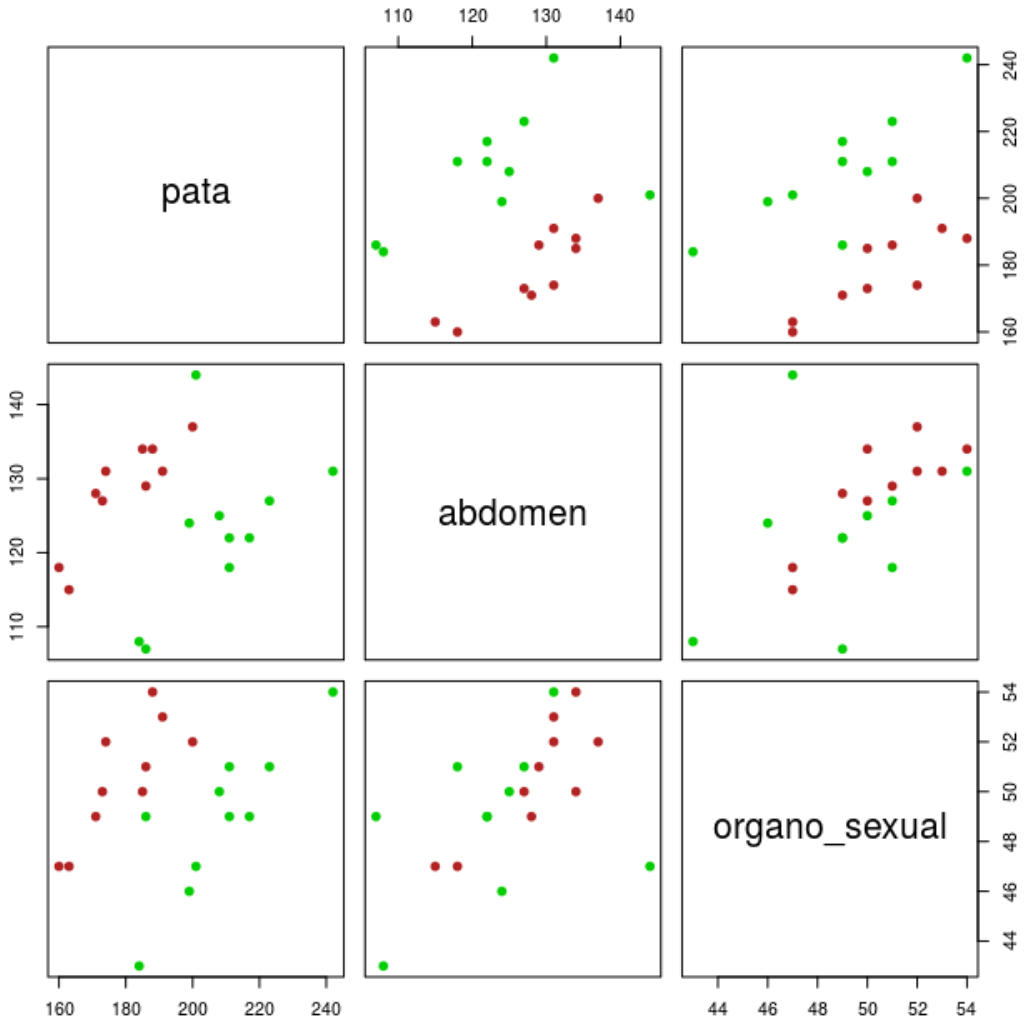
Exploración gráfica de los datos

```
library(ggplot2)
library(gridExtra)
p1 <- ggplot(data = datos, aes(x = pata, fill = especie)) + geom_histogram(position
= "identity", alpha = 0.5)
p2 <- ggplot(data = datos, aes(x = abdomen, fill = especie)) +
```

```
  geom_histogram(position = "identity", alpha = 0.5)
p3 <- ggplot(data = datos, aes(x = organo_sexual, fill = especie)) +
  geom_histogram(position = "identity", alpha = 0.5)
grid.arrange(p1, p2, p3)
```

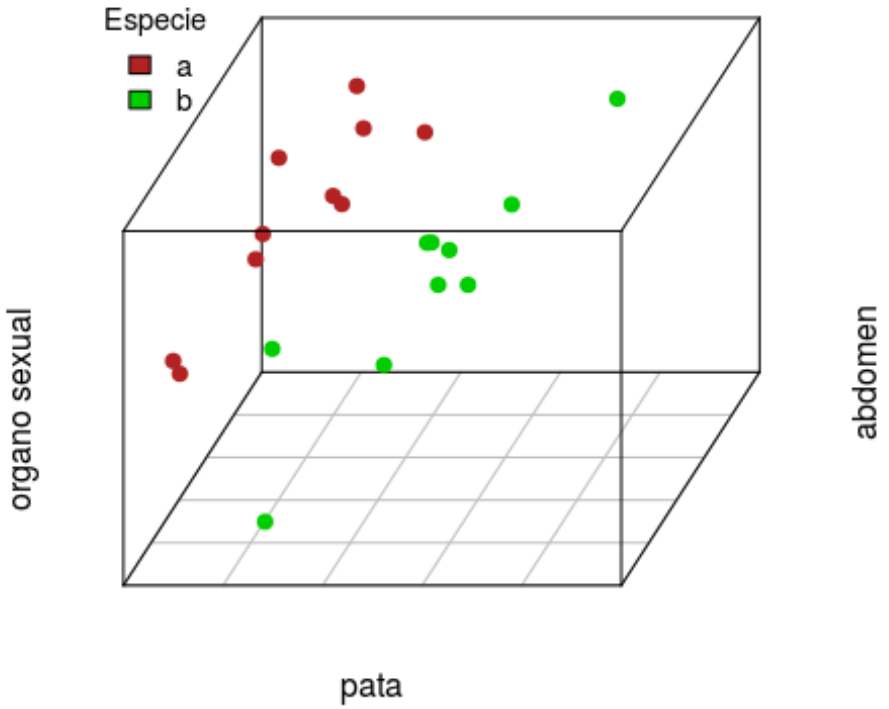
A nivel individual, la longitud de la pata parece ser la variable que más se diferencia entre especies (menor solapamiento entre poblaciones).

```
pairs(x = datos[, c("pata", "abdomen", "organo_sexual")], col = c("firebrick",
    "green3")[datos$especie], pch = 19)
```



El par de variables abdomen-pata y el par pata-organo_sexual parecen separar bien las dos especies.

```
library(scatterplot3d)
scatterplot3d(datos$pata, datos$abdomen, datos$organo_sexual, color =
c("firebrick", "green3")[datos$especie], pch = 19, grid = TRUE, tick.marks = FALSE,
xlab = "pata", ylab = "abdomen", zlab = "organo sexual", angle = 65)


legend("topleft",# location and inset
    bty = "n", cex = .9, # suppress legend box, shrink text 50%
    title = "Especie",
    c("a", "b"), fill = c("firebrick", "green3"))
```



La representación de las tres variables de forma simultanea parece indicar que las dos especies sí están bastante separadas en el espacio 3D generado.

Prior probabilities

Como no se dispone de información sobre la abundancia relativa de las especies a nivel poblacional, se considera como probabilidad previa de cada especie el número de observaciones de la especie entre el número de observaciones totales.

$$\hat{\pi_a} = \hat{\pi_b} = \frac{10}{20}$$

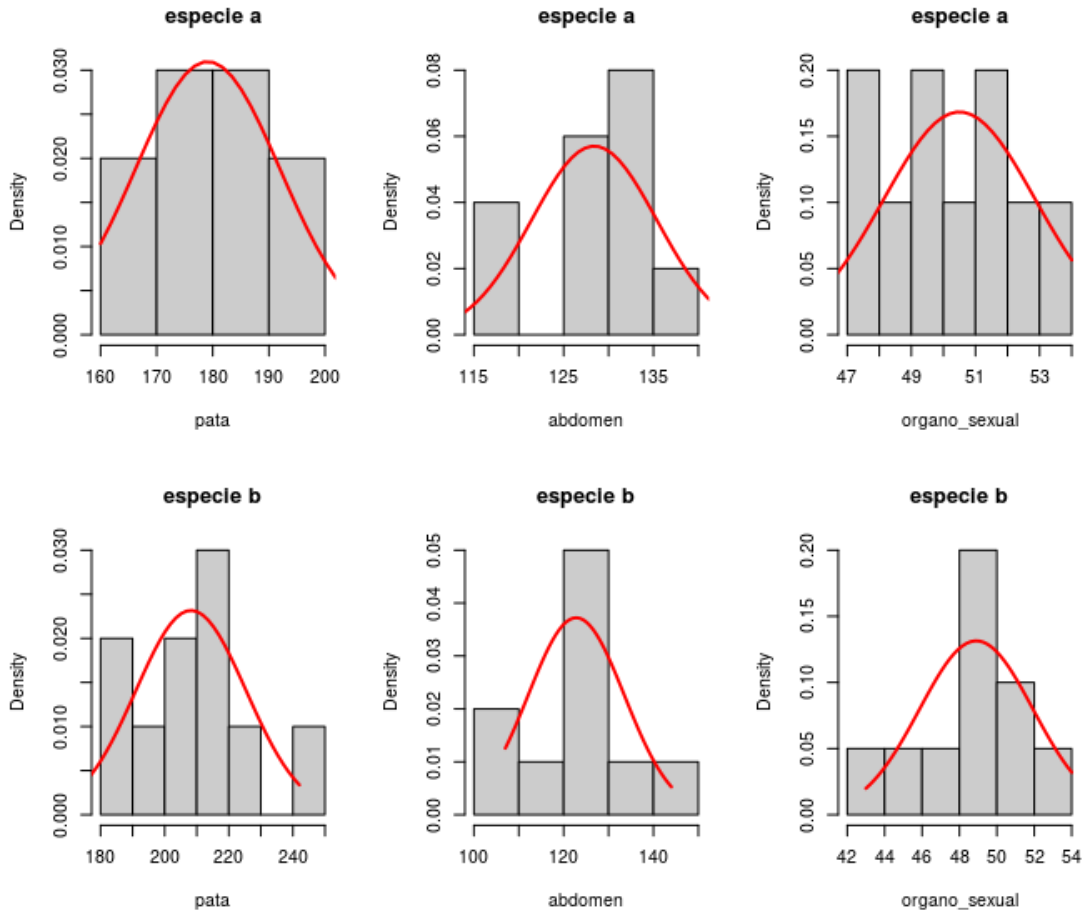
Homogeneidad de Varianza

De entre los diferentes test que contrastan la homogeneidad de varianza (https://rpubs.com/Joaquin_AR/218466), el más recomendable cuando solo hay un predictor, dado que se asume que se distribuye de forma normal, es el test de *Barttlet*. Cuando se emplean múltiples predictores, se tiene que contrastar que la matriz de covarianzas (∑) es constante en todos los grupos, siendo recomendable comprobar también la homogeneidad de varianza para cada predictor a nivel individual.

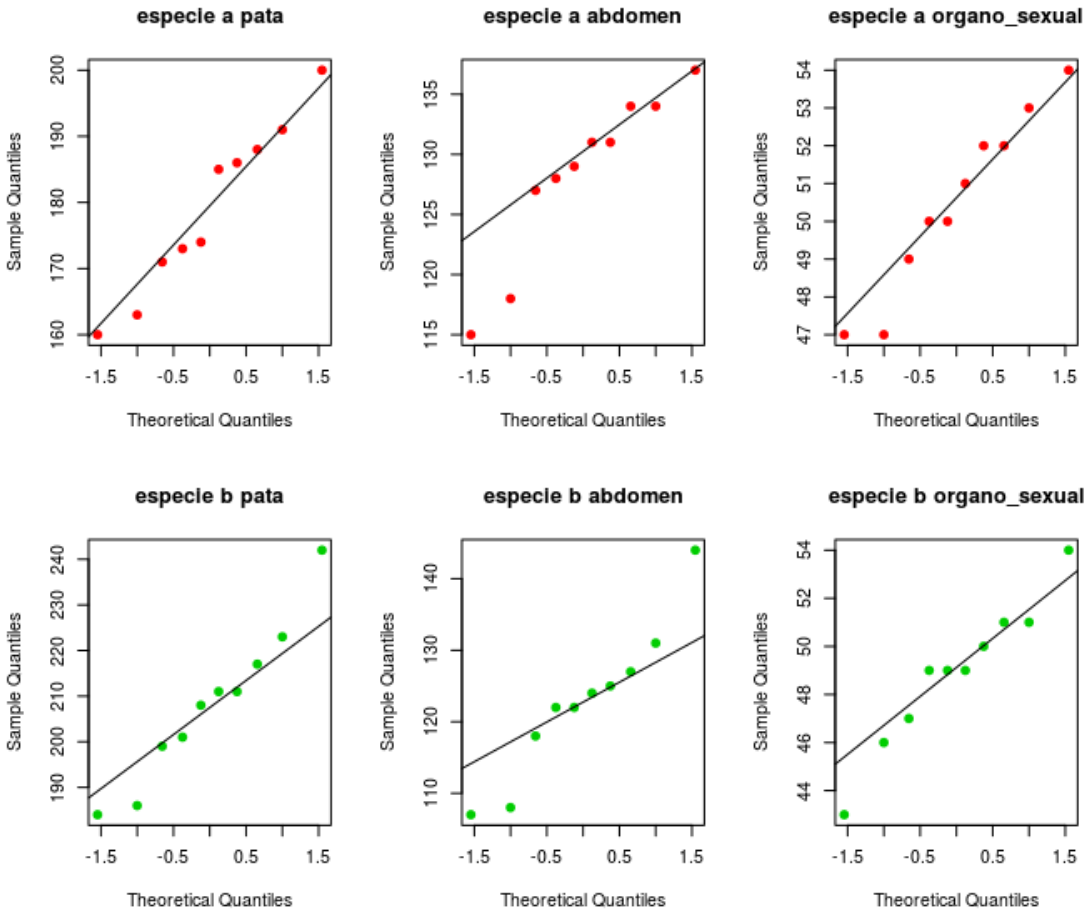
El test *Box M* fue desarrollado por el matemático Box (1949) como una extensión del test de *Barttlet* para escenarios multivariante y que permite contrastar la igualdad de matrices entre grupos. El test *Box M* es muy sensible a violaciones de la normalidad multivariante, por lo que esta debe ser contrastada con anterioridad. Ocurre con frecuencia, que el resultado de un test *Box M* resulta significativo debido a la falta de distribución normal multivariante en lugar de por falta de homogeneidad en las matrices de covarianza. Dada la sensibilidad del test se recomienda emplear un límite de significancia de 0.001 (Tabachnick & Fidell, 2001, y http://www.real-statistics.com/multivariate-statistics/).

Distribución de los predictores de forma individual:

```
# representación mediante Histograma de cada variable para cada especie
par(mfcol = c(2, 3))
for (k in 2:4) {
    j0 <- names(datos)[k]
    # br0 <- seq(min(datos[, k]), max(datos[, k]), le = 11)
    x0 <- seq(min(datos[, k]), max(datos[, k]), le = 50)
    for (i in 1:2) {
        i0 <- levels(datos$especie)[i]
        x <- datos[datos$especie == i0, j0]
        hist(x, proba = T, col = grey(0.8), main = paste("especie", i0), xlab = j0)
        lines(x0, dnorm(x0, mean(x), sd(x)), col = "red", lwd = 2)
    }
}
```




```
# representación de cuantiles normales de cada variable para cada especie
for (k in 2:4) {
    j0 <- names(datos)[k]
    x0 <- seq(min(datos[, k]), max(datos[, k]), le = 50)
    for (i in 1:2) {
        i0 <- levels(datos$especie)[i]
        x <- datos[datos$especie == i0, j0]
        qqnorm(x, main = paste("especie", i0, j0), pch = 19, col = i + 1)  # los
colores 2 y 3 son el rojo y verde
        qqline(x)
    }
}
```




```
# Contraste de normalidad Shapiro-Wilk para cada variable en cada especie
library(reshape2)
library(knitr)
library(dplyr)
datos_tidy <- melt(datos, value.name = "valor")
kable(datos_tidy %>% group_by(especie, variable) %>% summarise(p_value_Shapiro.test
= shapiro.test(valor)$p.value))
```

especie	variable	p_value_Shapiro.test
a	pata	0.7763034
a	abdomen	0.1845349
a	organo_sexual	0.6430844
b	pata	0.7985711
b	abdomen	0.5538213
b	organo_sexual	0.8217855

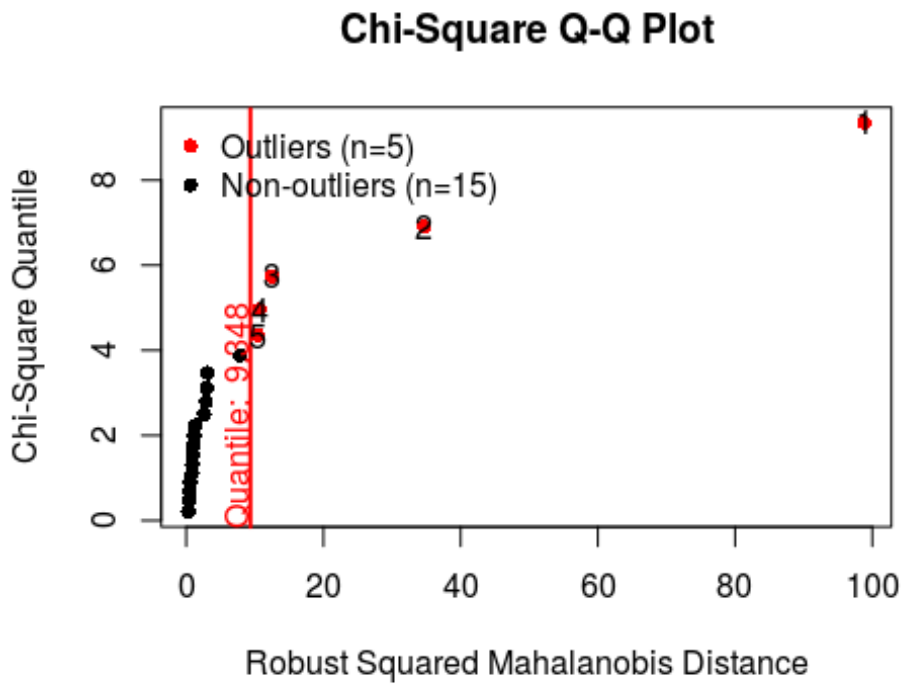
```
# misma operación con aggregate
aggregate(formula = valor ~ especie + variable, data = datos_tidy, FUN =
            function(x) {
                         shapiro.test(x)$p.value
                        }
)
```

No hay evidencias de falta de normalidad univariante en ninguna de las variables empleadas como predictores, en ninguno de los grupos.

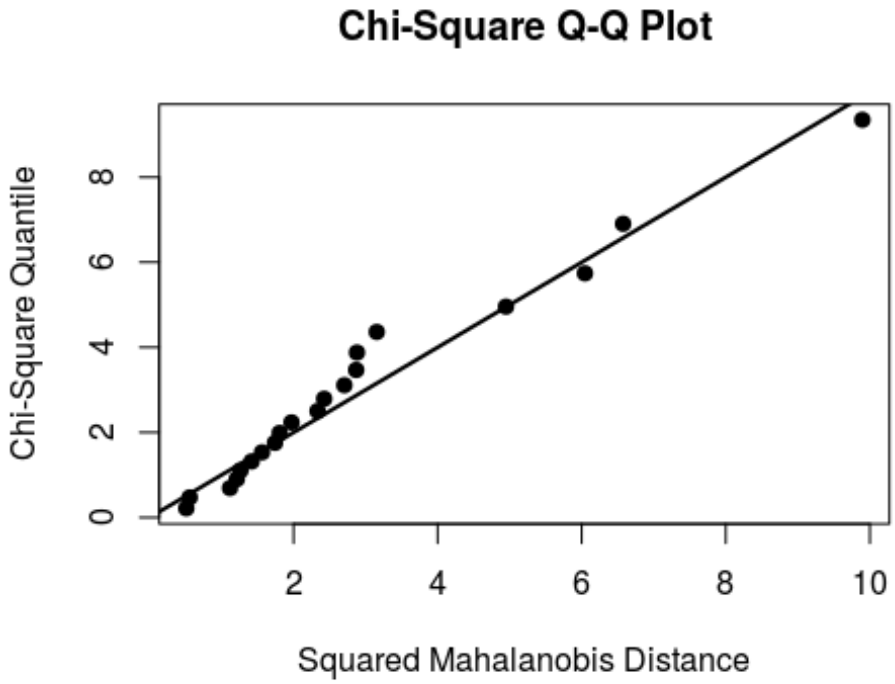
El paquete *MVN* contiene funciones que permiten realizar los tres test de hipótesis comúnmente empleados para evaluar la normalidad multivariante (*Mardia, Henze-Zirkler y Royston*) y también funciones para identificar *outliers* que puedan influenciar en el contraste. Para información detallada de cada uno consultar:

https://cran.r-project.org/web/packages/MVN/vignettes/MVN.pdf.

```
library(MVN)

outliers <- mvOutlier(datos[,-1], qqplot = TRUE, method = "quan")
```



```
roystonTest(data = datos[,-1], qqplot = TRUE)
```




```
##   Royston's Multivariate Normality Test
## ---------------------------------------------
##   data : datos[, -1]
##
##   H       : 0.4636176
##   p-value : 0.9299447
##
##   Result  : Data are multivariate normal.
## ---------------------------------------------
```

```
hzTest(data = datos[,-1], qqplot = FALSE)
```

```
##   Henze-Zirkler's Multivariate Normality Test
## ---------------------------------------------
##   data : datos[, -1]
##
##   HZ      : 0.7870498
##   p-value : 0.07666139
##
##   Result  : Data are multivariate normal.
## ---------------------------------------------
```

A pesar de los 5 *outliers* detectados, ninguno de los dos test encuentran evidencias significativas ($\alpha = 0.05$) de falta de normalidad multivariante. Finalmente, mediante la función `boxM()` del paquete *biotools* se realiza el contraste de matrices de covarianza.

```
library(biotools)
boxM(data = datos[, 2:4], grouping = datos[, 1])
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  datos[, 2:4]
## Chi-Sq (approx.) = 9.831, df = 6, p-value = 0.132
```

Se puede aceptar que la matriz de covarianza es igual en todos los grupos.

Estimación de los parámetros de la función de densidad ($\hat{\mu}(X), \sum$) y cálculo de la función discriminante.

Estos dos pasos se realizan mediante la función `lda()` del paquete *MASS*. `lda()` realiza la clasificación mediante la aproximación de Fisher.

```
modelo_lda <- lda(formula = especie ~ pata + abdomen + organo_sexual, data = datos)
```

Una vez obtenidas las funciones discriminantes, se puede clasificar un nuevo insecto en función de sus medidas. Por ejemplo, un nuevo espécimen cuyas medidas sean: pata = 194, abdomen = 124, organo_sexual = 49.

```
nuevas_observaciones <- data.frame(pata = 194, abdomen = 124, organo_sexual = 49)
predict(object = modelo_lda, newdata = nuevas_observaciones)
```

```
## $class
## [1] b
## Levels: a b
## $posterior
##            a         b
## 1 0.05823333 0.9417667
##
## $x
##         LD1
## 1 0.5419421
```

El resultado muestra que, según la función discriminante, la probabilidad posterior de que el espécimen pertenezca a la especie b es del 94.2% frente al 5.8% de que pertenezca a la especie a.

Evaluación de los errores de clasificación.

```
predicciones <- predict(object = modelo_lda, newdata = datos[, -1], method =
"predictive")
table(datos$especie, predicciones$class, dnn = c("Clase real", "Clase predicha"))
```

```
##            Clase predicha
## Clase real  a  b
##          a 10  0
##          b  0 10
```

```
trainig_error <- mean(datos$especie != predicciones$class) * 100
paste("trainig_error=", trainig_error, "%")
```

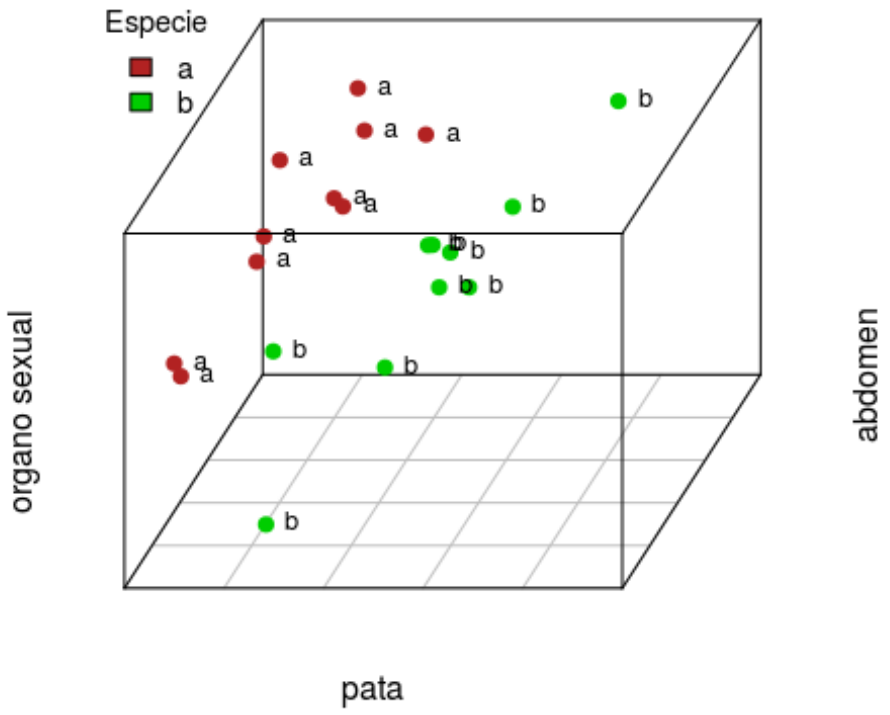
```
## [1] "trainig_error= 0 %"
```

Empleando las mismas observaciones con las que se ha generado el modelo discriminante (*trainig data*), la precisión de clasificación es del 100%. Evaluar un modelo con los mismos datos con los que se ha creado suele resultar en estimaciones de la precisión demasiado optimistas (*test error muy bajo*). Como se describe en el capítulo *Validación de modelos de regresión*, la estimación del *test error* mediante validación cruzada es más adecuada.

La siguiente imagen muestra la representación de las observaciones, coloreadas por la verdadera especie a la que pertenecen y acompañadas por una etiqueta con la especie que ha predicho el *LDA*.

```
with(datos, {
  s3d <- scatterplot3d(pata, abdomen, organo_sexual, color = c("firebrick",
"green3")[datos$especie], pch = 19, grid = TRUE, tick.marks = FALSE, xlab = "pata",
ylab = "abdomen", zlab = "organo sexual", angle = 65)

  s3d.coords <- s3d$xyz.convert(pata, abdomen, organo_sexual) # convierte
coordenadas 3D en proyecciones 2D
```

```
    text(s3d.coords$x, s3d.coords$y,   # cordenadas x, y
         labels = datos$especie,    # texto
         cex = .8, pos = 4)

    legend("topleft",
    bty = "n", cex = .9,
    title = "Especie",
    c("a", "b"), fill = c("firebrick", "green3"))
})
```



Ejemplo con *Iris data*

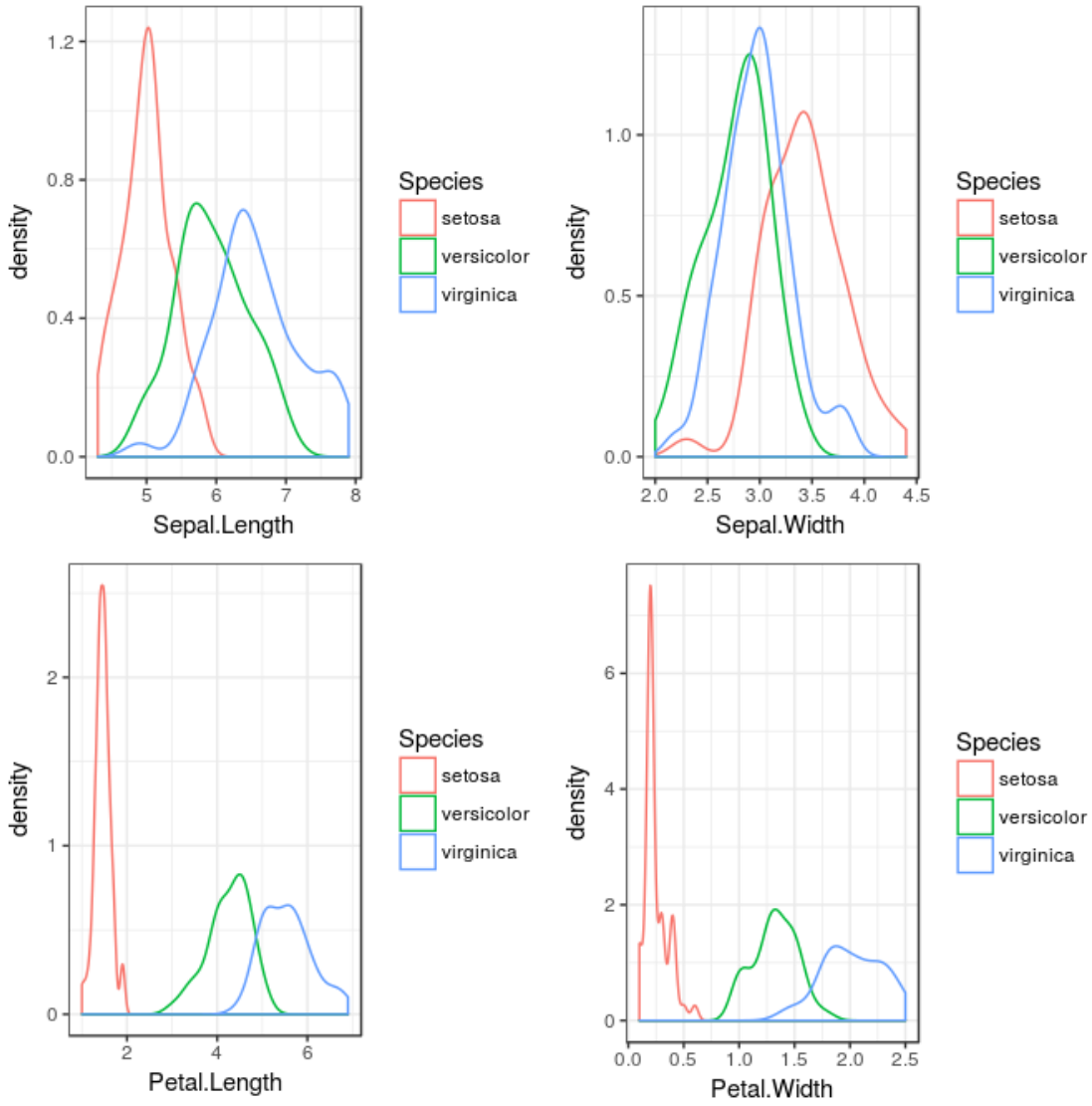
El set de datos *Iris* contiene métricas de 150 flores de 3 especies diferentes de planta Iris. Para cada flor se han registrado 4 variables: sepal length, sepal width, petal length y petal width, todas ellas en centímetros. Se desea generar un modelo discriminante que permita clasificar las flores en las distintas especies empleando las variables mencionadas.

```
data("iris")
kable(head(iris, n = 3))
```

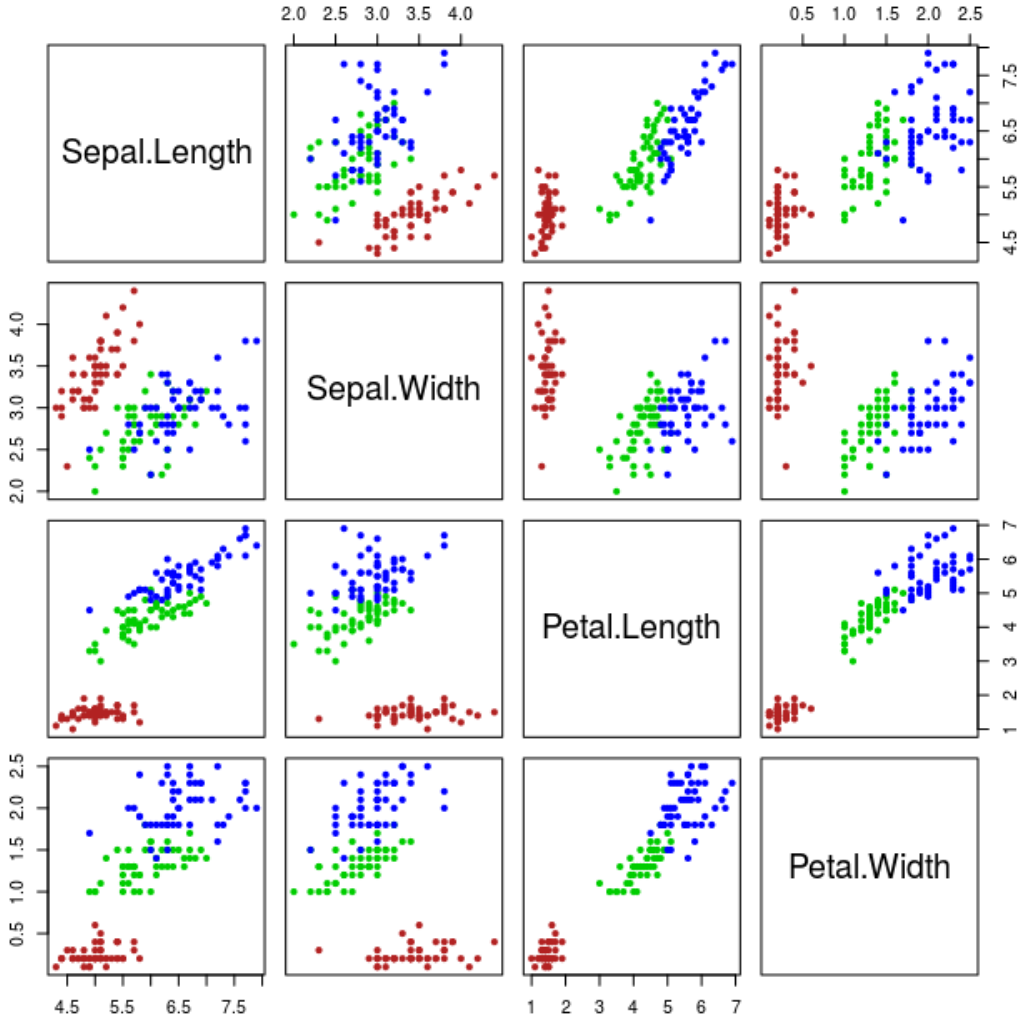
	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa

Exploración gráfica de los datos

```
library(ggplot2)
library(gridExtra)

plot1 <- ggplot(data = iris, aes(x = Sepal.Length)) +
 geom_density(aes(colour = Species)) +
 theme_bw()

plot2 <- ggplot(data = iris, aes(x = Sepal.Width)) +
  geom_density(aes(colour = Species)) +
  theme_bw()

plot3 <- ggplot(data = iris, aes(x = Petal.Length)) +
  geom_density(aes(colour = Species)) +
  theme_bw()

plot4 <- ggplot(data = iris, aes(x = Petal.Width)) +
 geom_density(aes(colour = Species)) +
 theme_bw()

grid.arrange(plot1, plot2, plot3, plot4)
```




```
pairs(x = iris[, -5], col = c("firebrick", "green3", "blue")[iris$Species], pch =
20)
```



Las variables *Petal.Lenght* y *Petal.Width* son las dos variables con más potencial para poder separar entre clases. Sin embargo, están altamente correlacionadas, por lo que la información que aportan es en gran medida redundante.

Prior probabilities

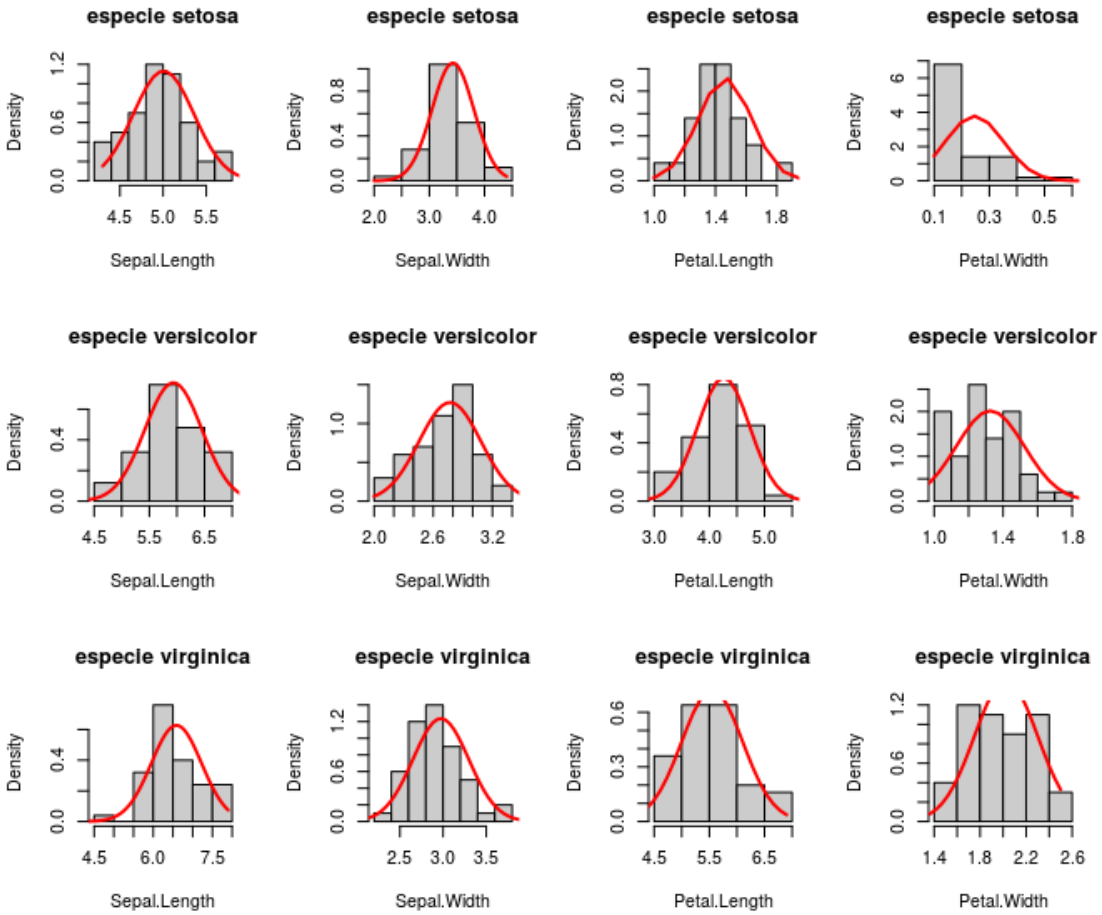
Como no se dispone de información sobre la abundancia relativa de las especies a nivel poblacional, se considera como probabilidad previa de cada especie el número de observaciones de la especie entre el número de observaciones totales.

$$\hat{\pi_a} = \hat{\pi_b} = \frac{50}{150}$$

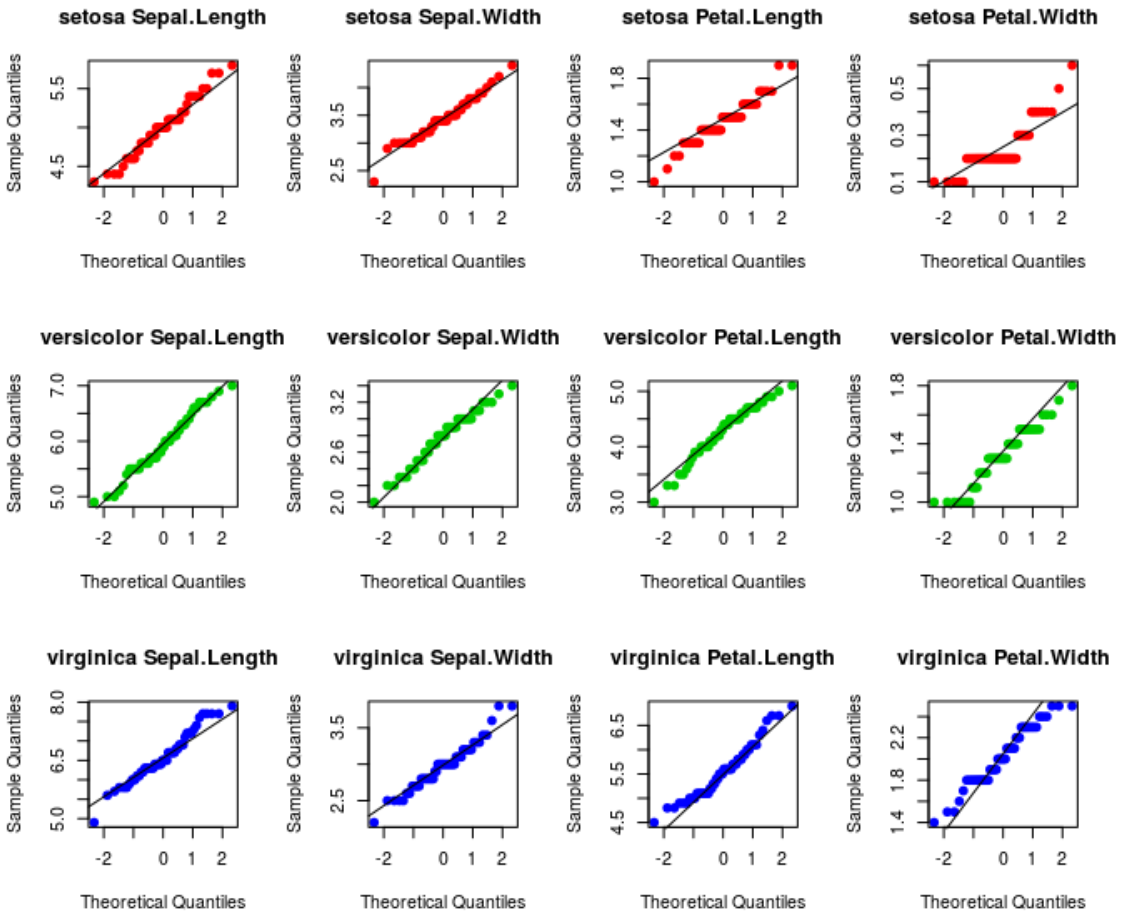
Normalidad univariante, normalidad multivariante y homogeneidad de varianza

Distribución de los predictores de forma individual:

```
# representación mediante Histograma de cada variable para cada especie
par(mfcol = c(3, 4))
for (k in 1:4) {
    j0 <- names(iris)[k]
    x0 <- seq(min(iris[, k]), max(iris[, k]), le = 50)
    for (i in 1:3) {
        i0 <- levels(iris$Species)[i]
        x <- iris[iris$Species == i0, j0]
        hist(x, proba = T, col = grey(0.8), main = paste("especie", i0), xlab = j0)
        lines(x0, dnorm(x0, mean(x), sd(x)), col = "red", lwd = 2)
    }
}
```



```
# representación de cuantiles normales de cada variable para cada especie
for (k in 1:4) {
    j0 <- names(iris)[k]
    x0 <- seq(min(iris[, k]), max(iris[, k]), le = 50)
    for (i in 1:3) {
        i0 <- levels(iris$Species)[i]
        x <- iris[iris$Species == i0, j0]
        qqnorm(x, main = paste(i0, j0), pch = 19, col = i + 1)  # los colores 2 y 3
son el rojo y verde
        qqline(x)
    }
}
```



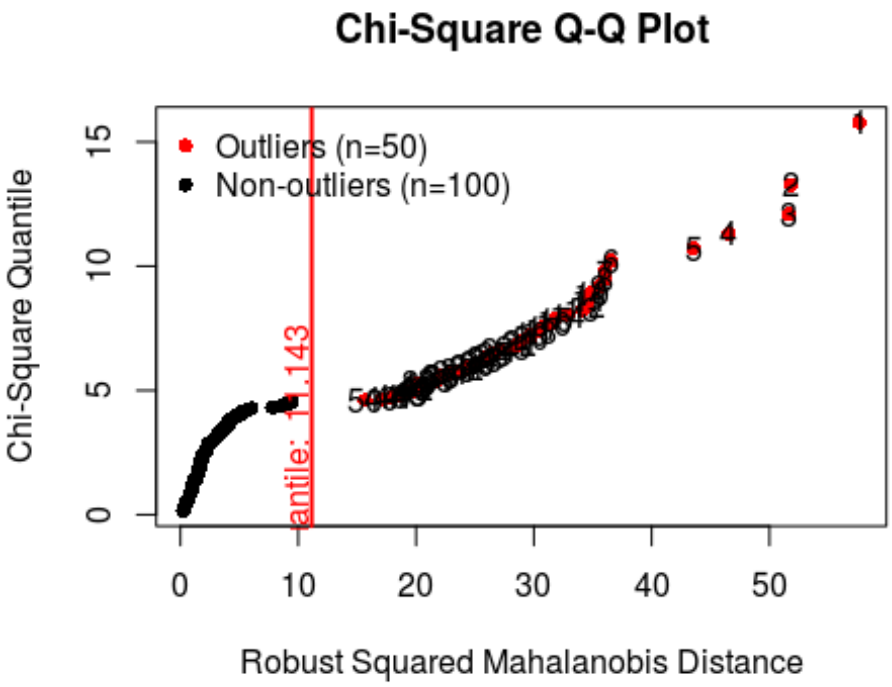
```
# Contraste de normalidad Shapiro-Wilk para cada variable en cada especie
library(reshape2)
library(knitr)
library(dplyr)
datos_tidy <- melt(iris, value.name = "valor")
kable(datos_tidy %>% group_by(Species, variable) %>% summarise(p_value_Shapiro.test
= round(shapiro.test(valor)$p.value,
    5)))
```

Species	variable	p_value_Shapiro.test
setosa	Sepal.Length	0.45951
setosa	Sepal.Width	0.27153
setosa	Petal.Length	0.05481
setosa	Petal.Width	0.00000
versicolor	Sepal.Length	0.46474
versicolor	Sepal.Width	0.33800
versicolor	Petal.Length	0.15848
versicolor	Petal.Width	0.02728
virginica	Sepal.Length	0.25831
virginica	Sepal.Width	0.18090
virginica	Petal.Length	0.10978
virginica	Petal.Width	0.08695

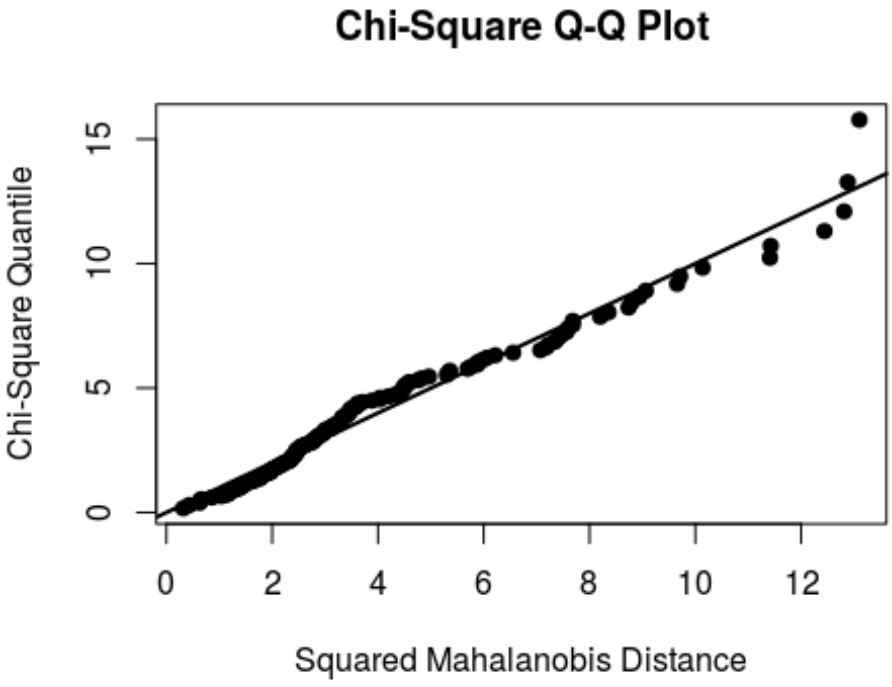
La variable *petal.width* no se distribuye de forma normal en los grupos setosa y versicolor.

Normalidad multivariante:

```
library(MVN)
outliers <- mvOutlier(iris[,-5], qqplot = TRUE, method = "quan")
```




```
roystonTest(data = iris[,-5], qqplot = TRUE)
```



```
##   Royston's Multivariate Normality Test
## ---------------------------------------------
##   data : iris[, -5]
##
##   H       : 50.39667
##   p-value : 3.098229e-11
##
##   Result  : Data are not multivariate normal.
## ---------------------------------------------
```

```
hzTest(data = iris[,-5], qqplot = FALSE)
```

```
##   Henze-Zirkler's Multivariate Normality Test
## ---------------------------------------------
##   data : iris[, -5]
##
##   HZ      : 2.336394
##   p-value : 0
##
##   Result  : Data are not multivariate normal.
## ---------------------------------------------
```

Ambos test muestran evidencias significativas de falta de normalidad multivariante. El LDA tiene cierta robustez frente a la falta de normalidad multivariante, pero es importante tenerlo en cuenta en la conclusión del análisis.

```
library(biotools)
boxM(data = iris[, -5], grouping = iris[, 5])
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  iris[, -5]
## Chi-Sq (approx.) = 140.94, df = 20, p-value < 2.2e-16
```

El test Box's M muestra evidencias de que la matriz de covarianza no es constante en todos los grupos, lo que descartaría el método *LDA* en favor del *QDA*. Sin embargo, este test es muy sensible a la falta de normalidad multivariante, cosa que ocurre para los datos de *Iris*, por lo que *LDA* puede alcanzar una buena precisión en la clasificación. En la evaluación del modelo se verá como de buena es esta aproximación.

Cálculo de la función discriminante

```
library(MASS)
modelo_lda <- lda(Species ~ Sepal.Width + Sepal.Length + Petal.Length +
Petal.Width,
    data = iris)
modelo_lda
```

```
## Call:
## lda(Species ~ Sepal.Width + Sepal.Length + Petal.Length + Petal.Width,
##     data = iris)
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
##  0.3333333  0.3333333  0.3333333
##
## Group means:
##            Sepal.Width Sepal.Length Petal.Length Petal.Width
## setosa           3.428        5.006        1.462       0.246
## versicolor       2.770        5.936        4.260       1.326
## virginica        2.974        6.588        5.552       2.026
##
## Coefficients of linear discriminants:
##                     LD1         LD2
## Sepal.Width   1.5344731  2.16452123
## Sepal.Length  0.8293776  0.02410215
## Petal.Length -2.2012117 -0.93192121
## Petal.Width  -2.8104603  2.83918785
##
## Proportion of trace:
##    LD1    LD2
## 0.9912 0.0088
```

Evaluación de los errores de clasificación

```
predicciones <- predict(object = modelo_lda, newdata = iris[, -5])
table(iris$Species, predicciones$class, dnn = c("Clase real", "Clase predicha"))
```

```
##             Clase predicha
## Clase real   setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          1        49
```

```
trainig_error <- mean(iris$Species != predicciones$class) * 100
paste("trainig_error=", trainig_error, "%")
```

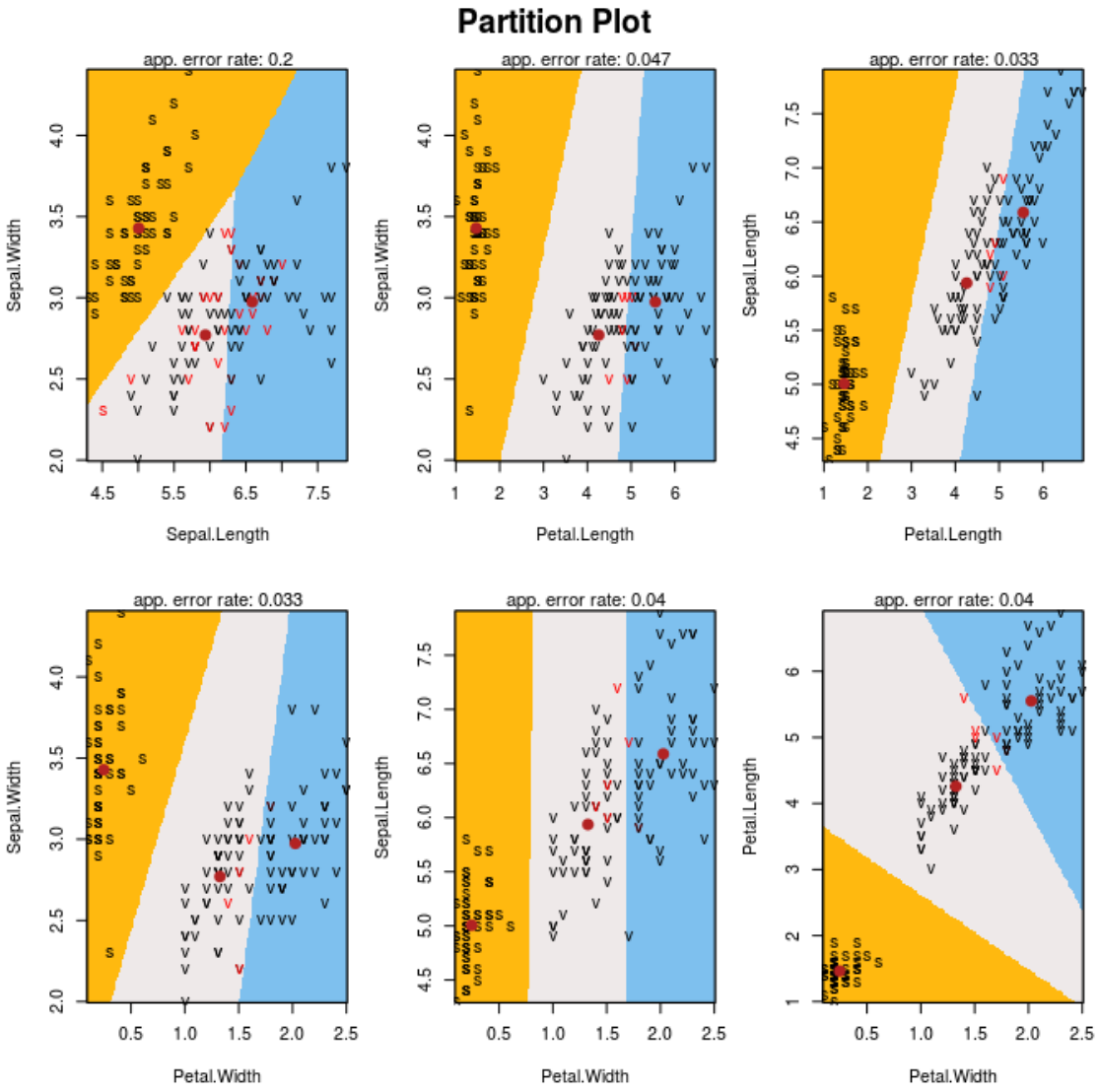
```
## [1] "trainig_error= 2 %"
```

Solo 3 de las 150 predicciones que ha realizado el modelo han sido erróneas. El *trainig error* es muy bajo (2%) lo que apunta a que el modelo es bueno. Sin embargo, para validarlo es necesario un nuevo set de datos con el que calcular el *test error* o recurrir a validación cruzada.

Visualización de las clasificaciones

La función `partimat()` del paquete *klar* permite representar los límites de clasificación de un modelo discriminante lineal o cuadrático para cada par de predictores. Cada color representa una región de clasificación acorde al modelo, se muestra el centroide de cada región y el valor real de las observaciones.

```
library(klaR)
partimat(Species ~ Sepal.Width + Sepal.Length + Petal.Length + Petal.Width,
    data = iris, method = "lda", prec = 200, image.colors = c("darkgoldenrod1",
    "snow2", "skyblue2"), col.mean = "firebrick")
```


Partition Plot
app. error rate: 0.2
Sepal.Width
Sepal.Length
app. error rate: 0.047
Sepal.Width
Petal.Length
app. error rate: 0.033
Sepal.Length
Petal.Length
app. error rate: 0.033
Sepal.Width
Petal.Width
app. error rate: 0.04
Sepal.Length
Petal.Width
app. error rate: 0.04
Petal.Length
Petal.Width

Análisis Discriminante Cuadrático

Idea intuitiva

El clasificador cuadrático o *Quadratic Discriminat Analysis QDA* se asemeja en gran medida al *LDA*, con la única diferencia de que el *QDA* considera que cada clase k tiene su propia matriz de covarianza ($\sum_k$) y como consecuencia la función discriminante toma forma cuadrática:

$$log(P(Y = k|X = x)) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) + \log(\pi_k)$$

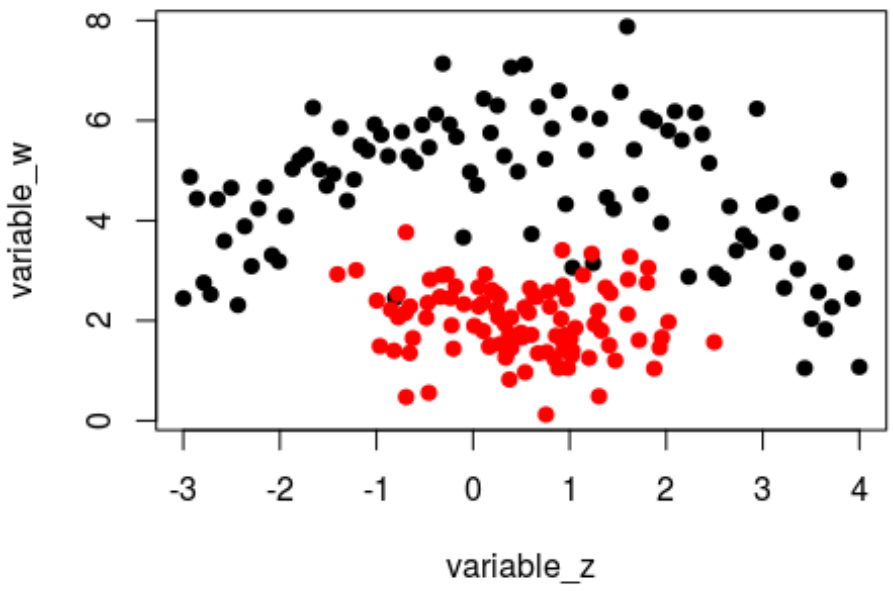
Para poder calcular la *posterior probability* a partir de esta ecuación discriminante es necesario estimar para cada clase ($\sum_k$), μ_k y π_k a partir de las muestras. Cada nueva observación se clasifica en aquella clase para la que el valor de la *posterior probability* sea mayor.

QDA genera límites de decisión curvos por lo que puede aplicarse a situaciones en las que la separación entre grupos no es lineal.

Ejemplo QDA 2 predictores

Se dispone de los siguientes datos simulados.

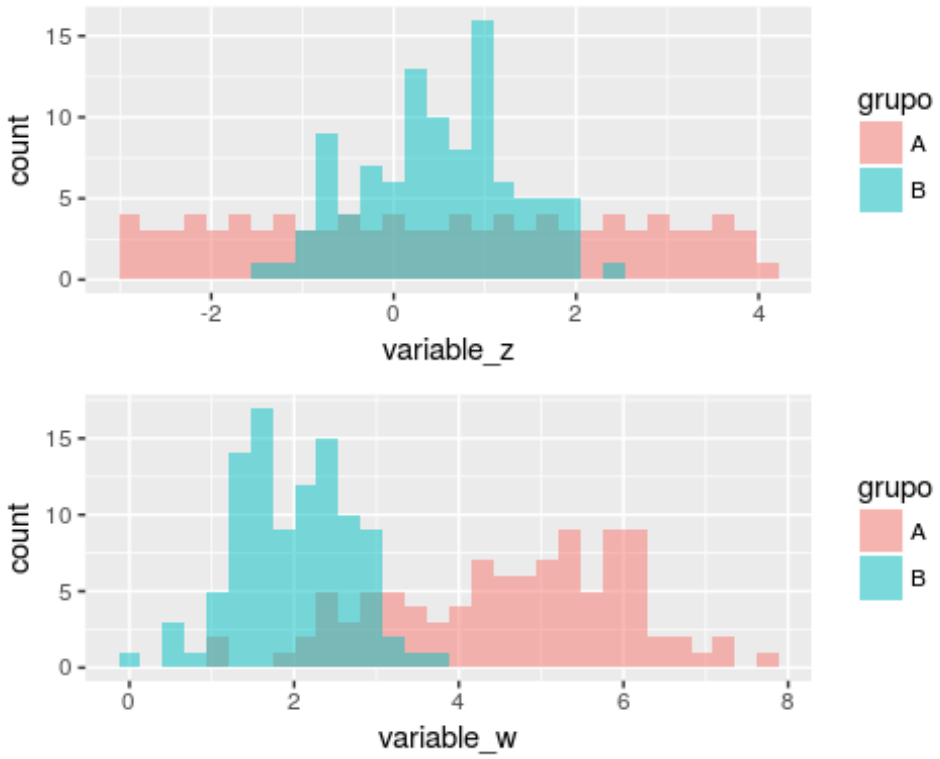
```
set.seed(8558)
grupoA_x <- seq(from = -3, to = 4, length.out = 100)
grupoA_y <- 6 + 0.15 * grupoA_x - 0.3 * grupoA_x^2 + rnorm(100, sd = 1)
grupoA <- data.frame(variable_z = grupoA_x, variable_w = grupoA_y, grupo = "A")

grupoB_x <- rnorm(n = 100, mean = 0.5, sd = 0.8)
grupoB_y <- rnorm(n = 100, mean = 2, sd = 0.8)
grupoB <- data.frame(variable_z = grupoB_x, variable_w = grupoB_y, grupo = "B")

datos <- rbind(grupoA, grupoB)
plot(datos[, 1:2], col = datos$grupo, pch = 19)
```



La separación entre los grupos no es de tipo lineal, sino que muestra cierta curvatura. En este tipo de escenarios el método *QDA* es más adecuado que el *LDA*.

Exploración gráfica de los datos

```
library(ggplot2)
library(gridExtra)
p1 <- ggplot(data = datos, aes(x = variable_z, fill = grupo)) +
geom_histogram(position = "identity",
    alpha = 0.5)
p2 <- ggplot(data = datos, aes(x = variable_w, fill = grupo)) +
geom_histogram(position = "identity",
    alpha = 0.5)
grid.arrange(p1, p2)
```

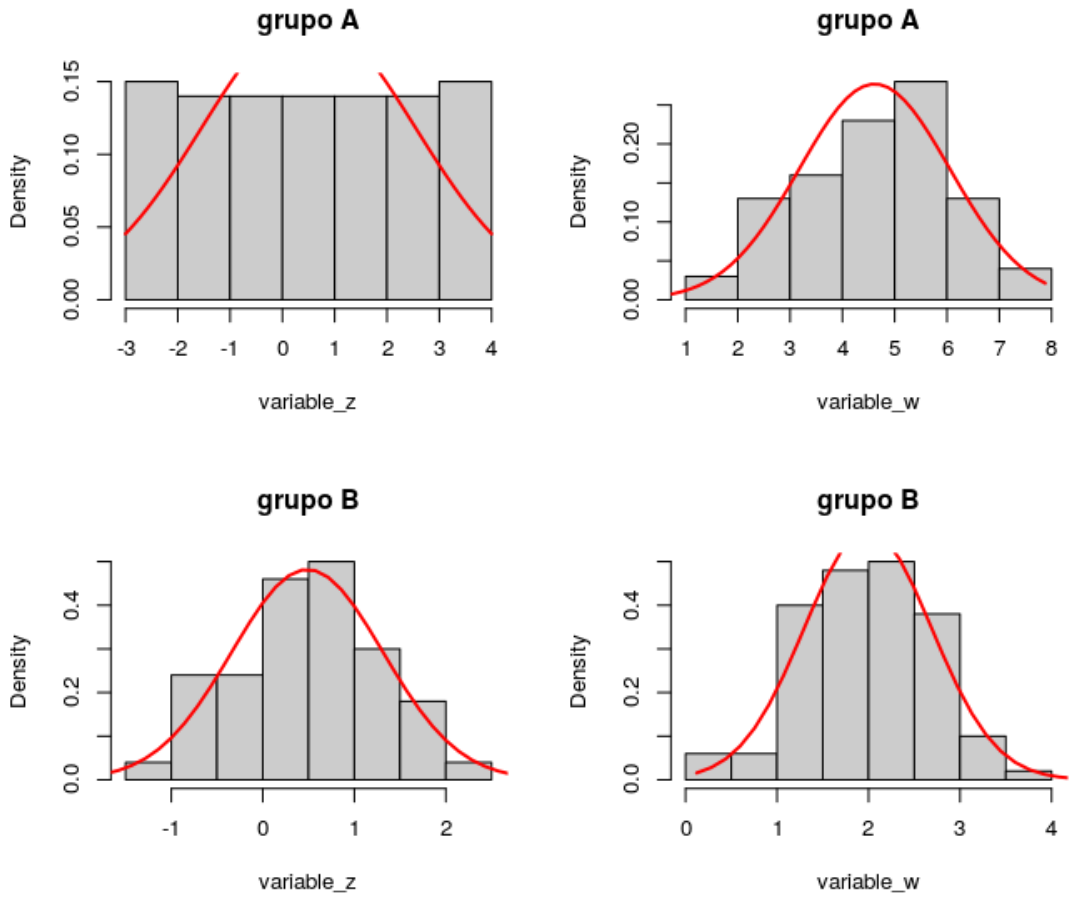


La variable *W* permite discriminar entre grupos mejor que la variable *z*.

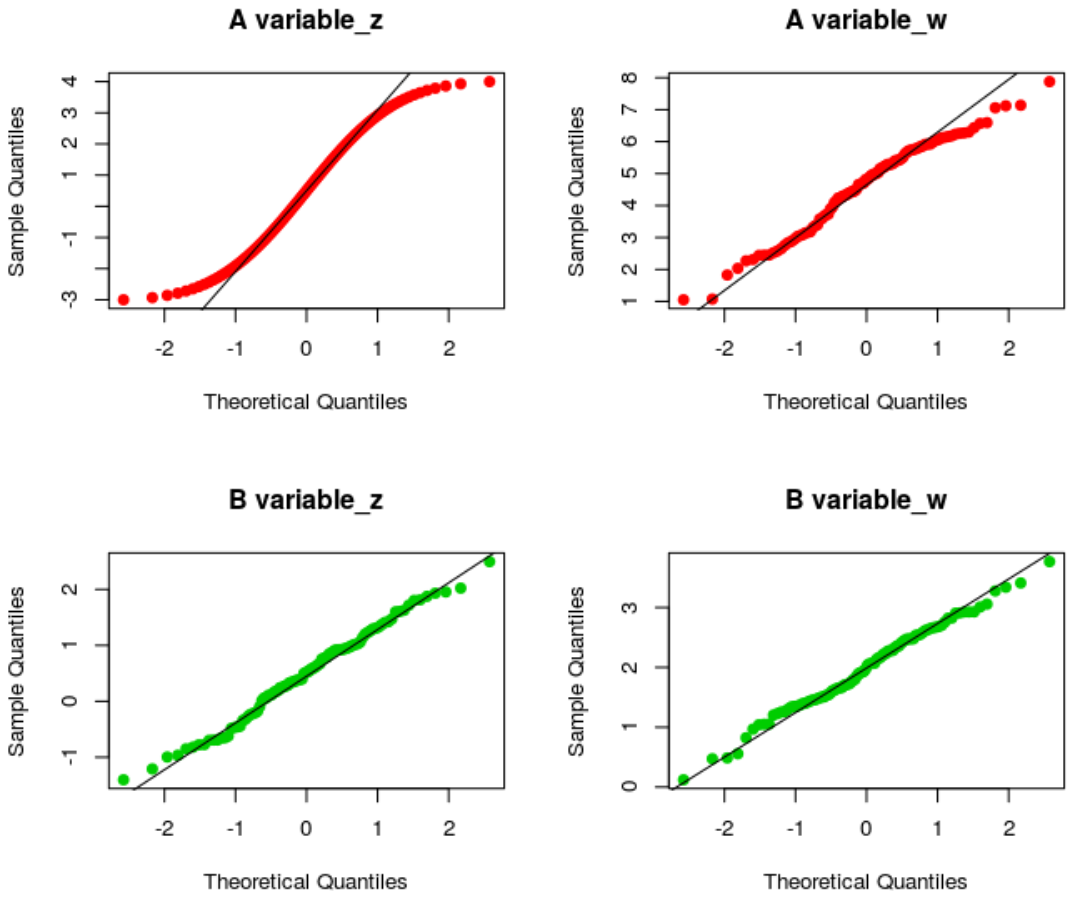
Normalidad univariante, normalidad multivariante y homogeneidad de varianza

Distribución de los predictores de forma individual:

```
# representación mediante Histograma de cada variable para cada especie
par(mfcol = c(2, 2))
for (k in 1:2) {
    j0 <- names(datos)[k]
    x0 <- seq(min(datos[, k]), max(datos[, k]), le = 50)
    for (i in 1:2) {
        i0 <- levels(datos$grupo)[i]
        x <- datos[datos$grupo == i0, j0]
        hist(x, proba = T, col = grey(0.8), main = paste("grupo", i0), xlab = j0)
        lines(x0, dnorm(x0, mean(x), sd(x)), col = "red", lwd = 2)
    }
}
```




```
# representación de cuantiles normales de cada variable para cada especie
for (k in 1:2) {
    j0 <- names(datos)[k]
    x0 <- seq(min(datos[, k]), max(datos[, k]), le = 50)
    for (i in 1:2) {
        i0 <- levels(datos$grupo)[i]
        x <- datos[datos$grupo == i0, j0]
        qqnorm(x, main = paste(i0, j0), pch = 19, col = i + 1)  # los colores 2 y 3
son el rojo y verde
        qqline(x)
    }
}
```



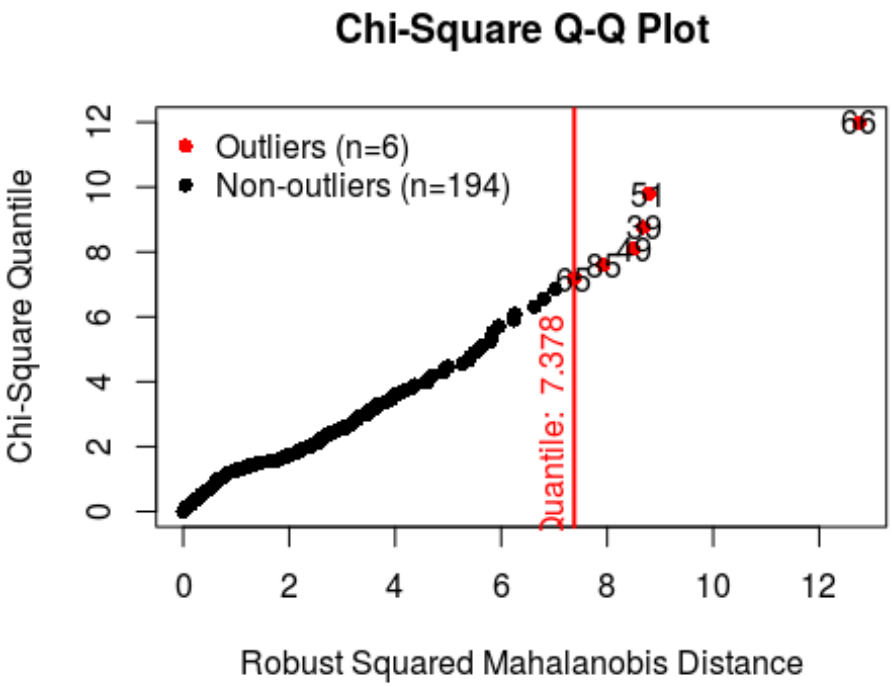

```
# Contraste de normalidad Shapiro-Wilk para cada variable en cada especie
library(reshape2)
datos_tidy <- melt(datos, value.name = "valor")
library(dplyr)
kable(datos_tidy %>% group_by(grupo, variable) %>% summarise(p_value_Shapiro.test =
round(shapiro.test(valor)$p.value, 5)))
```

grupo	variable	p_value_Shapiro.test
A	variable_z	0.00172
A	variable_w	0.09319
B	variable_z	0.62479
B	variable_w	0.81022

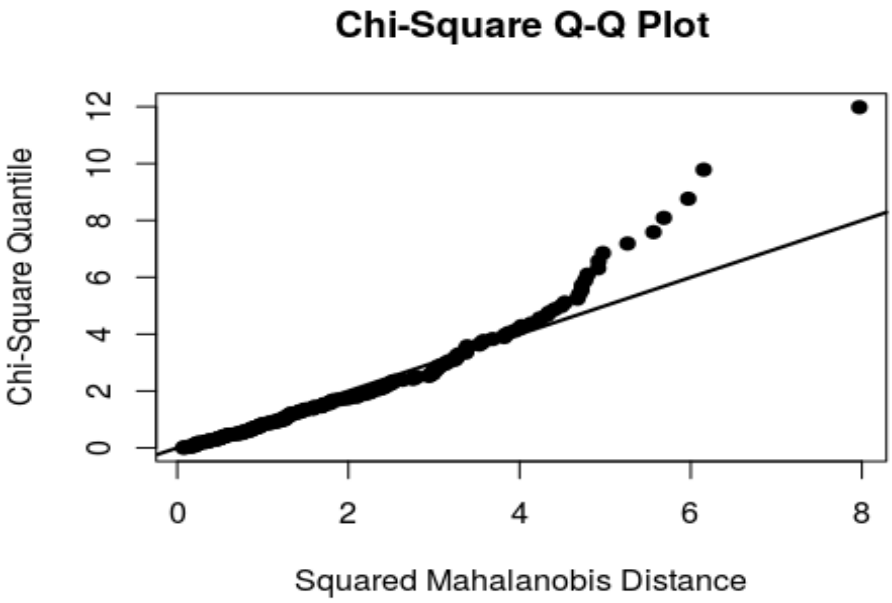
La variable Z no se distribuye de forma normal en el grupo A.

Normalidad multivariante:

```
library(MVN)
outliers <- mvOutlier(datos[,-3], qqplot = TRUE, method = "quan")
```




```
roystonTest(data = datos[,-3], qqplot = TRUE)
```



```
##   Royston's Multivariate Normality Test
## ---------------------------------------------
##   data : datos[, -3]
##
##   H       : 29.11024
##   p-value : 4.77274e-07
##
##   Result  : Data are not multivariate normal.
## ---------------------------------------------
```

```
hzTest(data = datos[,-3], qqplot = FALSE)
```

```
##   Henze-Zirkler's Multivariate Normality Test
## ---------------------------------------------
##   data : datos[, -3]
##
##   HZ      : 6.739874
##   p-value : 5.317968e-14
##
##   Result  : Data are not multivariate normal.
## ---------------------------------------------
```

Ambos test muestran evidencias significativas de falta de normalidad multivariante. El *QDA* tiene cierta robustez frente a la falta de normalidad multivariante, pero es importante tenerlo en cuenta en la conclusión del análisis.

Cálculo de la función discriminante

```
library(MASS)
modelo_qda <- qda(grupo ~ variable_z + variable_w, data = datos)
modelo_qda
```

```
## Call:
## qda(grupo ~ variable_z + variable_w, data = datos)
##
## Prior probabilities of groups:
##   A   B
## 0.5 0.5
##
## Group means:
##   variable_z variable_w
## A  0.5000000   4.615307
## B  0.4864889   1.992911
```

Evaluación de los errores de clasificación

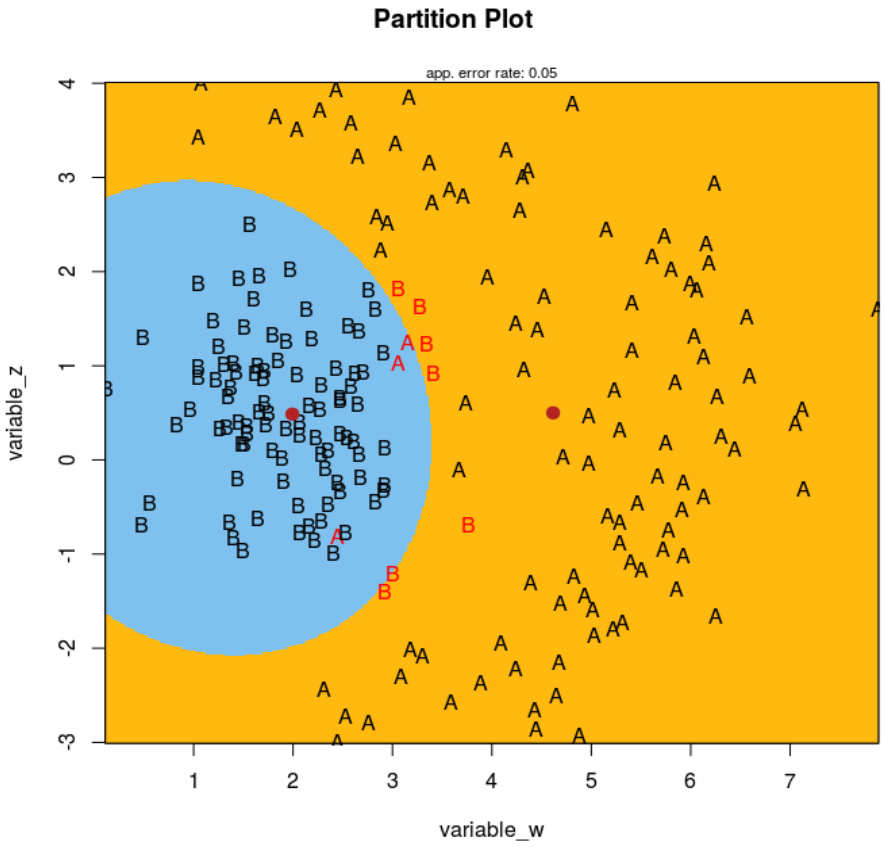
```
predicciones <- predict(object = modelo_qda, newdata = datos)
table(datos$grupo, predicciones$class, dnn = c("Clase real", "Clase predicha"))
```

```
##            Clase predicha
## Clase real  A  B
##          A 97  3
##          B  7 93
```

```
trainig_error <- mean(datos$grupo != predicciones$class) * 100
paste("trainig_error=", trainig_error, "%")
```

```
## [1] "trainig_error= 5 %"
```

```
library(klaR)
partimat(formula = grupo ~ variable_z + variable_w, data = datos, method = "qda",
    prec = 400, image.colors = c("darkgoldenrod1", "skyblue2"), col.mean =
"firebrick")
```



Ejemplo QDA billetes falsos

Se pretende generar un modelo discriminante que permita diferenciar entre billetes verdaderos y falsos. Se han registrado múltiples variables para 100 billetes verdaderos y 100 billetes falsos:

- Status: si es verdadero (*genuine*) o falso (*counterfeit*).
- Length: longitud (mm)
- Left: Anchura del borde izquierdo (mm)
- Right: Anchura del borde derecho (mm)
- Bottom: Anchura del borde inferior (mm)
- Top: Anchura del borde superior (mm)
- Diagonal: longitud diagonal (mm)

```
library(mclust)
library(knitr)
data(banknote)
# se recodifican las clases de la variable Status: verdadero = 0, falso = 1
levels(banknote$Status)
```

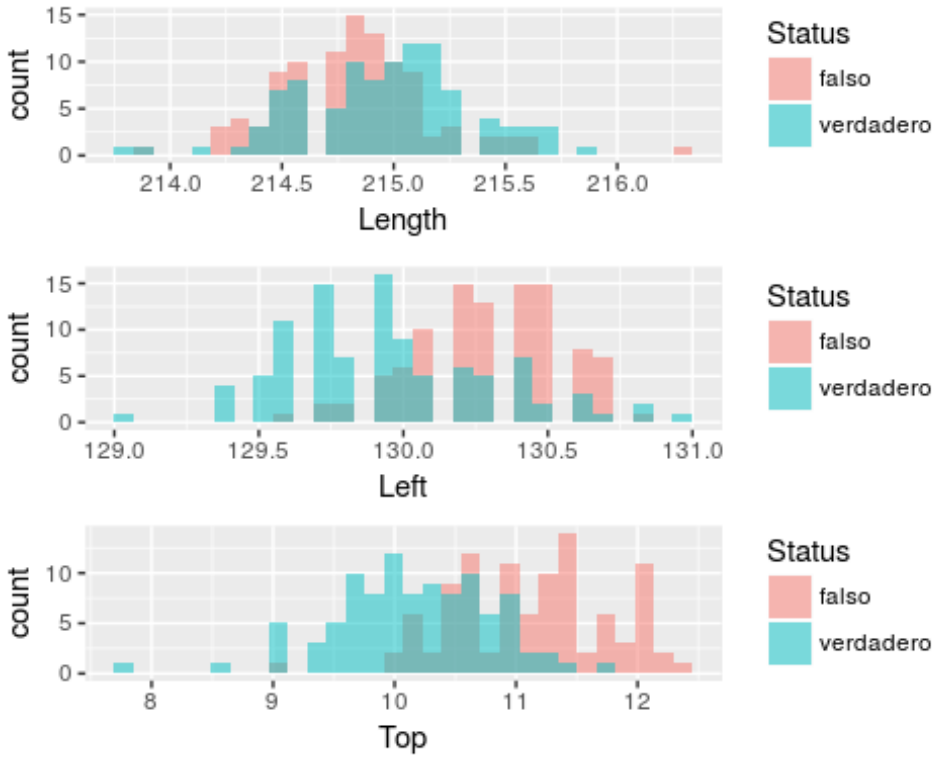
```
## [1] "counterfeit" "genuine"
```

```
levels(banknote$Status) <- c("falso", "verdadero")
kable(head(banknote))
```

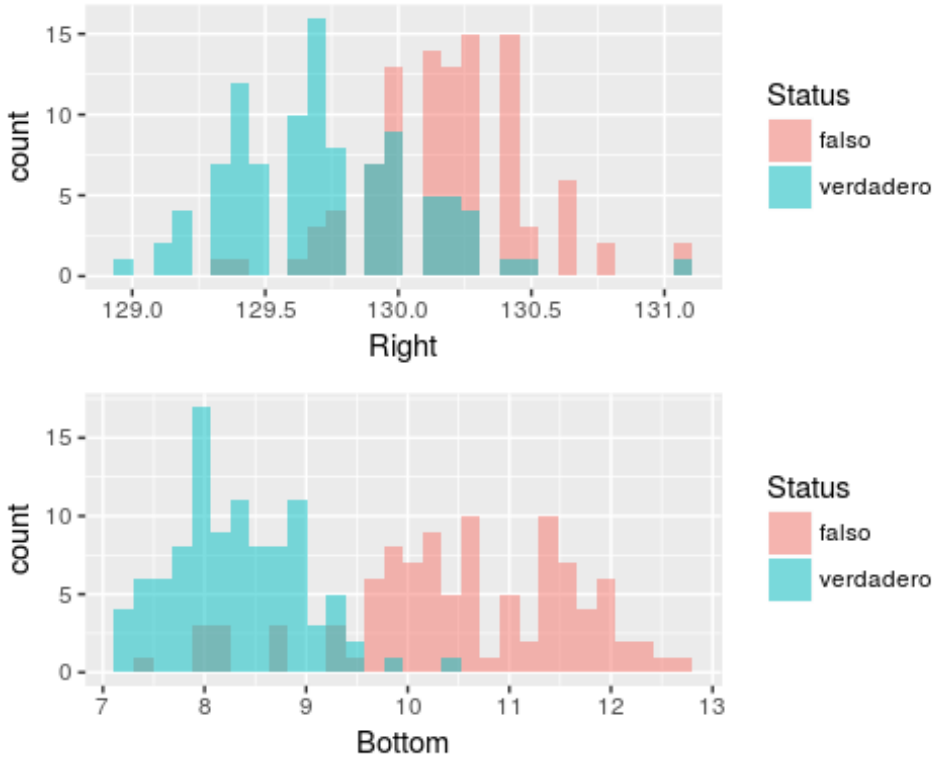
Status	Length	Left	Right	Bottom	Top	Diagonal
verdadero	214.8	131.0	131.1	9.0	9.7	141.0
verdadero	214.6	129.7	129.7	8.1	9.5	141.7
verdadero	214.8	129.7	129.7	8.7	9.6	142.2
verdadero	214.8	129.7	129.6	7.5	10.4	142.0
verdadero	215.0	129.6	129.7	10.4	7.7	141.8
verdadero	215.7	130.8	130.5	9.0	10.1	141.4

```
library(ggplot2)
library(gridExtra)
```

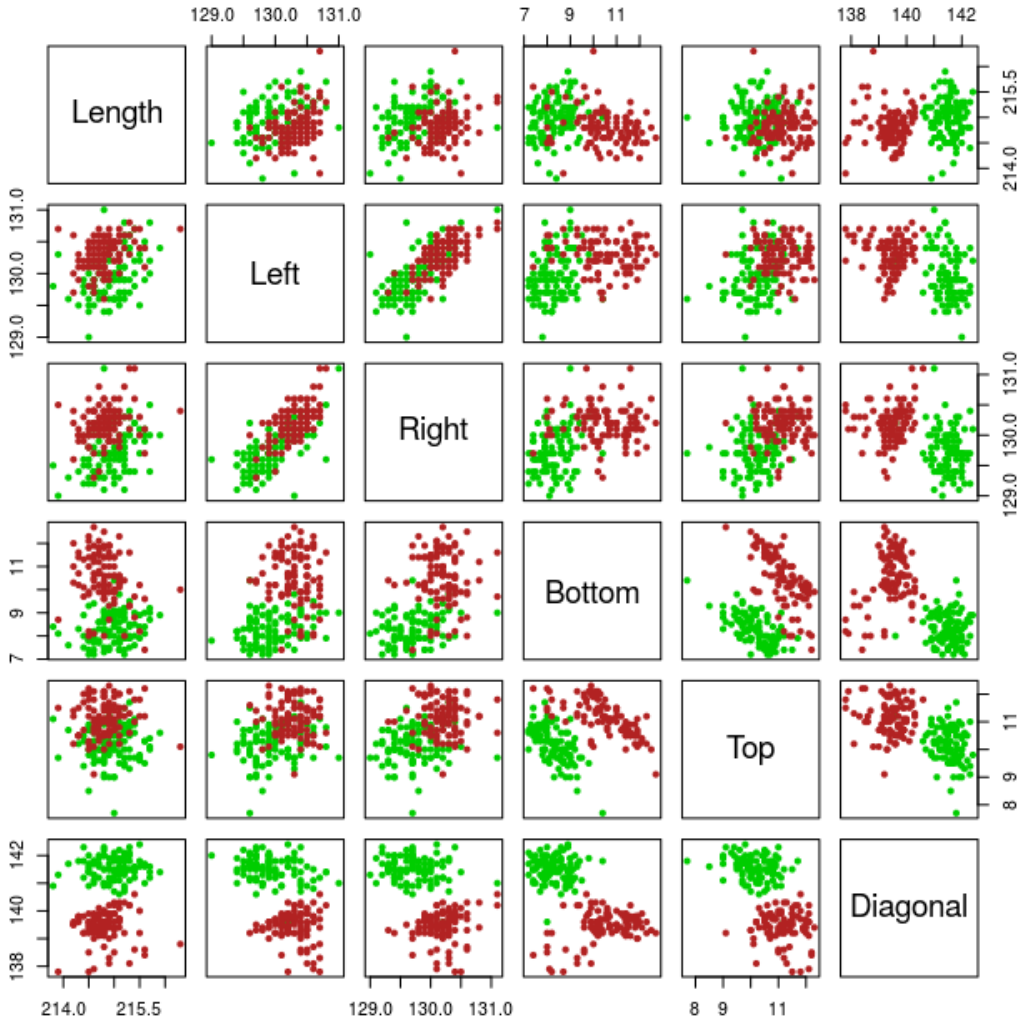
```
p1 <- ggplot(data = banknote, aes(x = Length, fill = Status)) +
geom_histogram(position = "identity", alpha = 0.5)


p2 <- ggplot(data = banknote, aes(x = Left, fill = Status)) +
geom_histogram(position = "identity", alpha = 0.5)


p3 <- ggplot(data = banknote, aes(x = Right, fill = Status)) +
geom_histogram(position = "identity",  alpha = 0.5)


p4 <- ggplot(data = banknote, aes(x = Bottom, fill = Status)) +
geom_histogram(position = "identity", alpha = 0.5)


p5 <- ggplot(data = banknote, aes(x = Top, fill = Status)) +
geom_histogram(position = "identity", alpha = 0.5)


grid.arrange(p1, p2, p5)
```




```
grid.arrange(p3, p4)
```


count
15
10
5
0
129.0
129.5
130.0
130.5
131.0
Right
Status
falso
verdadero
count
15
10
5
0
7
8
9
10
11
12
13
Bottom
Status
falso
verdadero

```
pairs(x = banknote[, -1], col = c("firebrick", "green3")[banknote$Status], pch =
20)
```

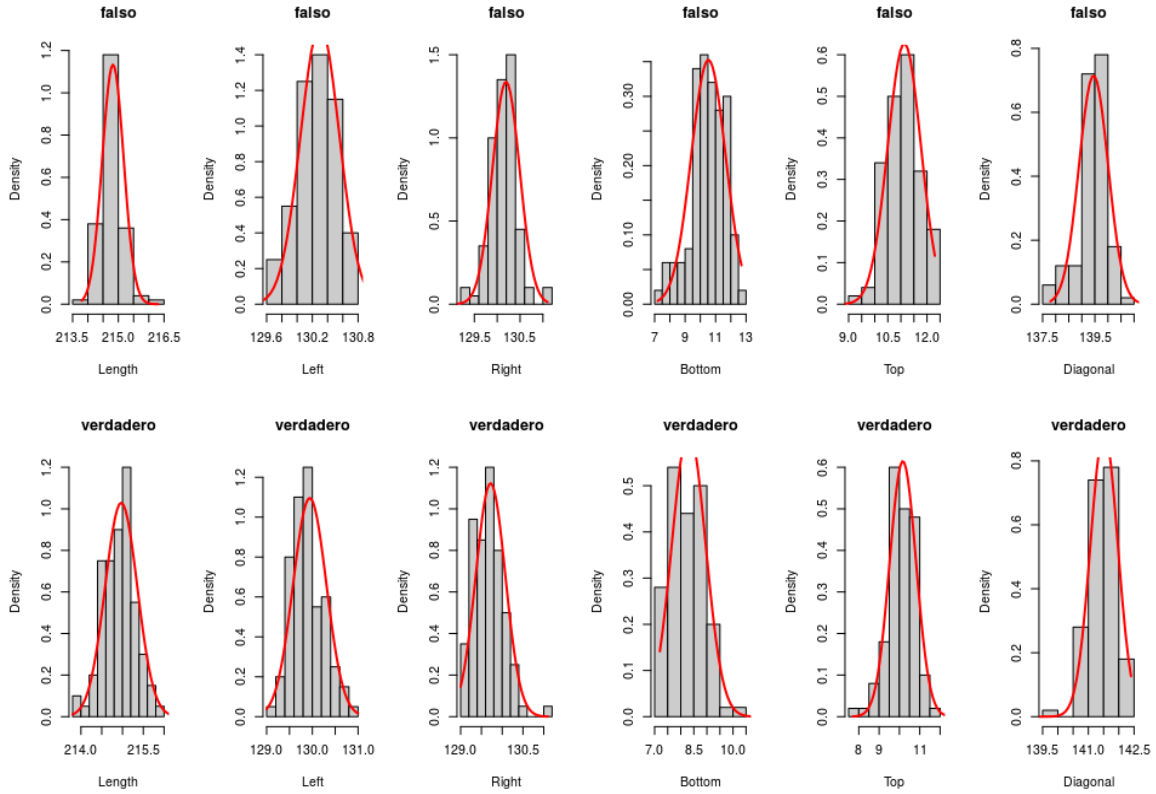


La proporción de billetes falsos en circulación es mucho menor que la de billetes verdaderos, por lo tanto, a pesar de que en la muestra se dispone de la misma cantidad de billetes de cada clase, no es conveniente considerar que las probabilidades previas son iguales. En este tipo de escenario se suele recurrir a estudios previos o muestras piloto que permitan estimar las proporciones poblacionales de cada clase. En nuestro caso se va a suponer que solo el 1% de los billetes en circulación son falsos.

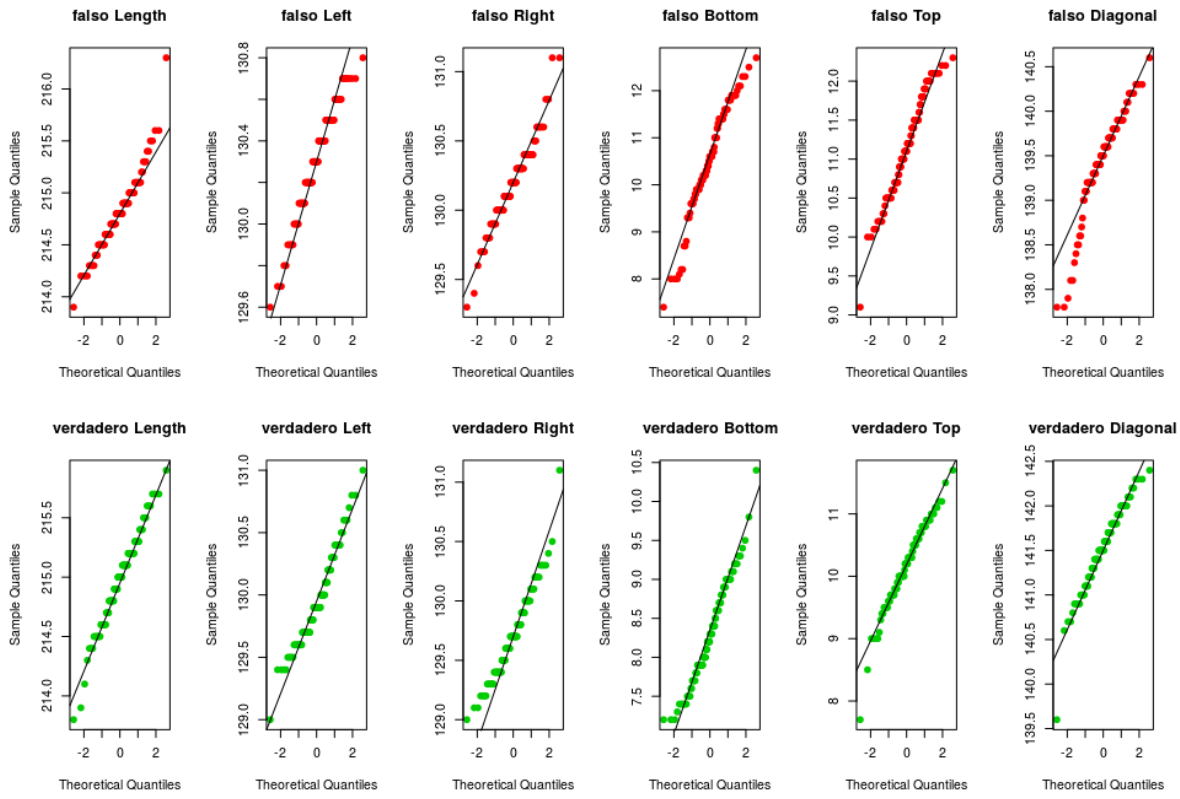
$$\hat{\pi}_{verdadero} = 0.99 \quad \hat{\pi}_{falso} = 0.01$$

Distribución de los predictores de forma individual:

```
# representación mediante Histograma de cada variable para cada especie
par(mfcol = c(2, 6))
for (k in 2:7) {
    j0 <- names(banknote)[k]
    x0 <- seq(min(banknote[, k]), max(banknote[, k]), le = 50)
    for (i in 1:2) {
        i0 <- levels(banknote$Status)[i]
        x <- banknote[banknote$Status == i0, j0]
        hist(x, proba = T, col = grey(0.8), main = paste(i0), xlab = j0)
        lines(x0, dnorm(x0, mean(x), sd(x)), col = "red", lwd = 2)
    }
}
```



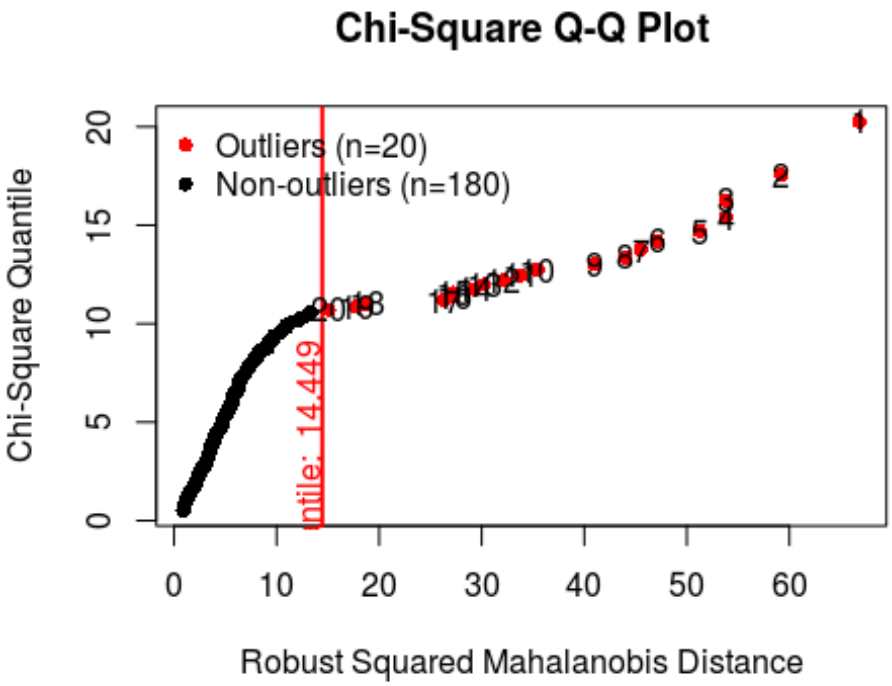
```
# representación de cuantiles normales de cada variable para cada especie
for (k in 2:7) {
    j0 <- names(banknote)[k]
    x0 <- seq(min(banknote[, k]), max(banknote[, k]), le = 50)
    for (i in 1:2) {
        i0 <- levels(banknote$Status)[i]
        x <- banknote[banknote$Status == i0, j0]
        qqnorm(x, main = paste(i0, j0), pch = 19, col = i + 1)  # los colores 2 y 3
son el rojo y verde
        qqline(x)
    }
}
```



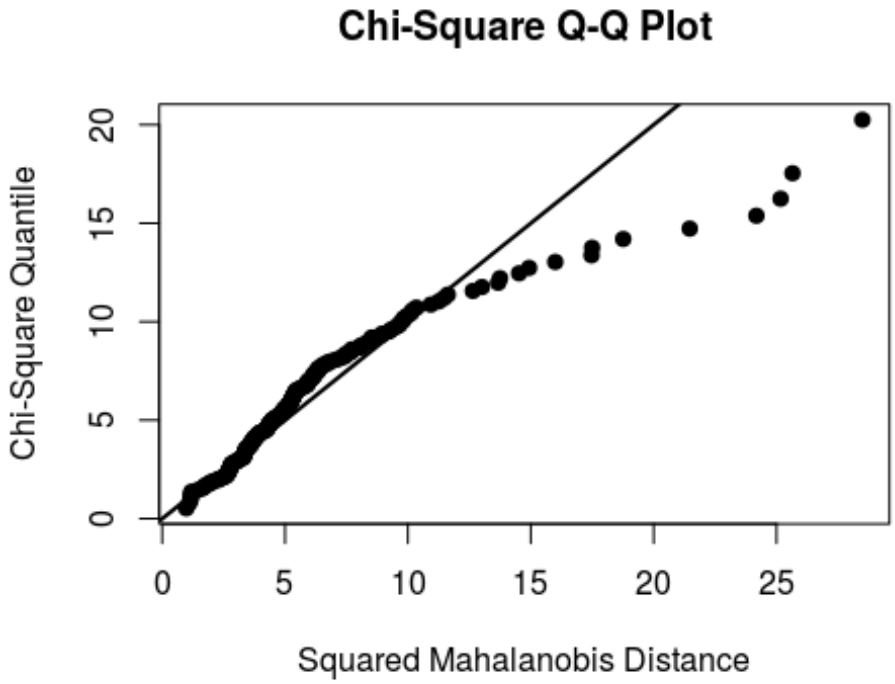

```
# Contraste de normalidad Shapiro-Wilk para cada variable en cada especie
library(reshape2)
library(knitr)
library(dplyr)
datos_tidy <- melt(banknote, value.name = "valor")
kable(datos_tidy %>% group_by(Status, variable) %>% summarise(p_value_Shapiro.test
= round(shapiro.test(valor)$p.value, 5)))
```

Status	variable	p_value_Shapiro.test
falso	Length	0.00290
falso	Left	0.02372
falso	Right	0.01683
falso	Bottom	0.01115
falso	Top	0.04909
falso	Diagonal	0.00003
verdadero	Length	0.25674
verdadero	Left	0.00825
verdadero	Right	0.01174
verdadero	Bottom	0.04035
verdadero	Top	0.04984
verdadero	Diagonal	0.00330

```
library(MVN)
outliers <- mvOutlier(banknote[,-1], qqplot = TRUE, method = "quan")
```



```
roystonTest(data = banknote[,-1], qqplot = TRUE)
```




```
##   Royston's Multivariate Normality Test
## ---------------------------------------------
##   data : banknote[, -1]
##
##   H       : 67.03927
##   p-value : 5.820549e-13
##
##   Result  : Data are not multivariate normal.
## ---------------------------------------------
```

```
hzTest(data = banknote[,-1], qqplot = FALSE)
```

```
##   Henze-Zirkler's Multivariate Normality Test
## ---------------------------------------------
##   data : banknote[, -1]
##
##   HZ      : 1.780591
##   p-value : 0
##
##   Result  : Data are not multivariate normal.
## ---------------------------------------------
```

Los datos no siguen una distribución normal multivariante, lo que tiene implicaciones directas en la precisión del *QDA*.

```
library(biotools)
boxM(data = banknote[, -1], grouping = banknote[, 1])
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  banknote[, -1]
## Chi-Sq (approx.) = 121.9, df = 21, p-value = 3.198e-16
```

El test Box's M muestra fuertes evidencias de que la matriz de covarianza no es constante en todos los grupos, esta condición hacen que el QDA sea más adecuado.

```
library(MASS)
modelo_qda <- qda(formula = Status ~ ., data = banknote, prior = c(0.01, 0.99))
modelo_qda
```

```
## Call:
## qda(Status ~ ., data = banknote, prior = c(0.01, 0.99))
##
## Prior probabilities of groups:
##     falso verdadero
##      0.01      0.99
##
## Group means:
##            Length    Left   Right Bottom    Top Diagonal
## falso     214.823 130.300 130.193 10.530 11.133  139.450
## verdadero 214.969 129.943 129.720  8.305 10.168  141.517
```

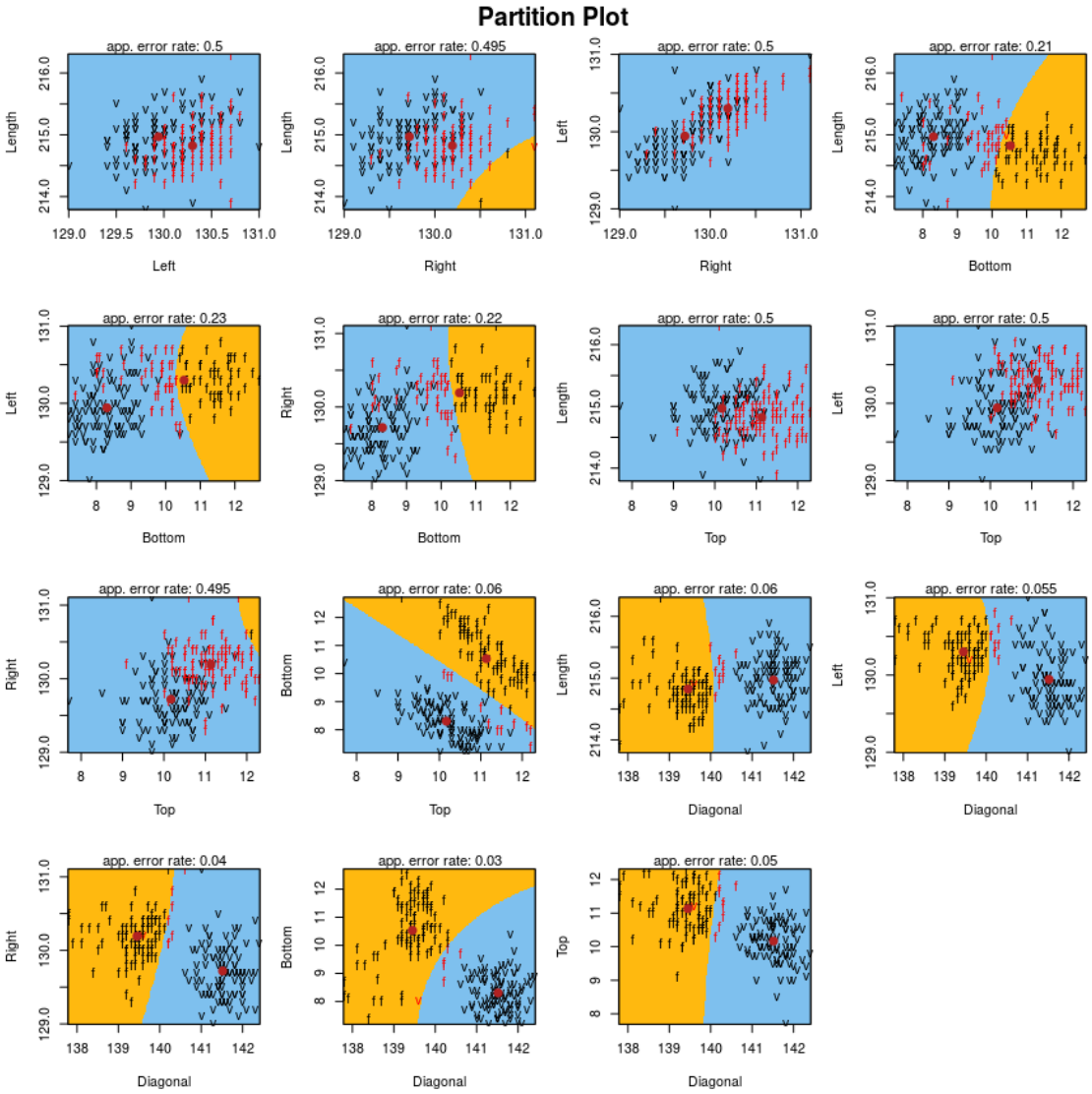
```
predicciones <- predict(object = modelo_qda, newdata = banknote)
table(banknote$Status, predicciones$class, dnn = c("Clase real", "Clase predicha"))
```

```
##            Clase predicha
## Clase real  falso verdadero
##   falso        99         1
##   verdadero     0       100
```

```
trainig_error <- mean(banknote$Status != predicciones$class) * 100
paste("trainig_error=", trainig_error, "%")
```

```
## [1] "trainig_error= 0.5 %"
```

```
library(klaR)
partimat(formula = Status ~ ., data = banknote, prior = c(0.01, 0.99), method =
"qda", prec = 200, image.colors = c("darkgoldenrod1", "skyblue2"), col.mean =
"firebrick", nplots.vert = 4)
```



Comparación entre QDA y LDA

Que clasificador es más adecuado depende de las implicaciones que tiene, en el balance *bias-varianza*, el asumir que todos los grupos comparten una matriz de covarianza común. *LDA* produce límites de decisión lineales lo que se traduce en menor flexibilidad y por lo tanto menor problema de varianza. Sin embargo, si la separación de los grupos no es lineal, tendrá un bias grande. El método *QDA* produce límites cuadráticos y por lo tanto curvos, lo que aporta mayor flexibilidad permitiendo ajustarse mejor a los datos, menor bias pero mayor riesgo de varianza.

En términos generales, *LDA* tiende a conseguir mejores clasificaciones que *QDA* cuando hay pocas observaciones con las que entrenar al modelo, escenario en el que evitar la varianza es crucial. Por contra, si se dispone de una gran cantidad de observaciones de entrenamiento o si no es asumible que existe una matriz de covarianza común entre clases, *QDA* es más adecuado.

Si se dispone de p predictores, calcular una matriz de covarianza común requiere estimar $p(p+1)/2$ parámetros, mientras que calcular una matriz diferente para cada grupo requiere de $Kp(p+1)/2$. Para valores de p muy altos, la elección del método puede estar limitada por la capacidad computacional.

Bibliografía

Introduction to Statistical Learning

Using R With Multivariate Statistics Escrito por Randall E. Schumacker

http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema1dm.pdf

http://www2.stat.unibo.it/montanari/Didattica/Multivariate/Discriminant_analysis.pdf

Using discriminant analysis for multi-class classification: an experimental investigation (Tao Li, Shenghuo Zhu, Mitsunori Ogihara)

STAT 505 - Applied Multivariate Statistical Analysis, PennState