Análisis discriminante Lineal (LDA)

Omar Sanchez Hernandez

5/6/2022

Introducción

La técnica de análisis discriminante lineal (LDA) tiene la finalidad de proporcionar reglas de clasificación optimas de nuevas observaciones de las cuales se desconoce su grupo de procedencia basándonos en la información proporcionada por los valores que en ella toman las variables independientes. La técnica usa datos numéricos como regresores para explicar un dato cualitativo, en este caso, la categoría a la que pertenece un sujeto. Esta técnica tiene 2 finalidades:

- **Discriminar a los elementos basados en la información proporcionada.**
- **Clasificar a las nuevas observaciones mediante un modelo.**

Al tratarse de un modelo para inferencia estadística debe cumplir con los supuestos de normalidad multivariada y homocedasticidad multivariada.

Descripción de la matriz de datos

Se tiene un conjunto de datos sobre cráneos recogidos en dos zonas del Tíbet. Los datos que tienen asociado el valor 1 corresponden a tumbas de Sikkim y alrededores.

Los cráneos que tienen asociado el numero 2 corresponden a cráneos encontrados en el campo de batalla de ***Lhasa.*** En total se tienen 32 cráneos de los cuales 17 pertenecen al grupo 1 de ***Sikkim*** y alrededores mientras que los 15 restantes pertenecen al grupo 2 encontrados en el campo de batalla. Las características que se midieron a los cráneos son las siguientes: mayor longitud horizontal del cráneo (x1), mayor anchura horizontal del cráneo (x2), altura del cráneo (x3), altura de la parte superior de la cara (x4) anchura de la cara entre los huesos de las mejillas (x5).

Se cargan las librerías necesarias

```
#Librerías a utilizar
library(ggplot2)
library(MVN)
library(readxl)
library(MASS)
library(Hotelling)
library(scatterplot3d)
library(klaR)
library(biotools)
library(scatterplot3d)
library(knitr)
library(tidyverse)
```

Se carga la matriz de datos

```
#CARGO LOS DATOS DE LOS CRANEOS
rut = "BASE DE CRANEOS DISCRIMINANTE LINEAL.xlsx"
datos <- as.data.frame(read_excel(rut))
attach(datos)
kable(head(datos),caption = "Base de los craneos")
```

Table 1: Base de los craneos

LONG	ANCH	ALT	ALT.C	ANCH.C	TIPO
190.5	152.5	145.0	73.5	136.5	1
172.5	132.0	125.5	63.0	121.0	1
167.0	130.0	125.5	69.5	119.5	1
169.5	150.5	133.5	64.5	128.0	1
175.0	138.5	126.0	77.5	135.5	1
177.5	142.5	142.5	71.5	131.0	1

```
#CONVIERTO TIPO EN VARIABLE DE TIPO FACTOR
datos$TIPO = as.factor(datos$TIPO)
```

Dimensión de la matriz de datos

```
#DIMENSION DE LA MATRIZ DE DATOS
dim(datos)
```

```
## [1] 32  6
```

La matriz de datos cuenta con 32 observaciones y 6 variables.

Nombre de las variables

```
#NOMBRE DE LAS VARIABLES
names(datos)
```

```
## [1] "LONG"   "ANCH"   "ALT"    "ALT.C"  "ANCH.C" "TIPO"
```

Lista de variables

- **LONG:** mayor longitud horizontal del cráneo.
- **ANCH**: mayor anchura horizontal del cráneo.
- **ALT**: altura del cráneo.
- **ALT.C**: altura de la parte superior de la cara.
- **ANCH.C**: anchura de la cara entre los huesos de las mejillas.
- **TIPO**: Lugar donde fue encontrado el cráneo (Sikkim y Lhasa)

Tipo de variables

```
#TIPO DE VARIABLES
str(datos)
```

```
## 'data.frame':    32 obs. of  6 variables:
##  $ LONG  : num  190 172 167 170 175 ...
##  $ ANCH  : num  152 132 130 150 138 ...
##  $ ALT   : num  145 126 126 134 126 ...
##  $ ALT.C : num  73.5 63 69.5 64.5 77.5 71.5 70.5 73.5 70 62 ...
##  $ ANCH.C: num  136 121 120 128 136 ...
##  $ TIPO  : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
```

La matriz de datos esta conformada por 6 variables de las cuales 5 son numéricas y una es de tipo categórica (factor).

Detección de valores faltantes (NA's)

```
#DETECCION DE VALORES FALTANTES NA'S
anyNA(datos)
```

```
## [1] FALSE
```

No se detectaron valores faltantes dentro de la matriz de datos.

Estadísticas descriptivas

Media por variable y grupo

```
#Estadísticas descriptivas
## Medias y varianzas
#Media
m1 = tapply(datos$LONG , datos$TIPO , mean)
m2 = tapply(datos$ANCH , datos$TIPO , mean)
m3 = tapply(datos$ALT , datos$TIPO , mean)
m4 = tapply(datos$ALT.C , datos$TIPO , mean)
m5 = tapply(datos$ANCH.C , datos$TIPO , mean)
medias1 = cbind.data.frame(m1,m2,m3,m4,m5)
names(medias1) = c("Longitud","Ancho","Altura","Altura de cara","Ancho de Cara")
Grupo = c("Sikkim","Lhasa")
medias1 = cbind.data.frame(medias1,Grupo)
kable(medias1,caption ="Medias de las variables por grupo")
```

Table 2: Medias de las variables por grupo

Longitud	Ancho	Altura	Altura de cara	Ancho de Cara	Grupo
174.8235	139.3529	132.0000	69.82353	130.3529	Sikkim
185.7333	138.7333	134.7667	76.46667	137.5000	Lhasa

Se observa que las medias de las variables resultan muy similares en las variables de **Ancho** y **Altura** mientras que en las variables de **Longitud**, **Altura de cara** y **Ancho de cara** presentan diferencias notables.

Desviación estándar por variable y grupo

```
#Desviación estándar
sd1 = tapply(datos$LONG , datos$TIPO , sd)
sd2 = tapply(datos$ANCH , datos$TIPO , sd)
sd3 = tapply(datos$ALT , datos$TIPO , sd)
sd4 = tapply(datos$ALT.C , datos$TIPO , sd)
sd5 = tapply(datos$ANCH.C , datos$TIPO , sd)
sds1 = cbind.data.frame(sd1,sd2,sd3,sd4,sd5)
names(sds1) = c("Longitud","Ancho","Altura","Altura de cara","Ancho de Cara")
Grupo = c("Sikkim","Lhasa")
sds1 = cbind.data.frame(sds1,Grupo)
kable(sds1,caption = "Desviacion estandar de las variables por grupo")
```

Table 3: Desviacion estandar de las variables por grupo

Longitud	Ancho	Altura	Altura de cara	Ancho de Cara	Grupo
6.747549	7.602970	6.007807	4.575550	8.137039	Sikkim
8.626924	6.111659	6.026331	3.911826	4.238430	Lhasa

En la tabla de desviaciones estándar podemos observar como ocurre lo mismo que con la tabla de medias. Las variables longitud, ancho de cara y ancho son muy distintas entre sí, lo que nos indicaría que un grupo tiene mayor dispersión comparado con el otro.

Tratamiento de la matriz de datos

La matriz tal cual esta se encuentra depurada y lista para que se pueda aplicar el correspondiente tipo de análisis debido a que se encuentra organizada y libre de valores faltantes (**NA's**).

Sujetos por grupo

```
sujes = as.data.frame(table(datos$TIPO))
names(sujes) = c("Grupo","Frecuencia")
sujes[,1] = c("Sikkim","Lhasa")
kable(sujes,caption = "Sujetos por grupo")
```

Table 4: Sujetos por grupo

Grupo	Frecuencia
Sikkim	17
Lhasa	15

En la matriz de datos se encuentran presentes 17 observaciones de cráneos encontrados en **Sikkim** y 15 encontrados en **Lhasa**.

Obtención de la muestra de entrenamiento al 80%

Para entrenar al modelo discriminante es necesario separar la matriz de datos en 2 subconjuntos, en el subconjunto de entrenamiento que contiene el **80%** de los sujetos y el restante **20%** se va para probar que tan bueno es el modelo generado.

```
#Se crea la muestra de entrenamiento del 80%
set.seed(2510)
dentre = sample_frac(datos,.8)
kable(dentre,caption = "Muestra de entrenamiento")
```

Table 5: Muestra de entrenamiento

LONG	ANCH	ALT	ALT.C	ANCH.C	TIPO
188.5	130.0	143.0	79.5	136.0	2
171.5	148.5	132.5	65.0	146.5	1
175.0	138.5	126.0	77.5	135.5	1
175.0	153.0	130.0	76.5	142.0	2
181.0	142.0	132.5	79.0	136.5	2
173.5	136.5	126.0	71.5	136.5	2
169.5	130.0	131.0	68.0	119.0	1
182.5	136.0	138.5	76.0	134.0	2
179.5	142.5	127.5	70.5	134.5	1
170.0	126.5	134.5	66.0	118.5	1
183.0	149.0	121.5	76.5	142.0	1
172.0	140.0	136.0	70.5	133.5	1

LONG	ANCH	ALT	ALT.C	ANCH.C	TIPO
197.0	131.5	135.0	80.5	139.0	2
180.5	139.0	132.0	74.5	134.5	1
195.5	144.0	138.5	78.5	144.0	2
196.0	142.5	123.5	76.0	134.0	2
179.5	135.0	128.5	74.0	132.0	2
169.5	150.5	133.5	64.5	128.0	1
182.5	131.0	135.0	68.5	136.0	2
177.5	142.5	142.5	71.5	131.0	1
162.5	139.0	131.0	62.0	126.0	1
185.0	134.5	140.0	81.5	137.0	2
200.0	139.5	143.5	82.5	146.0	2
184.5	141.5	134.5	76.5	141.5	2
173.5	135.5	130.5	70.0	133.5	1
178.5	135.0	136.0	71.0	124.0	1

Obtención de la muestra de prueba

```
#El resto de los datos forman la base de prueba
dpru = setdiff(datos,dentre)
dpru1 = data.frame(dpru[,1:5],clase=as.vector(dpru[,6]))
kable(dpru1,caption = "Muestra de prueba")
```

Table 6: Muestra de prueba

LONG	ANCH	ALT	ALT.C	ANCH.C	clase
190.5	152.5	145.0	73.5	136.5	1
172.5	132.0	125.5	63.0	121.0	1
167.0	130.0	125.5	69.5	119.5	1
179.5	138.0	133.5	73.5	132.5	1
191.0	140.5	140.5	72.5	131.5	2
174.5	143.5	132.5	74.0	136.5	2

Metodología de análisis

Se utilizara el análisis discriminante lineal (**LDA**) debido a que se busca encontrar un modelo para hacer inferencia que logre clasificar bien a los sujetos participantes de la matriz de datos, para ello es necesario que los datos cumplan con los siguientes requisitos:

- Las variables independientes deben ser numéricas continuas.
- La variable dependiente debe ser categórica (en este se tienen 2 categorías).
- Las variables independientes deben cumplir la normalidad multivariada.
- Las variables independientes deben cumplir con la homogeneidad de varianzas multivariada.

Para la normalidad multivariada se recurre a la prueba de Royston

Para probar la normalidad multivariada se usa la prueba de **Royston** que obtiene las distancias de Mahalanobis y las compara frente a una versión mas robusta de la estandarización **Z**.

La prueba de **Royston** contrasta el siguiente juego de hipótesis:

H_0*: Los datos cumplen la normalidad multivariante.*

VS

H_α*: Los datos no cumplen la normalidad multivariante.*

Para probar la homogeneidad de varianzas multivariada se recurre a la prueba de M-Box.

La prueba **M-Box** contrasta el siguiente juego de hipótesis:

H_0*: La matriz de covarianzas de un grupo es igual a la del otro grupo.*

VS

H_α*: La matriz de covarianzas de un grupo es distinta a la del otro grupo.*

Evaluación del modelo

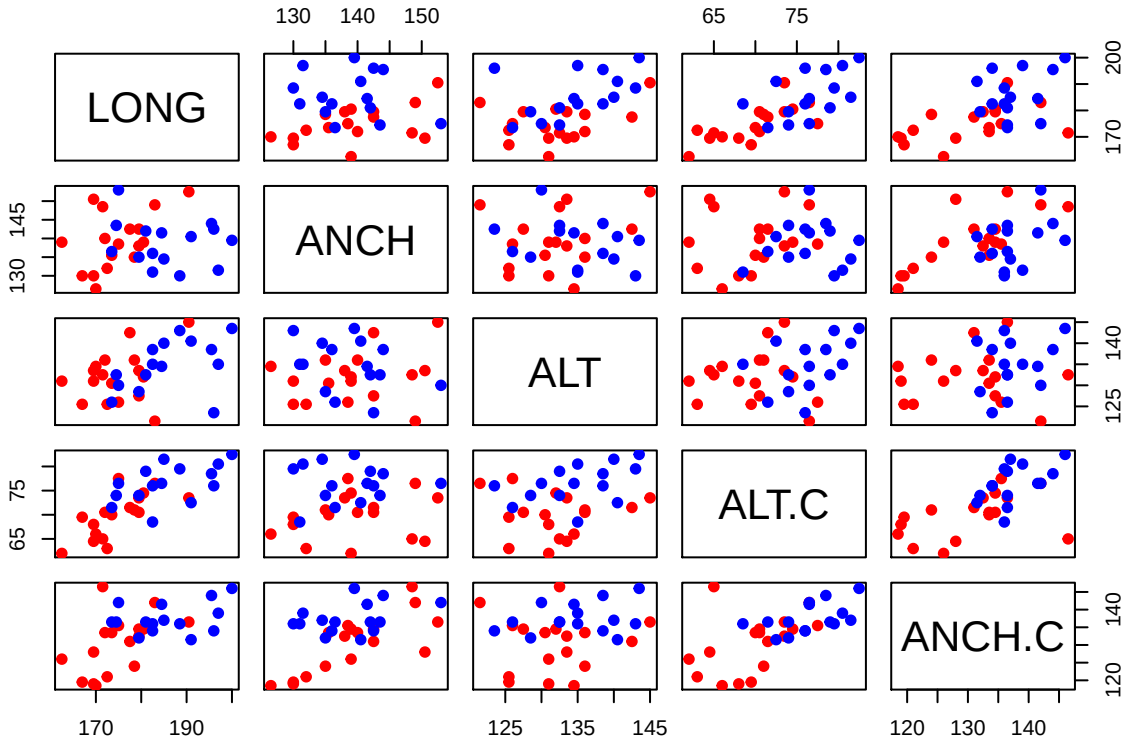
Posterior a la validación de los supuestos que fundamentan el uso del análisis, se procedería a obtener el modelo de discriminante lineal, a evaluar con los mismos datos que lo generaron que tan bien los discrimina y finalmente se procedería a hacer lo mismo pero con los datos del conjunto de prueba para evaluar el desempeño de clasificación del modelo cuando se encuentra con datos que no conoce.

Resultados

Resultados descriptivos

Gráficos de dispersión bivariados por grupo.

```
#Hago el gráfico de dispersión en pares de variables
pairs(x = datos[,1:5],
      col = c("red","blue")[datos$TIPO],pch = 19)
```

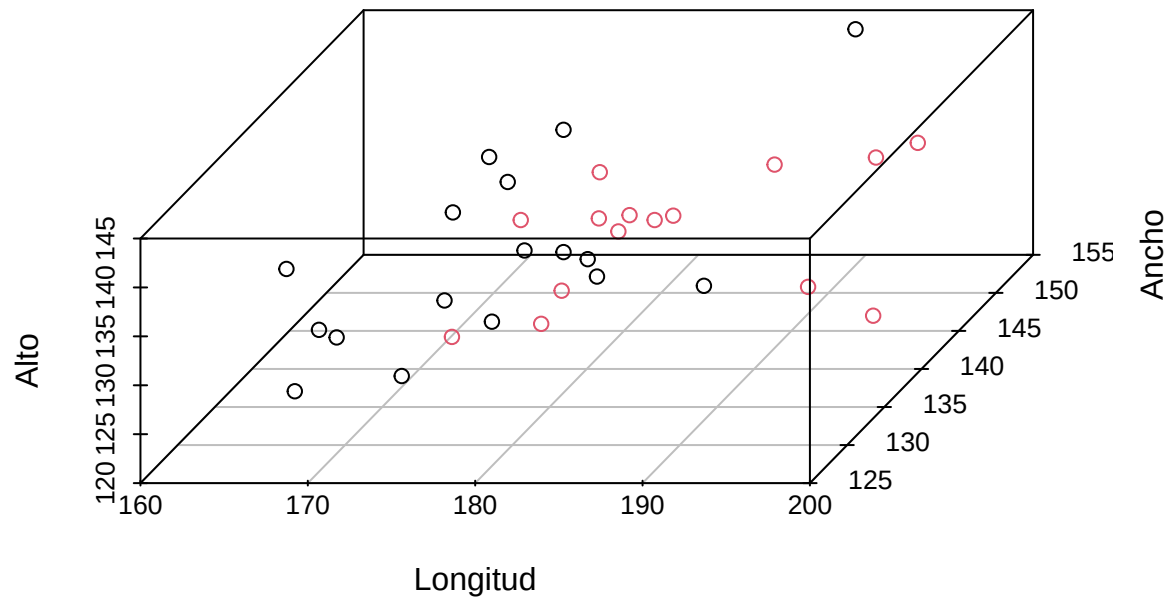


Se observa que los sujetos presentan un grado considerable de traslape por variable.

Gráficos de dispersión 3D

Se propone la realización de un gráfico de dispersión en **3D** con las variables **Longitud**, **Ancho** y **Alto** para observar de mejor manera el comportamiento de los datos.

```
#Scatter plot en 3 dimensiones
scatterplot3d(x = datos$LONG, y = datos$ANCH, z = datos$ALT,
          xlab = "Longitud", ylab =  "Ancho", zlab= "Alto",
            color = datos$TIPO,angle = 430 )
```

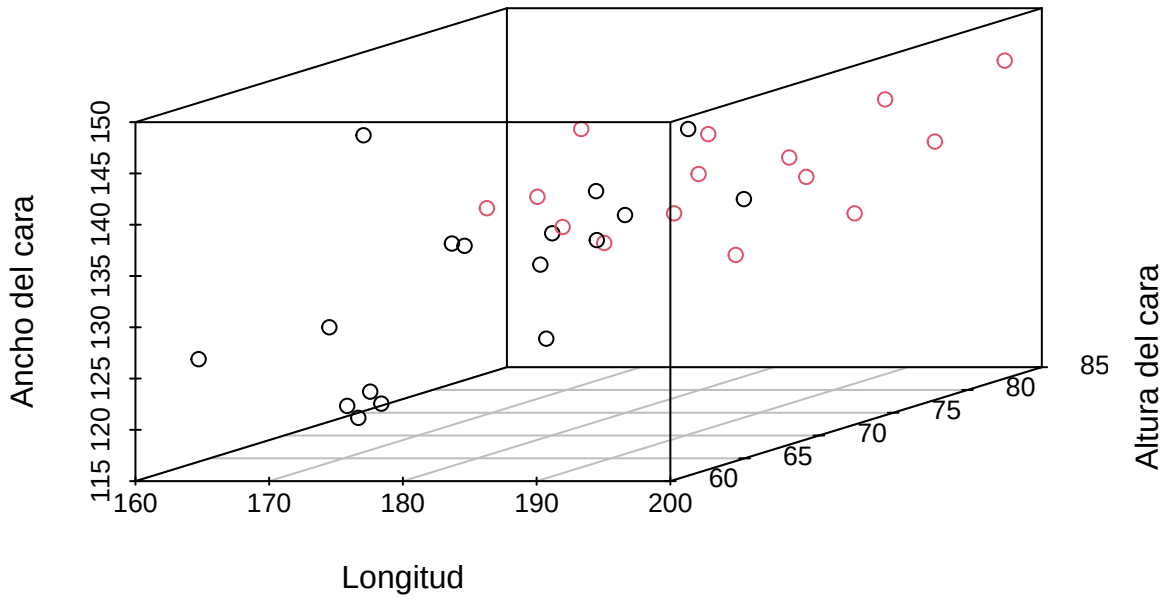


Se observa que si hay una tendencia de los valores a irse hacia los extremos pero en el centro se encuentran varios traslapados entre si.

Gráficos de dispersión 3D con otras variables

Se propone la realización otro gráfico de dispersión en **3D** con las variables **Longitud**, **Altura de la cara** y **Ancho de la cara** para observar de mejor manera el comportamiento de los datos.

```
#Scatter plot con otras variables en 3 dimensiones
scatterplot3d(x = datos$LONG, y = datos$ALT.C, z = datos$ANCH.C,
              xlab = "Longitud", ylab =  "Altura del cara", zlab= "Ancho del cara",
              color = datos$TIPO,angle = 2200 )
```

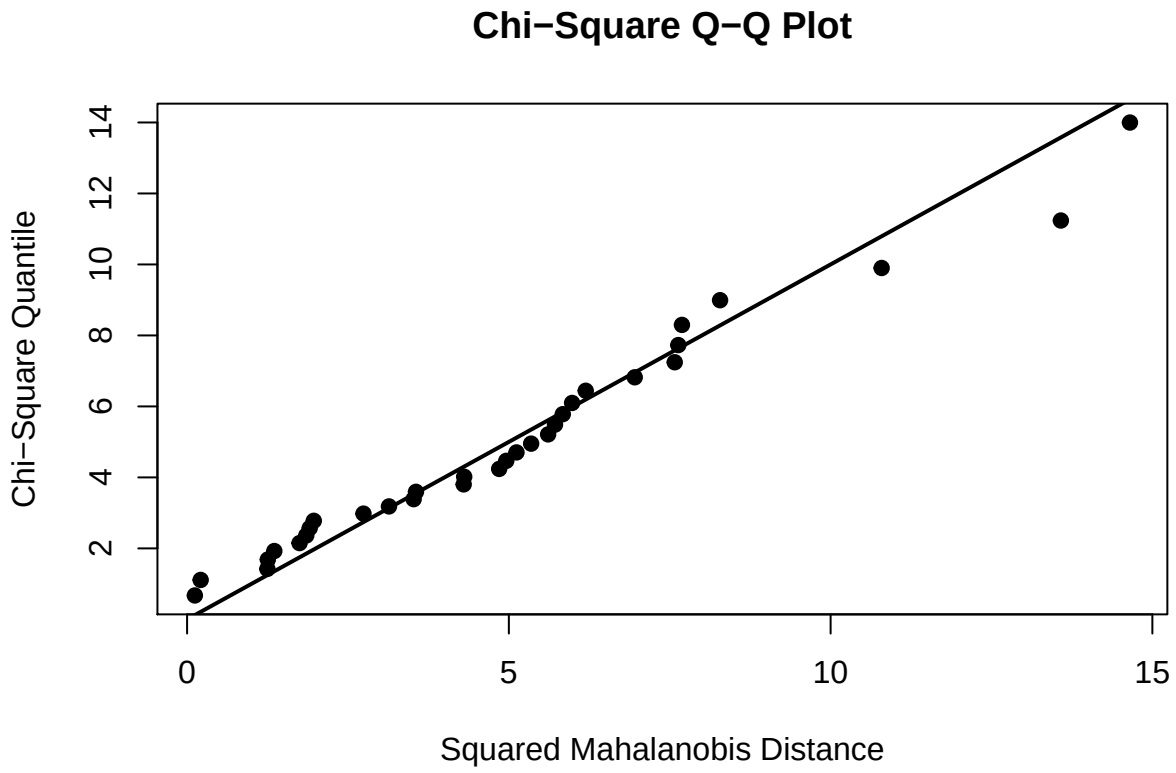


Se observa que nuevamente existe como una tendencia de los grupos hacia irse a los extremos pero en el centro hay varios sujetos que resultan similares en características de los 2 grupos.

Resultados inferenciales

Gráfico qqplot para las distancias de Mahalanobis.

```
#prueba de Royston para normalidad multivariada
pruebaroyston = mvn(data = datos[,-6],mvnTest ="royston",
                    multivariatePlot = "qq")
```



En el gráfico se puede observar como todas la mayoría de las observaciones se ajustan al comportamiento de la recta por lo que podría estar dando un indicio sobre el cumplimiento de la normalidad multivariante.

Normalidad univariada

A continuación se prueba el cumplimiento de la normalidad para cada variable.

```
kable(pruebaroyston$univariateNormality, caption = "Normalidad univariada")
```

Table 7: Normalidad univariada

Test	Variable	Statistic	p value	Normality
Anderson-Darling	LONG	0.4211	0.3048	YES
Anderson-Darling	ANCH	0.3057	0.5470	YES
Anderson-Darling	ALT	0.2250	0.8049	YES
Anderson-Darling	ALT.C	0.1872	0.8962	YES
Anderson-Darling	ANCH.C	0.7763	0.0392	NO

Se observa que todas las variables cumplen la normalidad univariada a excepción de la variable **ANCH.C**.

Normalidad multivariada

A continuación se prueba la normalidad multivariada con la prueba de **Royston**

```
kable(pruebaroyston$multivariateNormality,caption = "Normalidad multivariada" )
```

Table 8: Normalidad multivariada

Test	H	p value	MVN
Royston	4.942763	0.4233818	YES

Se obtiene un p valor de 0.4233818 por lo que **NO** se rechaza la hipótesis nula y se concluye que se cumple el supuesto de normalidad multivariada para la matriz de datos.

```
#Prueba de homocedasticidad MULTIVARIADA
z1 = boxM(data = datos[,1:5],grouping = datos[,6])
zm1 = cbind.data.frame(z1$statistic,z1$parameter,z1$p.value)
names(zm1) = c("Estadistico","G.l.","P-valor")
kable(zm1, caption = "Prueba M-Box")
```

Table 9: Prueba M-Box

	Estadistico	G.l.	P-valor
Chi-Sq (approx.)	18.37051	15	0.2436878

La prueba **M-Box** reporta un valor p = 0.2437 por lo que **NO** se rechaza la hipótesis nula y se concluye que se cumple el supuesto de homocedasticidad multivariante.

Resultados del modelo

```
ir_o = dentre[order(dentre$TIPO),]
#Armo el modelo
ir2.lda = lda(TIPO~.,datos)
ir2.lda
```

```
## Call:
## lda(TIPO ~ ., data = datos)
##
## Prior probabilities of groups:
##       1       2
## 0.53125 0.46875
##
## Group means:
##       LONG     ANCH      ALT    ALT.C   ANCH.C
## 1 174.8235 139.3529 132.0000 69.82353 130.3529
## 2 185.7333 138.7333 134.7667 76.46667 137.5000
##
## Coefficients of linear discriminants:
##                 LD1
## LONG    0.047726591
## ANCH   -0.083247929
## ALT    -0.002795841
## ALT.C   0.094695000
## ANCH.C  0.094809401
```

Se obtiene la matriz de confusión de los datos que generaron al modelo

```
#Para discriminar de las observaciones internas
#Armo la matriz de confusión
predicciones <- predict(object = ir2.lda, newdata = dentre[,-6])
mz1 = table(dentre$TIPO,predicciones$class,dnn=c("Clase real","Clase predicha"))
mz1
```

```
##           Clase predicha
## Clase real  1  2
##          1 10  3
##          2  2 11
```

Se observa que el modelo se equivoca discriminando a 5 sujetos.

Se calcula el error de mala discriminación

```
#El modelo tiene un error al discriminar del:
errorclas <- round(mean(dentre$TIPO!= predicciones$class)*100,4)
paste0("Error de discriminacion = ",errorclas,"%")
```

```
## [1] "Error de discriminacion = 19.2308%"
```

El modelo presenta un error alto de discriminación puesto que esperaríamos que clasifique con un 10% o menor.

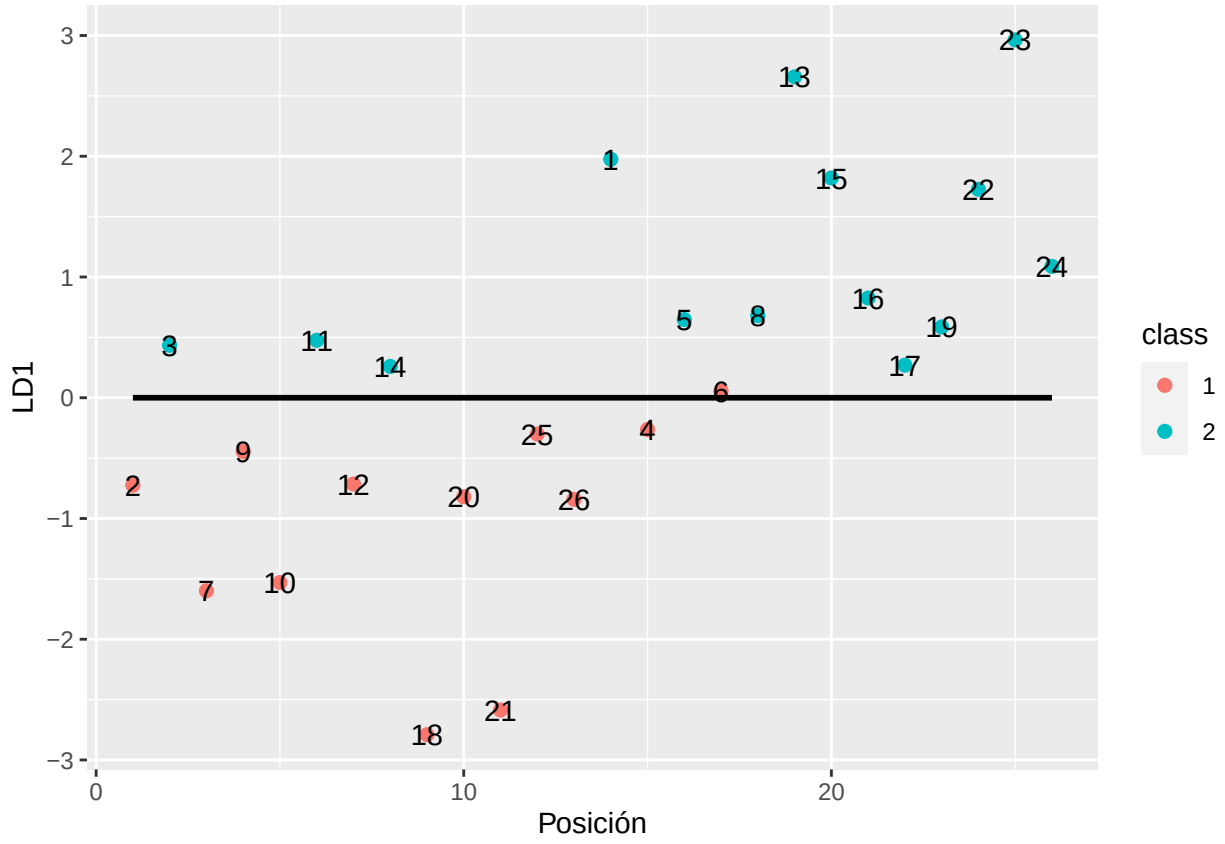
Probabilidades a priori y a posteriori de pertenecer a uno u otro grupo

```
#Armo la matriz de datos
probs = predict(ir2.lda,ir_o,type ="prob")
probs=data.frame(probs)
Posición= 1:nrow(probs)
probs=data.frame(cbind(probs,Posición))
kable(probs)
```

	class	posterior.1	posterior.2	LD1	Posición
2	1	0.8306539	0.1693461	-0.7244950	1
3	2	0.3595093	0.6404907	0.4339844	2
7	1	0.9618142	0.0381858	-1.5988413	3
9	1	0.7444719	0.2555281	-0.4461058	4
10	1	0.9568059	0.0431941	-1.5301904	5
11	2	0.3416798	0.6583202	0.4758413	6
12	1	0.8280092	0.1719908	-0.7145095	7
14	2	0.4377222	0.5622778	0.2591872	8
18	1	0.9957489	0.0042511	-2.7905613	9
20	1	0.8544783	0.1455217	-0.8206345	10
21	1	0.9937864	0.0062136	-2.5866630	11
25	1	0.6892169	0.3107831	-0.3002743	12
26	1	0.8592412	0.1407588	-0.8413888	13
1	2	0.0304314	0.9695686	1.9751662	14
4	1	0.6739715	0.3260285	-0.2627278	15
5	2	0.2734243	0.7265757	0.6476551	16
6	1	0.5326224	0.4673776	0.0555298	17
8	2	0.2612598	0.7387402	0.6808490	18
13	2	0.0086793	0.9913207	2.6574602	19
15	2	0.0402604	0.9597396	1.8201428	20
16	2	0.2123179	0.7876821	0.8259841	21
17	2	0.4328100	0.5671900	0.2698668	22
19	2	0.2968229	0.7031771	0.5862804	23
22	2	0.0476383	0.9523617	1.7260944	24
23	2	0.0049098	0.9950902	2.9639477	25
24	2	0.1416841	0.8583159	1.0880400	26

Gráfico de las probabilidades de discriminación del modelo

```
attach(probs)
ggplot(probs, aes(Posición,LD1)) +
  geom_point(aes(color = class),size=2) +
  geom_line(aes(y=0), size=1) +
  geom_text(label=rownames(probs))
```



En el gráfico se puede observar como se hace la separación de los valores en donde los que se encuentran por encima de la linea marcada pertenecen a una clase y los que están por debajo pertenecen a otra pero como tenemos valores mal discriminados entonces se puede observar que hay valores de una clase en la sección que le pertenece a la otra y viceversa.

Se obtiene la matriz de confusión de los datos de prueba

```
#Clasificando con la prueba
#Armo la matriz de confusión
predicciones1 <- predict(object = ir2.lda, newdata = dpru1[,-6])
zmk1 = table(dpru1$clase,predicciones1$class,dnn = c("Clase real","Clase predicha"))
zmk1
```

```
##           Clase predicha
## Clase real 1 2
##          1 4 0
##          2 1 1
```

El modelo únicamente se equivoca clasificando a 1 sujeto.

Se calcula el error de mala clasificación

```
#Se observa que hay 7 sujetos mal clasificados
error1_entre = round(mean(dpru1$clase != predicciones1$class)*100,4)
paste0("error de clasificación = ",error1_entre, "%")
```

```
## [1] "error de clasificación = 16.6667%"
```

El error de clasificación resulta de 16.66% lo cual resulta un valor alto puesto que nosotros esperaríamos que el modelo se equivoque un 10% o menos.

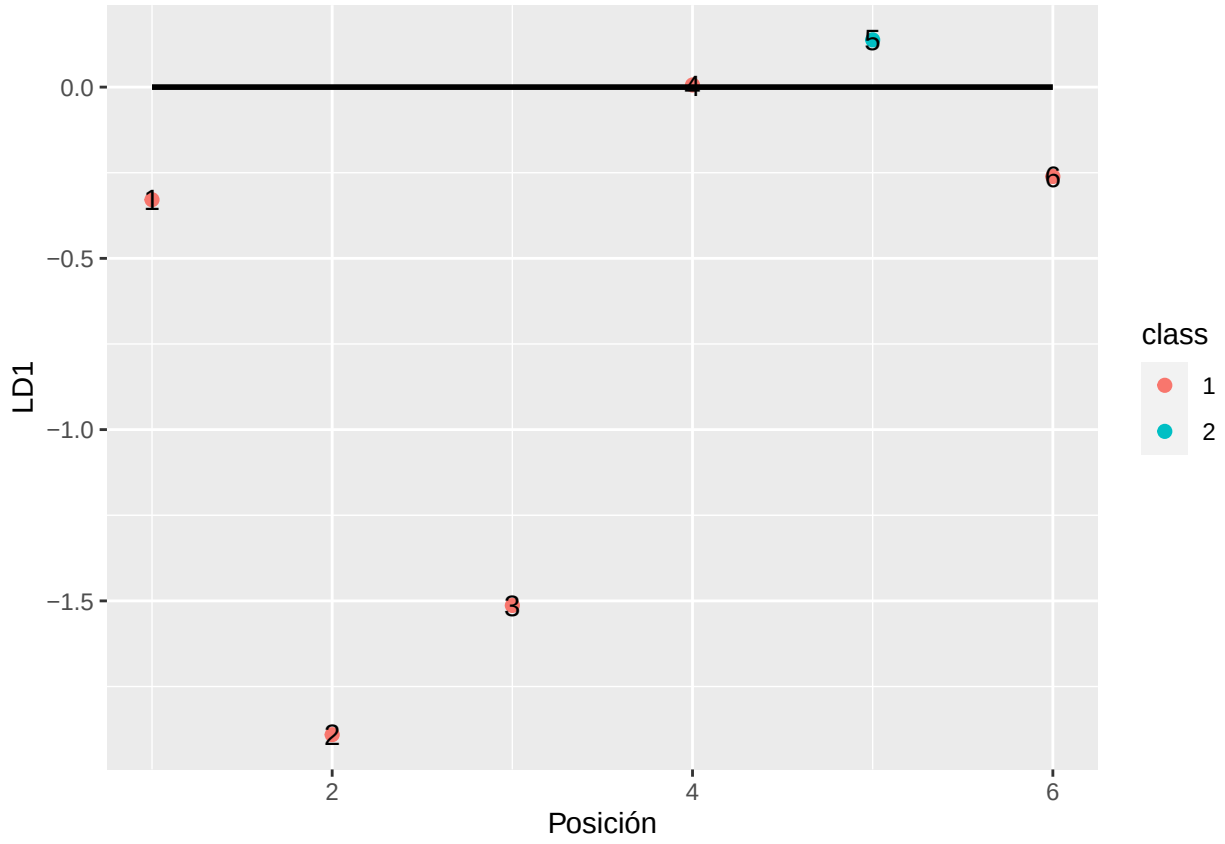
Probabilidades a priori y a posteriori de pertenecer a uno u otro grupo

```
#Armo el gráfico
probs = predict(ir2.lda,dpru1 ,type ="prob")
probs=data.frame(probs)
Posición= 1:nrow(probs)
probs=data.frame(cbind(probs,Posición))
kable(probs)
```

class	posterior.1	posterior.2	LD1	Posición
1	0.7005395	0.2994605	-0.3288160	1
1	0.9775195	0.0224805	-1.8906364	2
1	0.9554833	0.0445167	-1.5133334	3
1	0.5555151	0.4444849	0.0062010	4
2	0.4941539	0.5058461	0.1378617	5
1	0.6732256	0.3267744	-0.2609146	6

Gráfico de las probabilidades de discriminación del modelo

```
attach(probs)
ggplot(probs, aes(Posición,LD1)) +
  geom_point(aes(color = class),size=2) +
  geom_line(aes(y=0), size=1) +
  geom_text(label=rownames(probs))
```



En el gráfico se puede observar como un valor que pertenece a la clase 1 se encuentra en la sección de la clase 2 debido a que el modelo determina por apenas muy poco que debe pertenecer a la clase 2 aunque originalmente sea de la clase 1.

Conclusiones

Tras observar los gráficos exploratorios se concluye que el problema quizás no sea un buen candidato para ser separado de manera lineal puesto que existe mucho traslape entre los sujetos de cada grupo.

El modelo de discriminante lineal al que se llega resulta bueno pero no ideal puesto que su error de discriminación es del **19.2308%** y su error de discriminación es del **16.66%** lo cual no resulta tan beneficioso de cara a buscar un modelo que clasifique con mayor precisión.

Finalmente, se concluye haciendo mención de que los datos no son buenos candidatos para ser separados linealmente por el grado de traslape y similitud que tienen los de un grupo con otro pero posiblemente se puedan llegar a mejores resultados si se aplica un modelo discriminante cuadrático (**QDA**) o incluso con la aplicación de las maquinas de soporte vectorial.

Referencias

- Al, I., & Discriminante, A. (n.d.). Tema 1: Análisis Discriminante Lineal. Uc3m.Es. Retrieved June 5, 2022, from http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema1dm.pdf

Lista de paquetes utilizados

- ***ggplot2***
- ***MVN***
- ***readxl***
- ***MASS***
- ***Hotelling***
- ***scatterplot3d***
- ***klaR***
- ***biotools***
- ***scatterplot3d***
- ***knitr***
- ***tidyverse***