máximaformación

TEMA 8

DISCRIMINACIÓN (LDA)

MÁSTER DE ESTADÍSTICA APLICADA CON R SOFTWARE. TÉCNICAS CLÁSICAS, ROBUSTAS, AVANZADAS Y MULTIVARIANTES

TEMA 8: Discriminación (LDA)

8.1. Análisis discriminante lineal (LDA)

El DA nos permite considerar información extra sobre el agrupamiento de los datos.

Para este tipo de análisis contamos con una agrupación previa y nos preguntamos: ¿las variables difieren entre los grupos de observaciones? y si es así ¿qué variables?; utilizando estas variables, ¿podemos discriminar entre grupos de observaciones definidos a priori? y ¿cuáles son las más discriminantes?

8.1.1 Objetivos

- Determinar si existen diferencias significativas entre los perfiles de un conjunto de variables de dos o más grupos definidos a priori.
- Determinar cuál de las variables independientes cuantifica mejor las diferencias entre un grupo u otro.
- Establecer un procedimiento para clasificar a un individuo en base a los valores de un conjunto de variables independientes.

8.1.2 Formato de datos

En la figura 10 podemos observar el tipo de matriz con la que trabajaremos.

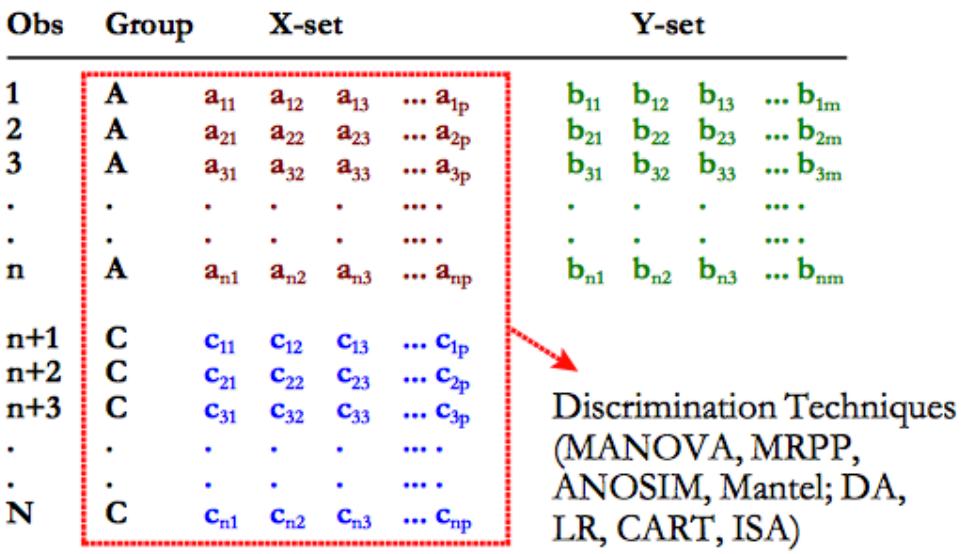


Figura 10. Análisis discriminante lineal (LDA).

Ejemplos:

- Si queremos analizar cómo difieren las relaciones entre 4 especies a lo largo de 3 transectos o entre estaciones, o entre ambos. Es decir, queremos investigar la relación entre especies, discriminando entre estaciones, entre transectos o entre ambos, para determinar cuáles son las más importantes a la hora de entender la relación entre especies.
- Queremos determinar si los valores de ácidos grasos nos permiten discriminar entre especies hembras y machos, tipo de dientes y área donde varan los delfines.
- Cuando queremos investigar si un conjunto de variables nos permite discriminar las especies de una muestra, por ejemplo, en datos de aves donde contamos con información sobre árboles visitados, velocidad, etc… si queremos averiguar si observadores distintos producen mediciones similares o si, por lo contrario, existe un efecto del observador.

Comparación con otras técnicas.

La técnica más común para establecer relaciones, predecir y explicar variables son las técnicas de regresión. El problema está cuando la variable a explicar no es una variable medible (o métrica); en este caso existen dos tipos de análisis con los que resolver el problema, el análisis discriminante y la regresión logística. En ambos análisis tendremos una variable dependiente categórica y varias variables independientes numéricas. En muchas ocasiones la variable categórica consta de dos grupos o clasificaciones (e.g. hembra-macho). En otras situaciones la variable categórica tendrá tres o más subgrupos (e.g. bajo, medio y alto nivel de cierta dosis). La regresión logística o logia, en su forma básica está restringida a dos grupos frente al análisis discriminante que vale para más de dos.

Supuestos.

- A la hora de estudiar las variables se tiene que cumplir que la *variable dependiente* (grupos) sea categórica en la que el número de grupos puede ser de dos o más, pero han de ser mutuamente excluyentes y exhaustivos. Aunque la variable dependiente puede ser originariamente numérica y que el investigador la cuantifique en términos de categorías.

- Las *variables independientes* numéricas se seleccionan identificando las variables en una investigación previa o mediante información a priori, de tal manera que se sepa que esas variables son importantes para predecir en qué grupo estará la variable dependiente. Se puede utilizar el análisis cluster para formar los grupos, pero se recomienda seguir los siguientes pasos: dividir los datos en 2 grupos, aplicar el análisis cluster en uno de ellos y utilizar los resultados en el DA para el segundo grupo de datos.

- Con respecto al *tamaño de las muestras*, se suele recomendar que los tamaños de cada grupo no sean muy diferentes, ya que con esto la probabilidad de pertenecer a un grupo o a otro puede variar

considerablemente. Se necesita que al menos tengamos 4 o 5 veces más observaciones por grupo que el número de variables que utilicemos. Además, el número de observaciones en el grupo más pequeño debe ser mayor que el número de variables.

- También existen dos hipótesis previas que deben ser contrastadas, estas son: la *normalidad multivariante* y la de la estructura de varianzas-covarianzas desconocidas pero iguales (*homogeneidad de varianzas* entre grupos). Los datos que no cumplen el supuesto de normalidad pueden causar problemas en la estimación y en ese caso se sugiere utilizar la regresión logística. Si existen grandes desviaciones en las varianzas, se puede solucionar con la ampliación de la muestra o con técnicas de clasificación cuadráticas. La homogeneidad de varianzas significa que la relación entre variables debe ser similar para los distintos grupos. Por tanto, una variable no puede tener el mismo valor para todas las observaciones dentro de un grupo.

- Los datos además no deben presentar *multicolinealidad*, es decir, que dos o más variables independientes estén muy relacionadas. Si las variables tienen un valor de correlación de 0.9 o mayor se debe eliminar una de ellas.

- También se supone *linealidad* entre las variables ya que se utiliza la matriz de covarianza.

Si no se cumplen los supuestos de normalidad y homogeneidad, podemos utilizar una transformación logarítmica o de la raíz cuadrada.

8.1.3 Metodología

El modelo.

El análisis discriminante implica un valor teórico como combinación lineal de dos o más variables independientes que discrimine entre los grupos definidos a priori. La discriminación se lleva a cabo estableciendo las ponderaciones del valor teórico de cada variable, de tal forma que maximicen la varianza entre-grupos frente a la intra-grupos. La combinación lineal o función discriminante, toma la siguiente forma:
$D_i = a + W_1X_{1,i} + W_2X_{2,i} + \ldots + W_nX_{n,i}$ donde: D_i es la puntuación discriminante (grupo de pertenencia) del individuo i-ésimo; a es una constante; W_j es la ponderación de la variable j-ésima. El resultado de esta función será para un conjunto de variables $X_1, \ldots, X_n$ un valor de D que discrimine al individuo en un grupo u otro. Destacamos que el análisis discriminante proporcionará una función discriminate menos que los subgrupos que tengamos, es decir, si la variable categórica tiene dos subgrupos, obtendremos una función discriminante, si tiene tres subgrupos obtendremos dos y así sucesivamente. Este concepto se ve muy claro desde un punto de vista gráfico, ya que si existen tres subgrupos, el procedimiento lo que hará será establecer dos funciones que separen a un grupo de otro.
Es interesante estudiar, con una metodología similar a la de regresión, las ponderaciones W_i estandarizados, de tal manera que estas será la contribución de cada variable a la función, por lo tanto las variables con valores grandes de la ponderación contribuirán más que las de valores pequeños. A la hora de realizar el procedimiento existen dos tipos de análisis:
La introducción simultánea de todas las variables por lo que las funciones discriminantes se calculan basándose en el conjunto de datos.
La introducción por etapas. Se empieza por la más discriminante y una a una se van induciendo en el modelo de tal manera que algunas variables podrán ser eliminadas cuando aporten poco al modelo discriminate. Este método es bueno cuando se tengan muchas variables, de tal manera que obtendremos modelos más sencillos tan buenos como si se hubieran introducido todas las variables.

Validación de los resultados.

Para la validación de los resultados se usa un procedimiento equivalente al de la R^2 en regresión que es el ratio de aciertos. Este ratio mide el porcentaje de individuos correctamente clasificados, evidentemente, cuanto más alto es este ratio mejor serán las funciones discriminantes. Este ratio aparecerá en las matrices de clasificación.

Estudio de los errores.

Una vez realizado todo el procedimiento podremos ver en qué individuos se ha producido error. Estos errores serán de dos tipos:
1. Que hayan sido mal clasificados los individuos.
2. Que no sean representativos del grupo.

El estudio de estos errores puede hacer que el modelo mejore considerablemente, de ahí su importancia. Para analizarlos se suele utilizar la representación gráfica de las observaciones, representando las puntuaciones Z discriminantes y buscando los casos mal representados para su estudio.

Criterios de clasificación alternativos.

Análisis discriminante cuadrático.

Se trata de un procedimiento más robusto que el lineal, y es útil cuando las matrices de covarianza no son iguales. Se basa en la distancia de Mahalanobis al cuadrado respecto al centro del grupo.

Análisis discriminante kernel.

Es un método no-paramétrico alternativo, basado en el kernel de densidad esimado.

Análisis discriminante del k-vecino más cercano.

Corresponde a otro tipo de método no-paramétrico, basado en los k-vecinos más cercanos. Este método calcula la distancia euclídea entre cada punto y sus k-vecinos más cercanos y clasifica cada punto en el grupo con la mínima distancia media.

8.1.4 Aplicación

Datos doubs.

Con este ejemplo realizaremos el análisis lineal discriminante de los datos de abundancia de peces a lo largo del río Doubs (francés-suizo) y las características ambientales del sitio, obtenidos por Verneaux (1973).
Tenemos 30 filas (sitios), 27 especies de peces (columnas del archivo spe), 11 variables ambientales (columnas del archivo env) y las coordenadas espaciales de los sitios (columnas del archivo spa). Para obtener más detalles de los datos puedes preguntar en R sobre los datos doubs del paquete ade4.

Objetivos.

Queremos:
1. Explicar las diferencias entre grupos: encontrar la separación “óptima” entre grupos (clusters) basados en ciertas transformaciones lineales de las variables explicativas (o discriminantes) y determinar qué variables son las que están más relacionadas con la separación de los grupos.
2. Predecir a qué grupo pertenece una nueva observación (caso o sitio): basado en los valores de las variables explicativas queremos determinar a qué grupo pertenece un nuevo sitio.

Preparación de los datos.

Evaluar la idoneidad de los datos.

- Los datos deben contener un conjunto de variables explicativas y una única variable de agrupación (categórica).
- Las variables exlicativas (o discriminantes) deben ser cuantitativas y preferiblemente del mismo tipo (continuas o discretas).
- Lo ideal sería que los datos presenten múltiples muestras en cada grupo, grupos de tamaños similares, al menos 2 o más muestras (filas) que variables (columnas) y al menos 2 muestras por grupo. Como regla tácita podemos proponer que hayan al menos 3 veces tantas observaciones por grupo como número de variables.
- No se permiten valores perdidos.

```
#cargamos los paquetes necesarios
library(ade4)
library(vegan)
library(ellipse)
library(mvnormtest)
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following object is masked from 'package:plotrix':
##
##     rescale
##
## The following object is masked from 'package:FSA':
##
##     headtail
##
## The following object is masked from 'package:pgirmess':
##
##     shannon
```

```
library(MASS)
library(klaR)
```

```
##
## Attaching package: 'klaR'
##
## The following object is masked from 'package:vegan':
##
##     rda
```

```
#cargamos los datos
data(doubs)
spe <- doubs$fish[-8,] #quitamos la línea 8 porque está formada solo por ceros
env <- doubs$env[-8,]
spa <- doubs$xy[-8,]
```

Agrupación previa.

En el LDA debemos tener una agrupación a priori de los sitios, que puede obtenerse mediante un análisis cluster previo o en base a hipótesis ecológicas previas. El LDA busca determinar qué tanto podemos explicar esta agrupación a partir de un conjunto de variables cuantitativas. Por lo tanto, la tipología de los sitios debe obtenerse de manera independiente a las variables explicativas usadas en el LDA.

Realizamos un análisis cluster sobre los datos de especies de peces, que servirá para obtener los grupos que utilizaremos en el LDA.

```
# realizamos la transformación de Hellinger de los datos de especies
spe.hel <- decostand(spe, "hellinger")
```

```
# siguiendo el ejercicio anterior, realizamos el análisis cluster por el método de Ward
# cortamos en k=4 grupos
gr <- cutree(hclust(vegdist(spe.hel, "euc"), "ward.D"), 4)
```

Evaluar los supuestos del modelo.

1. **Homogeneidad de varianzas multivariante**. Las matrices de covarianza de las variables discriminantes deben ser iguales. El LDA asume que los grupos tienen similares dispersiones (i.e. la estructura de varianza-covarianza intra-grupo es la misma para todos los grupos). Se realizan pruebas de homogeneidad multivariadas y en caso de rechazo se indaga en la homogeneidad univariada de cada variable discriminante. Aquella variable que no muestre patrones de homogeneidad, será la candidata a que le realicemos transformaciones. Si existen grandes desviaciones en las varianzas se puede solucionar con la ampliación de la muestra o con técnicas de clasificación cuadráticas. La homogeneidad de varianzas significa que la relación entre variables debe ser similar para los distintos grupos. Por tanto, una variable no puede tener el mismo valor para todas las observaciones dentro de un grupo.
2. **Normalidad multivariante**. Los datos que no cumplen el supuesto de normalidad pueden causar problemas en la estimación y en ese caso se sugiere utilizar la regresión logística.
3. **Multicolinealidad**. Los datos además no deben presentar multicolinealidad, es decir, que dos o más variables independientes estén muy relacionadas. Si las variables tienen un valor de correlación de 0.9 o mayor, se debe eliminar una de ellas.
4. **Linealidad**. También se supone linealidad entre las variables ya que se utiliza la matriz de covarianza.

Recordar que aún cuando tenemos variables medidas en diferentes escalas, no necesitamos estandarizarlas ya que el LDA utiliza la estandarización intrínseca de las columnas.

Homogeneidad de varianzas multivariante.

Primero debemos asegurarnos de que las matrices de covarianza intra-grupos de las variables explicativas sean homogénas.

```
# seleccionamos las 3 variables ambientales a analizar (alt, oxy y bdo)
env.pars2 <- as.matrix(env[, c("alt", "oxy", "bdo")])

# verificamos la homogeneidad multivariada de las matrices de covarianza intra-grupo
env.pars2.d1 <- dist(env.pars2)
(env.MHV <- betadisper(env.pars2.d1, gr))
```

```
##
##  Homogeneity of multivariate dispersions
##
## Call: betadisper(d = env.pars2.d1, group = gr)
##
## No. of Positive Eigenvalues: 3
## No. of Negative Eigenvalues: 0
##
## Average distance to median:
##      1      2      3      4
## 198.76 217.31  40.49  21.87
##
## Eigenvalues for PCoA axes:
##       PCoA1       PCoA2       PCoA3        <NA>        <NA>        <NA>
## 2044986.287   44022.742    3148.971          NA          NA          NA
##        <NA>        <NA>
##          NA          NA
```

```
anova(env.MHV)
```

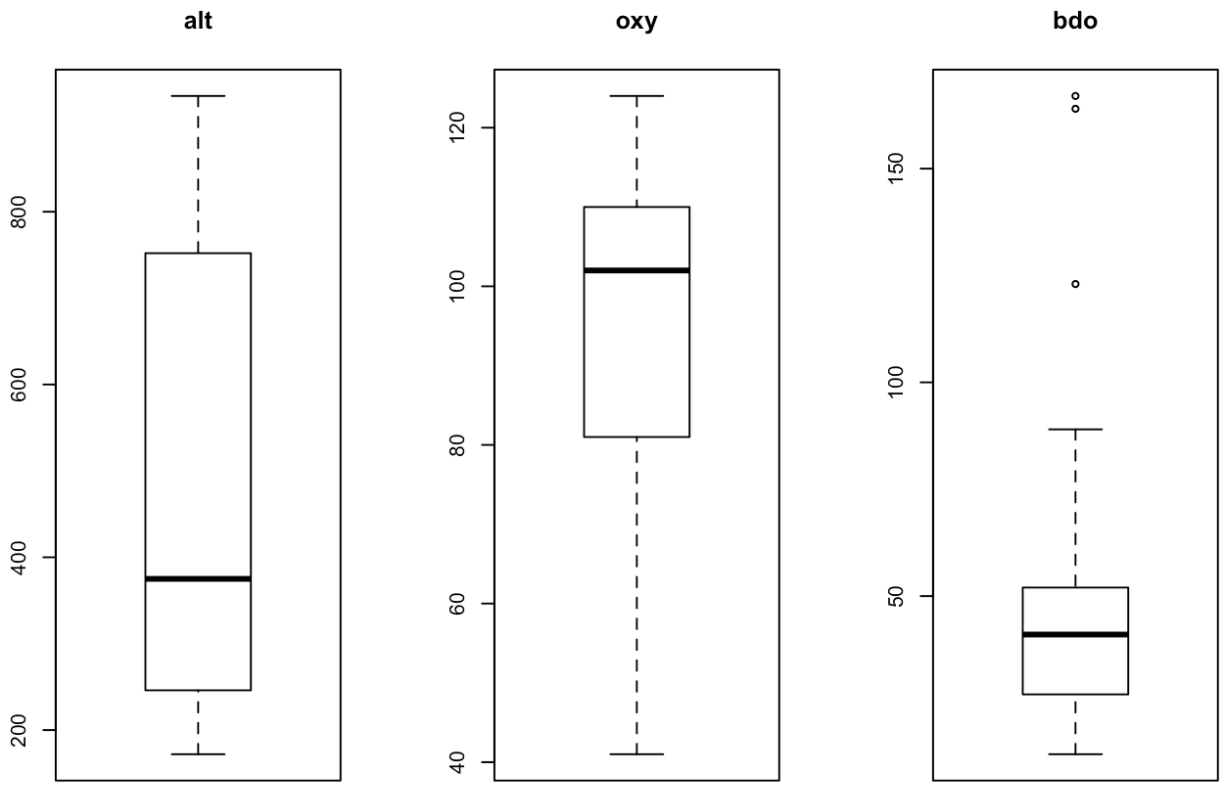
```
## Analysis of Variance Table
##
```

```
## Response: Distances
##           Df Sum Sq Mean Sq F value  Pr(>F)
## Groups     3 205755   68585  4.5758 0.01095 *
## Residuals 25 374713   14989
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
permutest(env.MHV)  # test de permutación
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##           Df Sum Sq Mean Sq      F N.Perm Pr(>F)
## Groups     3 205755   68585 4.5758    999  0.015 *
## Residuals 25 374713   14989
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#como las matrices de covarianza intra-grupos no son homogéneas (p<0.05), vamos a tratar con la transformación log de las variables para homogeneizar los datos.

# observamos la homogeneidad univariada para determinar qué variables se pueden someter a transformación para obtener datos homogéneos.
#generalmente los test son muy conservadores, por lo que en este paso utilizaremos los gráficos de cajas para el análisis.
par(mfrow=c(1,ncol(env.pars2)))
  for(i in 1:ncol(env.pars2)){
    boxplot(env.pars2[,i],main=colnames(env.pars2)[i])
  }
```



```
# Para las variables alt y bdo realizamos la transformación "Log" ya que tienen sesgos a la derecha, etc...

env.pars3 <- cbind(log(env$alt), env$oxy, log(env$bdo))
colnames(env.pars3) <- c("alt.ln", "oxy", "bdo.ln")
row.names(env.pars3) <- row.names(env)

#para probar la homogeneidad multivariada de los grupos debemos ingresar los datos con una matriz de distancias y
calcular la distancia media de los miembros del grupo respecto al centroide (betadisper), luego, probar si las
dispersiones de uno o más grupo son distintas (permutest; usa un test de permutación)
env.pars3.d1 <- dist(env.pars3)
(env.MHV2 <- betadisper(env.pars3.d1, gr))
```

```
##
##  Homogeneity of multivariate dispersions
##
## Call: betadisper(d = env.pars3.d1, group = gr)
##
## No. of Positive Eigenvalues: 3
## No. of Negative Eigenvalues: 0
##
## Average distance to median:
##     1     2     3     4
## 8.016 8.984 9.630 7.336
##
## Eigenvalues for PCoA axes:
##      PCoA1      PCoA2      PCoA3       <NA>       <NA>       <NA>
## 13647.8010     6.7785     2.6042         NA         NA         NA
##       <NA>       <NA>
##         NA         NA
```

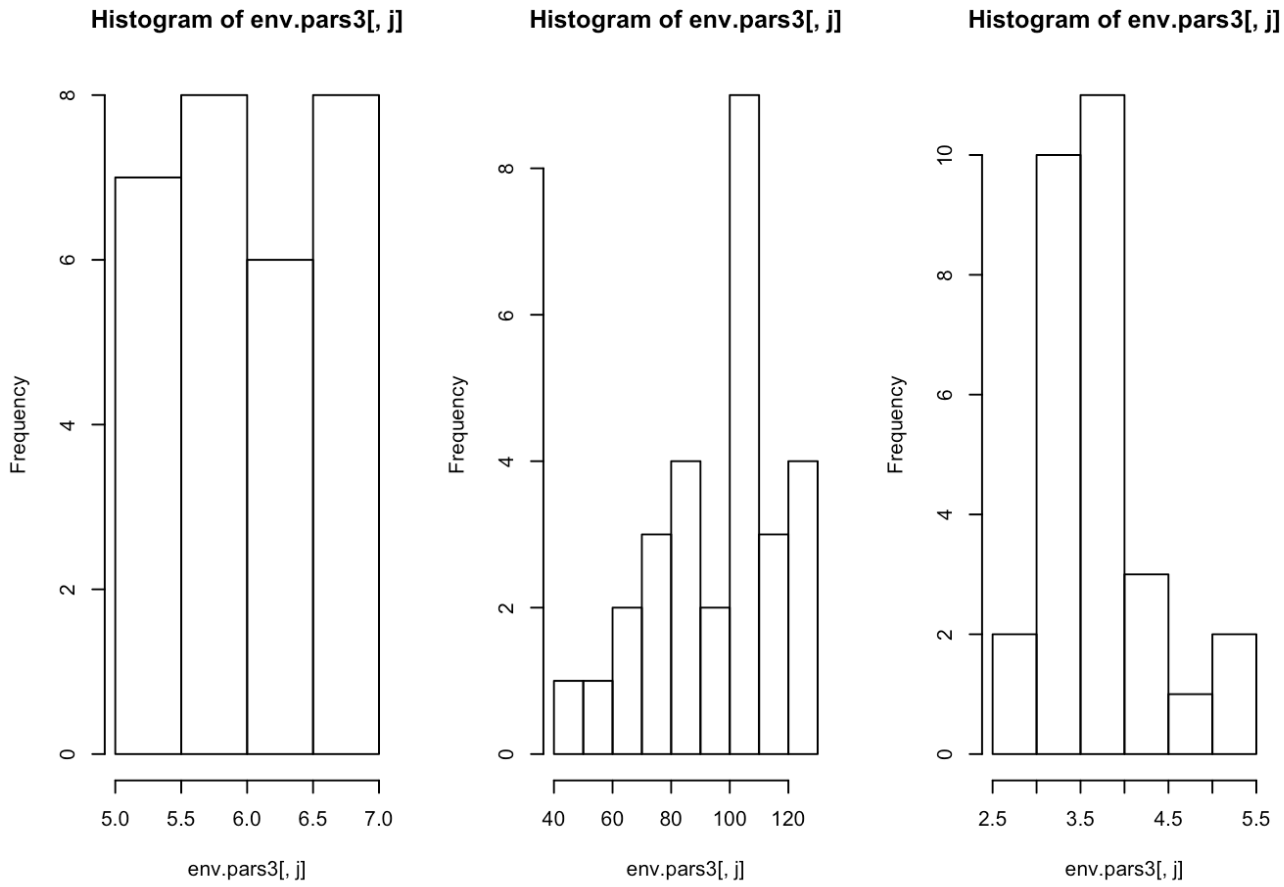
```
permutest(env.MHV2)
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##           Df  Sum Sq Mean Sq      F N.Perm Pr(>F)
## Groups     3   17.44   5.814 0.0882    999  0.975
## Residuals 25 1647.78  65.911
```

```
#ahora las matrices de covarianza intra-grupo son homogéneas
```

Normalidad multivariante.

```
#para verlo gráficamente
par(mfrow=c(1,ncol(env.pars3)))
for(j in 1:ncol(env.pars3))
{
    hist(env.pars3[,j])
}
```




```
#con el paquete mvnormtest
mshapiro.test(t(env.pars3))
```

```
## Shapiro-Wilk normality test with Z
## W = 0.9406, p-value = 0.1041
```

```
# no rechazamos la normalidad multivariante
# también se podría usar el paquete MVN
```

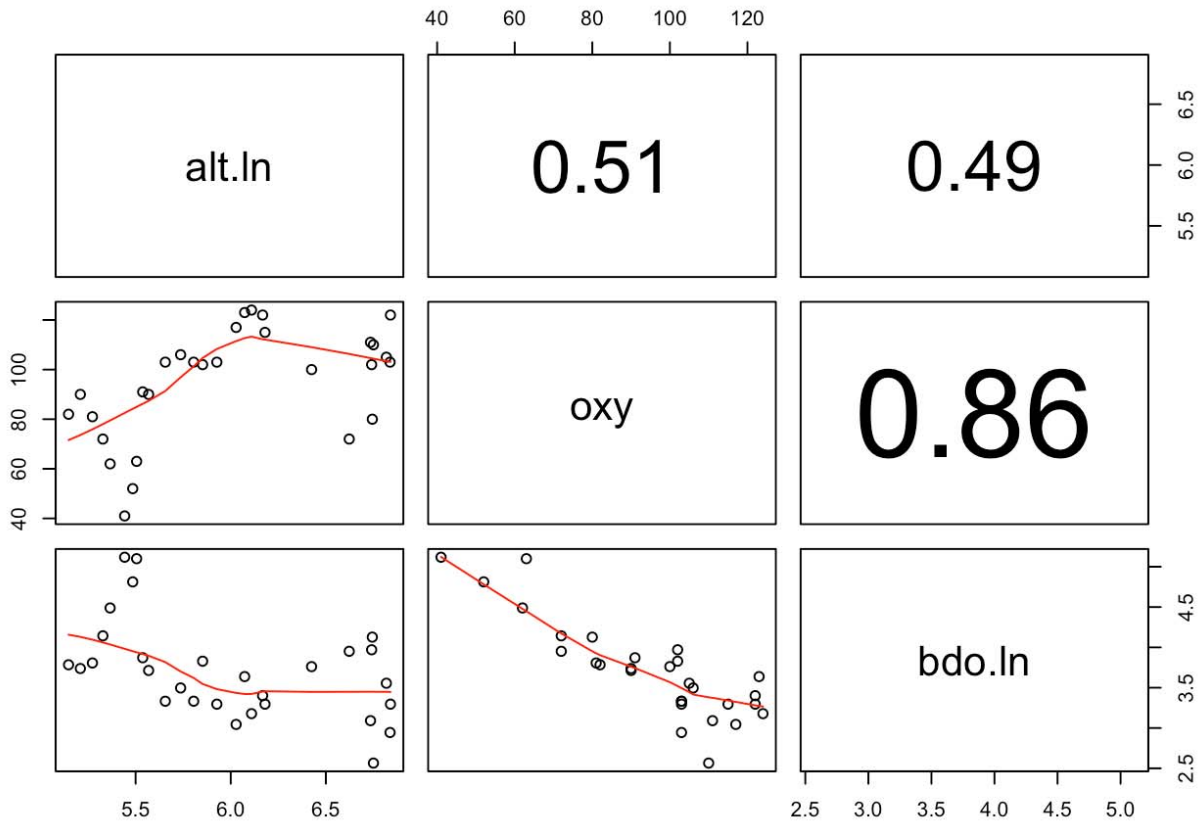
Multicolinealidad.

```
as.dist(cor(env.pars3))
```

```
##            alt.ln        oxy
## oxy     0.5133321
## bdo.ln -0.4903269 -0.8606781
```

Linealidad.

```
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- abs(cor(x, y))
    txt <- format(c(r, 0.123456789), digits = digits)[1]
    txt <- paste(prefix, txt, sep = "")
    if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor * r)
}

pairs(env.pars3, lower.panel = panel.smooth, upper.panel = panel.cor)
```



Realizamos el LDA.

```
# Realizamos el análisis LDA
env.pars3.df <- as.data.frame(env.pars3)
(spe.lda <- lda(gr ~ alt.ln + oxy + bdo.ln, data=env.pars3.df))
```

```
## Call:
## lda(gr ~ alt.ln + oxy + bdo.ln, data = env.pars3.df)
##
## Prior probabilities of groups:
##         1         2         3         4
## 0.3103448 0.3103448 0.2758621 0.1034483
##
## Group means:
##     alt.ln       oxy   bdo.ln
## 1 6.464881 113.88889 3.351171
## 2 6.245151  99.44444 3.512370
## 3 5.385688  83.87500 3.859659
## 4 5.477515  52.00000 5.010015
##
## Coefficients of linear discriminants:
##                LD1         LD2         LD3
## alt.ln -2.5129208 -1.72245625 -1.07719223
## oxy    -0.0773069 -0.02117976  0.08197343
## bdo.ln -0.3348384 -2.71474681  2.30128498
##
## Proportion of trace:
##    LD1    LD2    LD3
## 0.9032 0.0876 0.0092
```

```
#si conociéramos que las probabilidades a priori de cada grupo son distntas a lass obtenidas en las muestras
(defaoult), entonces podemos indicarlas en el análisis a través del argumento "prior"

# vemos los resultados
spe.lda
```

```
## Call:
## lda(gr ~ alt.ln + oxy + bdo.ln, data = env.pars3.df)
##
## Prior probabilities of groups:
##         1         2         3         4
## 0.3103448 0.3103448 0.2758621 0.1034483
##
## Group means:
##     alt.ln       oxy   bdo.ln
## 1 6.464881 113.88889 3.351171
## 2 6.245151  99.44444 3.512370
## 3 5.385688  83.87500 3.859659
## 4 5.477515  52.00000 5.010015
##
## Coefficients of linear discriminants:
##                LD1         LD2         LD3
## alt.ln -2.5129208 -1.72245625 -1.07719223
## oxy    -0.0773069 -0.02117976  0.08197343
## bdo.ln -0.3348384 -2.71474681  2.30128498
##
## Proportion of trace:
##    LD1    LD2    LD3
## 0.9032 0.0876 0.0092
```

Evaluar la magnitud de la discriminación canónica.

Principalmente nos interesan 2 cuestiones:

1. **Valores singulares** (o valores propios en este caso). Los valores propios asociados con cada eje canónico representan la proporción de desviación estándar entre y dentro (o intra) de los grupos en las variables discriminantes lineales. Es decir, estos valores miden qué tan diferenciados están los grupos para la dimensión especificada por la función canónica. La proporción de la discriminación total del primer eje puede ser bastante alta aún cuando la capacidad de separación del grupo es mínima, ya que no mide el grado de diferenciación de los grupos, sino que mide qué cantidad de la diferenciación total se asocia con cada eje, independientemente de la magnitud absoluta en diferenciación de los grupos.
2. **Correlaciones canónicas**. Mide la correlación canónica entre las variables de agrupación y cada eje canónico. La correlación canónica de cada eje toma valores entre 0 y 1 (porque está al cuadrado), donde cero indica la ausencia de relación y los valores cercanos a uno indican mayor asociación. Representa la proporción de variación total en la correspondiente función canónica que es explicada por las diferencias entre las medias de los grupos.

Valores propios.

Para una matriz de datos NxP con G grupos a clasificar, el número de valores propios Q será el menor de $G-1$ y P. En nuestro caso la matriz tiene dimensiones 29x3 y $G=3$, por lo que Q será igual a 3 ($G-1=3$ y $P=3$). Al ponerlo al cuadrado puedo comparar la varianza en lugar de la desviación estándard, para determinar qué tanto poder discriminatorio tiene cada función respecto al total.

```
# Valores propios canónicos
spe.lda$svd^2
```

```
## [1] 53.6898562  5.2088449  0.5478896
```

```
#vemos que la primer función es la que tieme mayor poder discriminatorio

#en términos relativos
spe.lda$svd^2/sum(spe.lda$svd^2) #equivale al "proportion of trace" del modelo spe.lda
```

```
## [1] 0.903161234 0.087622265 0.009216501
```

Correlaciones canónicas.

```
#scores: posición de los objetos en el espacio de covariables o los valores de cada observación en cada eje canónico.
(scores <- predict(spe.lda)$x)
```

```
##            LD1         LD2          LD3
## 1  -4.08638573 -0.89640510  0.368028660
## 2  -2.49450633  0.46365901 -1.995824032
## 3  -2.80466835 -1.20357224 -0.404993616
## 4  -2.68895361  1.49616437 -2.201175466
## 5  -0.87807014 -2.09926524 -1.059020040
## 6  -2.51740981 -2.13333526  0.387269196
## 7  -2.90386963  0.07319673 -0.891988271
## 9   0.10415477 -1.24335518 -1.988893958
## 10 -1.49957974 -0.97965055  0.082158618
## 11 -1.88806718  0.38774388  0.504579631
## 12 -2.43308232 -0.02501061  1.334323268
## 13 -2.36655395  0.63877375  1.047520033
## 14 -2.35214071 -0.52520225  2.062059098
## 15 -1.57722535  1.28900168  0.253631503
## 16 -0.32438764  1.07783858 -0.206474231
## 17 -0.23770978 -0.21870301  1.018179031
## 18 -0.03051363  1.18888940  0.008410463
## 19 -0.14515674  0.79740513  0.706294065
## 20  0.34427132  1.44578197  0.169066314
## 21  1.44181530  0.83677120  0.075460197
## 22  1.38965445  0.44108254  0.553586732
## 23  3.22326372 -2.24627627  1.120313152
## 24  4.22156847 -1.19694492 -0.421313261
## 25  5.07604329 -1.72116640 -0.573615673
## 26  3.85542329 -0.32572787 -0.218162145
## 27  3.29378337  0.46604929 -0.152484763
## 28  2.84858565  1.28339090 -0.129929770
## 29  2.33552915  1.38947026  0.517474958
## 30  3.09418785  1.53939626  0.035520310
```

```
#calculamos la correlación canónica para cada eje
summary(lm(scores~gr))
```

```
## Response LD1 :
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5.1322     0.4663  -11.01 1.74e-11 ***
## gr            2.3624     0.1955   12.09 2.11e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.037 on 27 degrees of freedom
## Multiple R-squared: 0.844,   Adjusted R-squared: 0.8382
## F-statistic: 146.1 on 1 and 27 DF,  p-value: 2.109e-12
##
##
## Response LD2 :
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.09157    0.55125   0.166    0.869
## gr          -0.04215    0.23111  -0.182    0.857
##
## Residual standard error: 1.226 on 27 degrees of freedom
## Multiple R-squared: 0.001231,    Adjusted R-squared: -0.03576
## F-statistic: 0.03327 on 1 and 27 DF,  p-value: 0.8566
##
##
## Response LD3 :
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

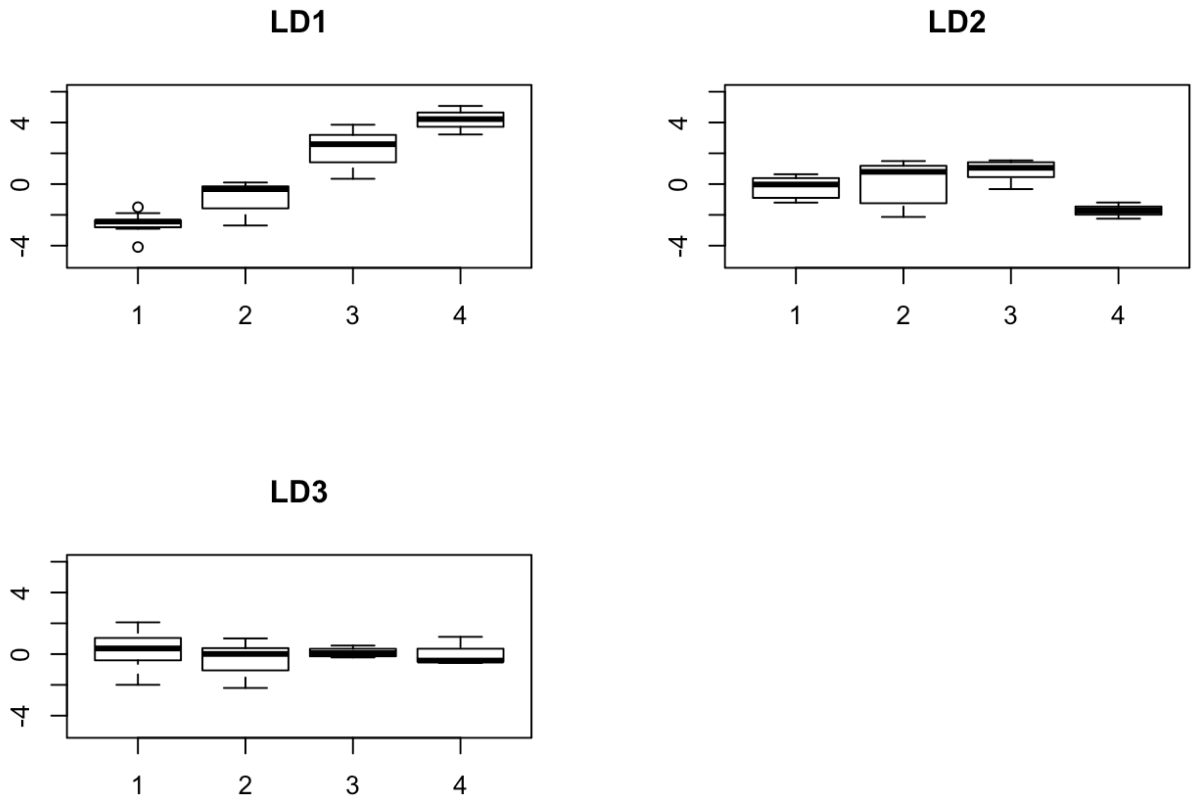
```
## (Intercept)  0.07756    0.44640   0.174    0.863
## gr          -0.03570    0.18714  -0.191    0.850
##
## Residual standard error: 0.9927 on 27 degrees of freedom
## Multiple R-squared: 0.001346,    Adjusted R-squared: -0.03564
## F-statistic: 0.03639 on 1 and 27 DF,  p-value: 0.8501
```

```
#El R-cuadrado múltiple corresponde al valor del coeficiente de correlación canónica (cuadrado)
#es importante tener en cuenta que aunque hayamos cumplido con los supuestos paramétricosla función canónica puede
no discriminar bien entre grupos (como aquí, que el R2m=0.13%) o puede no tener una interpretación ecológica útil.
```

Plot canónico.

Para ver gráficamente la separación canónica de los grupos.

```
x<-data.frame(gr=factor(gr), scores)
par(mfrow=c(2,2))
for(i in 1:ncol(env.pars3))
{
  boxplot(x[,-1][,i]~x[,1],main=colnames(x[,-1])[i], ylim = c(-5, 6)) #todos con igual rango de datos
}
```



Precisión de la clasificación.

```
# observamos las medias de los grupos (clusters) para las 3 variables analizadas
spe.lda$means
```

```
##     alt.ln       oxy   bdo.ln
## 1 6.464881 113.88889 3.351171
## 2 6.245151  99.44444 3.512370
## 3 5.385688  83.87500 3.859659
## 4 5.477515  52.00000 5.010015
```

```
# Vectores propios normalizados. Corresponden a los coeficientes de las funciones discriminantes, estandarizados.
(Cs <- spe.lda$scaling)
```

```
##               LD1         LD2         LD3
## alt.ln -2.5129208 -1.72245625 -1.07719223
## oxy    -0.0773069 -0.02117976  0.08197343
## bdo.ln -0.3348384 -2.71474681  2.30128498
```

```
#scores. posición de los objetos en el espacio de covariables
(Fp <- predict(spe.lda)$x)
```

```
##            LD1         LD2          LD3
## 1  -4.08638573 -0.89640510  0.368028660
## 2  -2.49450633  0.46365901 -1.995824032
## 3  -2.80466835 -1.20357224 -0.404993616
## 4  -2.68895361  1.49616437 -2.201175466
## 5  -0.87807014 -2.09926524 -1.059020040
## 6  -2.51740981 -2.13333526  0.387269196
## 7  -2.90386963  0.07319673 -0.891988271
## 9   0.10415477 -1.24335518 -1.988893958
## 10 -1.49957974 -0.97965055  0.082158618
## 11 -1.88806718  0.38774388  0.504579631
## 12 -2.43308232 -0.02501061  1.334323268
## 13 -2.36655395  0.63877375  1.047520033
## 14 -2.35214071 -0.52520225  2.062059098
## 15 -1.57722535  1.28900168  0.253631503
## 16 -0.32438764  1.07783858 -0.206474231
## 17 -0.23770978 -0.21870301  1.018179031
## 18 -0.03051363  1.18888940  0.008410463
## 19 -0.14515674  0.79740513  0.706294065
## 20  0.34427132  1.44578197  0.169066314
## 21  1.44181530  0.83677120  0.075460197
## 22  1.38965445  0.44108254  0.553586732
## 23  3.22326372 -2.24627627  1.120313152
## 24  4.22156847 -1.19694492 -0.421313261
## 25  5.07604329 -1.72116640 -0.573615673
## 26  3.85542329 -0.32572787 -0.218162145
## 27  3.29378337  0.46604929 -0.152484763
## 28  2.84858565  1.28339090 -0.129929770
## 29  2.33552915  1.38947026  0.517474958
## 30  3.09418785  1.53939626  0.035520310
```

```
# clasificación de los objetos
(spe.class <- predict(spe.lda)$class)
```

```
##  [1] 1 2 1 2 2 1 1 2 2 1 1 1 1 2 2 2 2 2 2 3 3 4 4 4 3 3 3 3 3
## Levels: 1 2 3 4
```

```
# probabilidades posteriores de los objetos que pertenecen a los grupos
(spe.post <- predict(spe.lda)$posterior)
```

```
##               1            2            3            4
## 1  9.858670e-01 1.413296e-02 8.589109e-10 1.420163e-15
## 2  4.940452e-01 5.059487e-01 6.085899e-06 6.418503e-12
## 3  8.588135e-01 1.411862e-01 2.982838e-07 1.233446e-11
## 4  4.792624e-01 5.207302e-01 7.439251e-06 3.764956e-13
## 5  1.881617e-01 8.115272e-01 3.063205e-04 4.798814e-06
## 6  8.836761e-01 1.163235e-01 3.978965e-07 2.998536e-10
## 7  7.973581e-01 2.026411e-01 7.545732e-07 9.620716e-13
## 9  2.171690e-02 9.658988e-01 1.224973e-02 1.345408e-04
## 10 4.808279e-01 5.190573e-01 1.147554e-04 2.862482e-08
## 11 6.122120e-01 3.876919e-01 9.607204e-05 3.223522e-10
## 12 8.718114e-01 1.281831e-01 5.490190e-06 1.869581e-11
## 13 8.146431e-01 1.853415e-01 1.540601e-05 1.071606e-11
```

```
## 14 9.112721e-01 8.872342e-02 4.439650e-06 6.168551e-11
## 15 3.980760e-01 6.011260e-01 7.980082e-04 4.615970e-10
## 16 6.299642e-02 8.902208e-01 4.678228e-02 4.895917e-07
## 17 1.417366e-01 8.258918e-01 3.236084e-02 1.077488e-05
## 18 3.993984e-02 8.237538e-01 1.363046e-01 1.813186e-06
## 19 7.959866e-02 8.283919e-01 9.200680e-02 2.625695e-06
## 20 1.533174e-02 5.625259e-01 4.221367e-01 5.688134e-06
## 21 3.189551e-04 6.254116e-02 9.366679e-01 4.720092e-04
## 22 6.638353e-04 8.237900e-02 9.158176e-01 1.139555e-03
## 23 5.212073e-08 4.643126e-05 2.492989e-02 9.750236e-01
## 24 2.218889e-10 3.122087e-06 5.333234e-02 9.466645e-01
## 25 3.359000e-13 1.801844e-08 2.924798e-03 9.970752e-01
## 26 4.997460e-09 4.293353e-05 5.209165e-01 4.790406e-01
## 27 5.899119e-08 2.395345e-04 9.602048e-01 3.955563e-02
## 28 2.153179e-07 5.145082e-04 9.973440e-01 2.141263e-03
## 29 2.518020e-06 1.857210e-03 9.975367e-01 6.035869e-04
## 30 5.013311e-08 1.724659e-04 9.981144e-01 1.713044e-03
```

```
# tabla de contingencia de las clasificaciones a prori y las predichas
(spe.table <- table(gr, spe.class))
```

```
##    spe.class
## gr  1 2 3 4
##   1 7 2 0 0
##   2 1 8 0 0
##   3 0 1 7 0
##   4 0 0 0 3
```

```
#las filas corresponden a los grupos (los grupos conocidos a priori) y la scolumnas corresponden a loa grupos
predichos por el modelo LDA.
#La diagonal da las predicciones correctas (frecuencia) y lo demás son errores

# proporción de clasificaciones correctas para cada grupo
diag(prop.table(spe.table, 1))
```

```
##         1         2         3         4
## 0.7777778 0.8888889 0.8750000 1.0000000
```

```
#proporción de calsificaciones correctas (CCR) de manera global
sum(diag(spe.table))/sum(spe.table)
```

```
## [1] 0.862069
```

```
#CCR=86%
```

Corregir el azar.

Es importante notar que cierto porcentaje de los datos estarán clasificados correctamente solo por azar. Además, este porcentaje es proporcional al tamaño del grupo. Por lo tanto, si queremos corregir este porcentaje solo debido al azar, podemos utilizar el coeficiente Kappa de Cohen que es apropiado cuando se toman las probabilidades a priori de los grupos como el tamaño muestral (lo que hace lda por defecto).

```
cohen.kappa(spe.table)
```

```
## Call: cohen.kappa1(x = x, w = w, n.obs = n.obs, alpha = alpha)
##
## Cohen Kappa and Weighted Kappa correlation coefficients and confidence boundaries
##                  lower estimate upper
## unweighted kappa  0.63     0.81  0.98
## weighted kappa    0.85     0.93  1.00
##
##  Number of subjects = 29
```

```
#nos indica que el 81% del 86% (CCR) corresponden a clasificaciones correctas que no se deben al azar.
```

Si las probabilidades a priori son conocidas o no se asume que sean iguales a los tamaños muestrales, entonces es conveniente usar el estadístico tau.

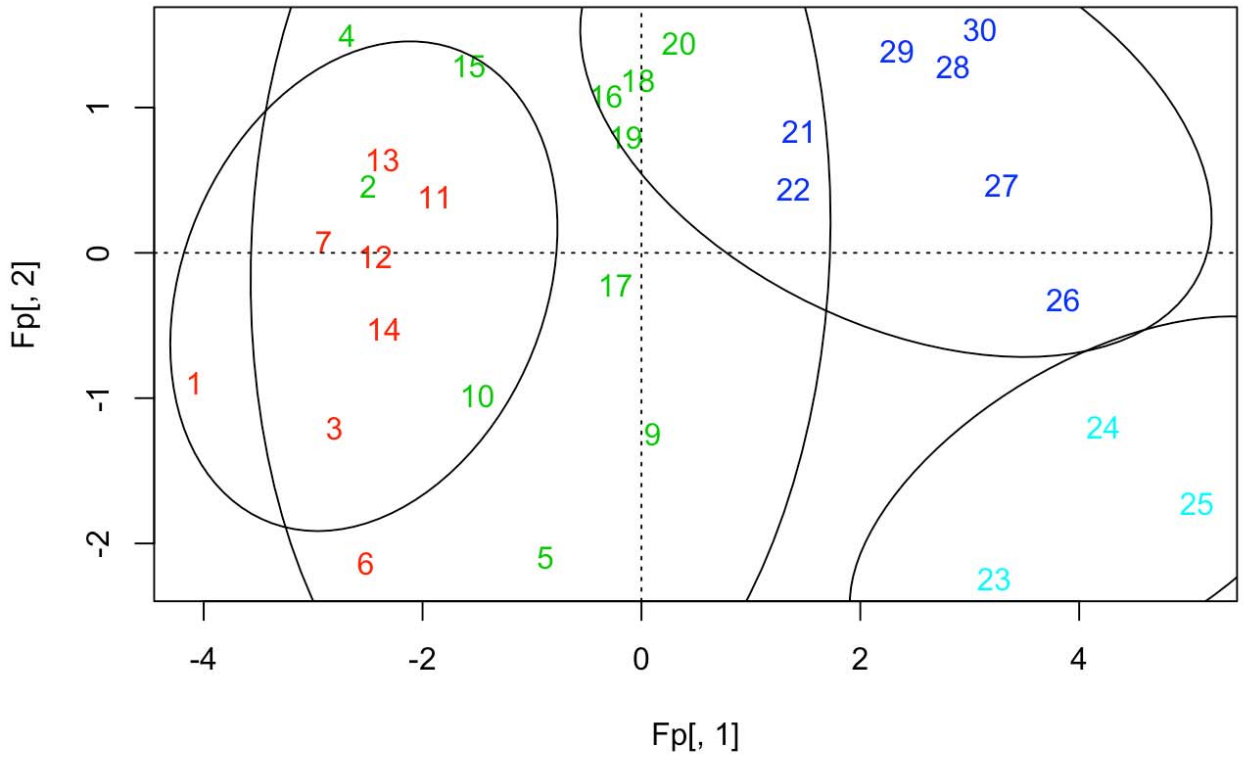
```
cor.test(gr, as.numeric(spe.class),method="kendall")
```

```
## Kendall's rank correlation tau with gr and as.numeric(spe.class)
## z = 5.4891, p-value = 4.04e-08
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.8837404
```

```
#nos indicaría que el 88% del 86% (CCR) corresponden a clasificaciones correctas que no se deben al azar.
```

Gráficos.

```
# Graficamos los objetos en el espacio de covariables

plot(Fp[, 1], Fp[, 2], type="n")
text(Fp[, 1], Fp[, 2], row.names(env), col=c(as.numeric(spe.class)+1))
abline(v=0, lty="dotted")
abline(h=0, lty="dotted")

# elipses del 95% alrededor de los grupos
for(i in 1:length(levels(as.factor(gr))))
{
  cov <- cov(Fp[gr==i, ])
  centre <- apply(Fp[gr==i, ], 2, mean)
    lines(ellipse(cov, centre=centre, level=0.95))
}
```



Validación.

Jackknife.

Evaluamos el éxito de la clasificación mediante validación cruzada, una técnica más realista que las clasificaciones a posteriori vistas anteriormente.

```
# LDA con clasificación basada en el método jackknife (i.e., leave-one-out cross-validation)
(spe.lda.jac <- lda(gr ~ alt.ln + oxy + bdo.ln, data=env.pars3.df, CV=TRUE))
```

```
## $class
##  [1] 1 2 1 1 2 1 1 2 2 1 1 1 1 1 2 2 2 2 2 3 3 4 4 4 4 3 3 3 3
## Levels: 1 2 3 4
##
## $posterior
##               1            2            3            4
## 1  9.856933e-01 1.430672e-02 1.672950e-10 2.340456e-16
## 2  2.052142e-01 7.947717e-01 1.408201e-05 1.575136e-11
## 3  8.264727e-01 1.735268e-01 4.779405e-07 3.687475e-11
## 4  9.296063e-01 7.039361e-02 1.335603e-07 1.981374e-18
## 5  3.608780e-01 6.387156e-01 3.810812e-04 2.530312e-05
## 6  9.921200e-01 7.879994e-03 1.463770e-09 1.358136e-11
## 7  7.506621e-01 2.493366e-01 1.326711e-06 2.232935e-12
## 9  2.540431e-02 9.231870e-01 5.048968e-02 9.190349e-04
## 10 4.093111e-01 5.905328e-01 1.560876e-04 2.306270e-08
## 11 5.758167e-01 4.240631e-01 1.202552e-04 6.768159e-10
## 12 8.452299e-01 1.547597e-01 1.041518e-05 5.549774e-11
## 13 7.725664e-01 2.274049e-01 2.867153e-05 3.153028e-11
## 14 8.673186e-01 1.326691e-01 1.224484e-05 2.925170e-10
## 15 5.108851e-01 4.880450e-01 1.069898e-03 3.173138e-10
## 16 7.462263e-02 8.627123e-01 6.266402e-02 1.043786e-06
## 17 1.894394e-01 7.565910e-01 5.394809e-02 2.154025e-05
## 18 4.593103e-02 7.585763e-01 1.954884e-01 4.215257e-06
## 19 9.989269e-02 7.611944e-01 1.389070e-01 5.986493e-06
## 20 1.464167e-02 8.008569e-01 1.845002e-01 1.212599e-06
```

```
## 21 3.740059e-04 7.139859e-02 9.276098e-01 6.175906e-04
## 22 7.829162e-04 1.031676e-01 8.944178e-01 1.631714e-03
## 23 3.852857e-07 3.730080e-04 1.441493e-01 8.554773e-01
## 24 7.456026e-10 7.136050e-06 8.107595e-02 9.189169e-01
## 25 4.629964e-13 3.321420e-08 6.661268e-03 9.933387e-01
## 26 1.217193e-09 2.224970e-05 2.948547e-01 7.051231e-01
## 27 7.089511e-08 2.921773e-04 9.501719e-01 4.953582e-02
## 28 3.279002e-07 6.627668e-04 9.965100e-01 2.826906e-03
## 29 4.432135e-06 2.499065e-03 9.967172e-01 7.793502e-04
## 30 5.892951e-08 2.011126e-04 9.973493e-01 2.449511e-03
##
## $terms
## gr ~ alt.ln + oxy + bdo.ln
## attr(,"variables")
## list(gr, alt.ln, oxy, bdo.ln)
## attr(,"factors")
##        alt.ln oxy bdo.ln
## gr          0   0      0
## alt.ln      1   0      0
## oxy         0   1      0
## bdo.ln      0   0      1
## attr(,"term.labels")
## [1] "alt.ln" "oxy"    "bdo.ln"
## attr(,"order")
## [1] 1 1 1
## attr(,"intercept")
## [1] 1
## attr(,"response")
## [1] 1
## attr(,".Environment")
## <environment: R_GlobalEnv>
## attr(,"predvars")
## list(gr, alt.ln, oxy, bdo.ln)
## attr(,"dataClasses")
##        gr    alt.ln       oxy    bdo.ln
## "numeric" "numeric" "numeric" "numeric"
##
## $call
## lda(formula = gr ~ alt.ln + oxy + bdo.ln, data = env.pars3.df,
##     CV = TRUE)
##
## $xlevels
## named list()
```

```
summary(spe.lda.jac)
```

```
##           Length Class  Mode
## class      29    factor numeric
## posterior 116    -none- numeric
## terms       3    terms  call
## call        4    -none- call
## xlevels     0    -none- list
```

```
# proporción de clasificaciones correctas
spe.jac.class <- spe.lda.jac$class
spe.jac.table <- table(gr, spe.jac.class)
diag(prop.table(spe.jac.table, 1))
```

```
##         1         2         3         4
## 0.7777778 0.6666667 0.7500000 1.0000000
```

Vemos que esta clasificación no es tan exitosa como la anterior, pero es más realista.

Validación cruzada.

Se realiza una partición previa de los datos en grupo entrenamiento y grupo prueba. Luego, se predice el grupo prueba a partir del LDA ajustado en el grupo entrenamiento.

```
N<-length(gr)
g<-runif(N)
prop<-.5
g[g<prop]<-0
g[g>=prop]<-1
table(g)
```

```
## g
##  0  1
## 15 14
```

```
y<-data.frame(g,gr,env.pars3)

y1<-y[which(y[,1]==0),-1] #entrenamiento
y2<-y[which(y[,1]==1),-1] #prueba

(spe.lda1 <- lda(gr ~ alt.ln + oxy + bdo.ln, data=y1))
```

```
## Call:
## lda(gr ~ alt.ln + oxy + bdo.ln, data = y1)
##
## Prior probabilities of groups:
##          1          2          3          4
## 0.26666667 0.20000000 0.46666667 0.06666667
##
## Group means:
##     alt.ln       oxy   bdo.ln
## 1 6.538269 107.75000 3.474643
## 2 6.410188  98.66667 3.396197
## 3 5.393947  85.57143 3.819162
## 4 5.505332  63.00000 5.099866
##
## Coefficients of linear discriminants:
##               LD1         LD2         LD3
## alt.ln -3.2685918 -0.74295643 -1.03901699
## oxy    -0.0894051 -0.02082013  0.08290566
## bdo.ln -1.8878699 -2.66257060  1.35014154
##
## Proportion of trace:
##    LD1    LD2    LD3
## 0.8879 0.1005 0.0117
```

```
(ytable<-predict(spe.lda1 , newdata=y2))
```

```
## $class
##  [1] 1 1 1 2 1 1 1 2 2 2 3 3 4 3
## Levels: 1 2 3 4
##
## $posterior
##               1            2            3            4
## 1  9.914150e-01 8.585041e-03 2.090272e-10 1.430305e-10
## 6  9.774512e-01 2.254855e-02 9.743152e-09 2.149529e-07
## 7  7.655619e-01 2.344364e-01 1.637641e-06 3.043120e-08
## 9  6.690933e-02 9.198177e-01 9.613536e-03 3.659390e-03
## 11 6.628463e-01 3.369147e-01 2.362858e-04 2.752334e-06
## 12 9.239373e-01 7.605785e-02 4.573623e-06 2.828742e-07
## 13 8.251688e-01 1.747837e-01 4.710027e-05 3.523677e-07
## 15 3.093197e-01 6.824607e-01 8.213967e-03 5.600306e-06
## 16 4.312232e-02 6.355531e-01 3.210430e-01 2.815877e-04
## 17 3.415559e-01 6.120304e-01 4.335694e-02 3.056785e-03
## 18 1.677031e-02 3.416082e-01 6.411955e-01 4.260028e-04
## 24 2.424338e-08 2.572844e-05 7.297762e-01 2.701980e-01
```

```
## 25 9.731521e-10 2.317443e-06 4.057548e-01 5.942429e-01
## 27 2.324294e-08 2.587102e-05 9.966378e-01 3.336338e-03
##
## $x
##           LD1         LD2        LD3
## 1  -4.8513158 -0.15196874  0.8878625
## 6  -4.0130469 -1.45782961  0.2431772
## 7  -3.1384089  0.70025705 -0.1916241
## 9  -0.9099486 -0.69500101 -2.1473318
## 11 -2.0699752  0.48372203  0.9927128
## 12 -2.8538600  0.06673841  1.7282919
## 13 -2.4209469  0.66252488  1.6533702
## 15 -1.2783665  1.22395901  0.9768726
## 16 -0.1698640  0.92159885  0.2608104
## 17 -0.8420838 -0.42069437  0.9749053
## 18  0.1595638  0.91525288  0.4364548
## 24  2.9722689 -1.72547434 -1.4607149
## 25  3.5169170 -2.27920631 -1.9157584
## 27  2.9601555 -0.24389969 -0.5428701
```

```
# proporción de clasificaciones correctas
(spe.class1 <- predict(spe.lda1)$class)
```

```
##  [1] 2 1 2 1 1 1 2 3 3 3 4 3 3 3 3
## Levels: 1 2 3 4
```

```
spe.table <- table(y1$gr, spe.class1)
diag(prop.table(spe.table, 1))
```

```
##         1         2         3         4
## 0.7500000 0.6666667 1.0000000 1.0000000
```

```
sum(diag(spe.table))/sum(spe.table)
```

```
## [1] 0.8666667
```

```
cohen.kappa(spe.table)
```

```
## Call: cohen.kappa1(x = x, w = w, n.obs = n.obs, alpha = alpha)
##
## Cohen Kappa and Weighted Kappa correlation coefficients and confidence boundaries
##                  lower estimate upper
## unweighted kappa  0.55     0.80     1
## weighted kappa    0.82     0.92     1
##
##  Number of subjects = 15
```

Este procedimiento se puede repetir varias veces de tal manera que realizamos un procedimiento Monte Carlo. No lo veremos en este curso.

Clasificaremos un nuevo caso (o sitio).

```
# clasificación de un nuevo objeto (identificación)
# Creamos un nuevo caso con los datos: ln(alt)=6.8, oxygen=90 y ln(bdo)=3.2
newo <- c(6.8, 90, 3.2)
newo <- as.data.frame(t(newo))  # Must be a row
colnames(newo) <- colnames(env.pars3)
newo
```

```
##   alt.ln oxy bdo.ln
## 1    6.8  90    3.2
```

```
(predict.new <- predict(spe.lda, newdata=newo))
```

```
## $class
## [1] 2
## Levels: 1 2 3 4
##
## $posterior
##           1         2            3            4
## 1 0.1388482 0.8609838 0.0001680531 2.984757e-09
##
## $x
##         LD1       LD2       LD3
## 1 -1.481274 0.1095166 -2.433145
```

```
#Vemos que el nuevo caso se clasifica dentro del grupo 1.
```

Ahora que sabemos que el nuevo sitio pertenece al grupo 1, podemos indagar en el perfil de peces que presenta dicho grupo y así predecir la comunidad de peces que tendrá.

Interpretación de las funciones canónicas.

Los vectores propios (o coeficienes canónicos) de las funciones lineales discriminantes corresponden a los datos originales (variables no estandarizadas, no normalizadas) por lo cual no son coeficientes interpretables.

```
spe.lda$scaling
```

```
##               LD1         LD2         LD3
## alt.ln -2.5129208 -1.72245625 -1.07719223
## oxy    -0.0773069 -0.02117976  0.08197343
## bdo.ln -0.3348384 -2.71474681  2.30128498
```

Sin embargo, la estructura canónica de correlaciones (las correlaciones de Pearson entre cada variable discriminante y cada función canónica -scores-) indica la fuerza y naturaleza (positiva o negativa) de la relación y puede utilizarse para interpretar adecuadamente cada eje. Ahora, las variables que tienen mayores cargas nos ayudan a interpretar las funciones canónicas, y estas cargas al cuadrado indican el porcentaje de varianza de la variable que está dada por esa función canónica.

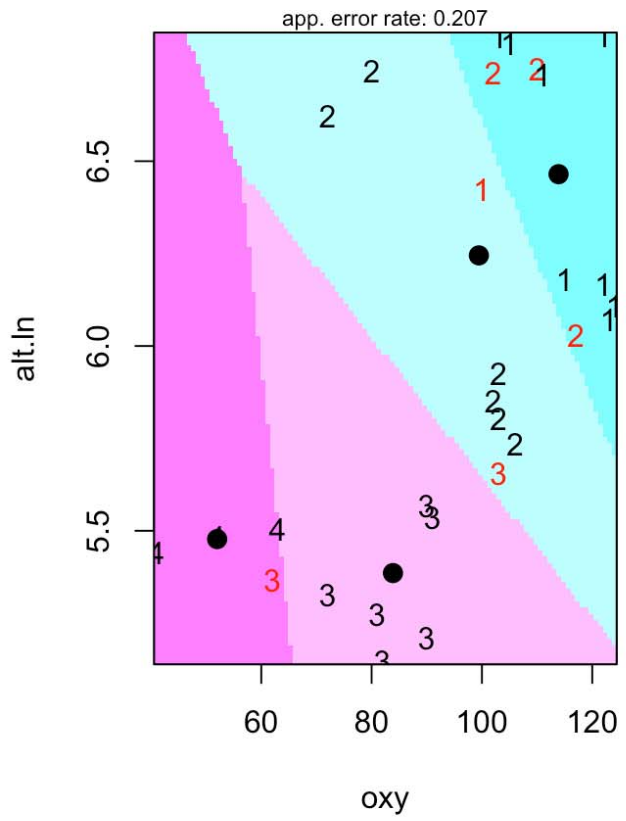
```
x<-y[,-c(1:2)]
dim<-ncol(scores)
cutoff<-0


#calculamos la estructura de correlationes
z<-cor(x,scores[,1:dim])
z<-round(z,digits=2)
z[abs(z)<cutoff]<-substring('',1,nchar(z[1,1]))
(z<-as.data.frame(z))
```

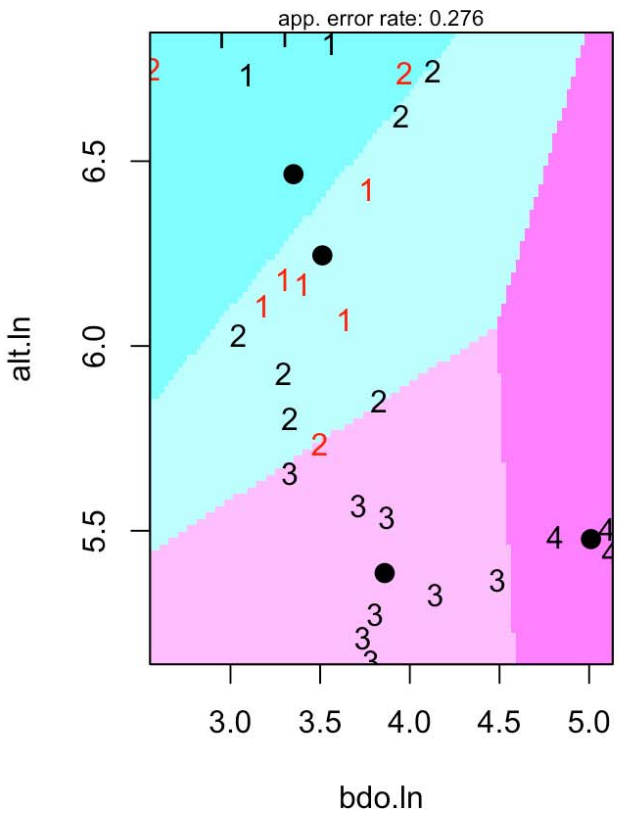
```
##          LD1   LD2   LD3
## alt.ln -0.86 -0.35 -0.38
## oxy    -0.88  0.37   0.3
## bdo.ln  0.76 -0.63  0.14
```

Aquí las tres variables aportan de manera negativa al primer eje, y la más importante corresponde a “alt”. Es decir, “alt” juega un papel importante para la discriminación a lo largo del primer eje.

Más gráficos.

```
#más gráficos útiles
partimat(factor(gr)~., data=y[,-1],method="lda")
```





Partition Plot