# Práctica (PR). Análisis Multivariante

Pérez Efremova, Daniel

Enero 2021

**Resumen**

En esta práctica vamos a utilizar el conjunto de datos *Iris.csv*, una vdistribución moderna del conjunto de datos original que usó R.A. Fisher en su artículo acerca del análisis discriminante, y que contiene sendas variables acerca de tres especies de flores. El objetivo es usar las técnicas del análisis discriminante, vistas durante el curso, para proponer un modelo de clasificación de estos tipos de flores.

# Índice



# Índice de Códigos

# 1. Introducción al conjunto de datos

El conjunto de datos es una versión moderna del conjunto de datos original que utilizó R.A. Fisher en uno de sus artículos acerca del análisis discriminante [1], concretamente, a cerca de la función lineal discriminante que ahora lleva su nombre.

El conjunto de datos contiene cuatro variables:

1. *SepalLengthCm*, que es la longitud del sépalo de la pieza floral en centímetros.
2. *SepalWidthCm*, la anchura del sépalo de la pieza floral.
3. *PetalLengthCm*, la longitud del pétalo de la pieza.
4. *PetalWidthCm*, la anchura del pétalo de la pieza.
5. *Specie*, especie a la que pertenece la pieza.

Además puede obtenerse facilmente en R con el siguiente comando.

```
data(iris) # los datos se cargan automaticamente en el dataframe iris
```

# 2. Análisis descriptivo

En esta sección vamos a describir los datos con el objetivo de seleccionar las variables más adecuadas para definir una función discriminante que nos permita clasificar las flores en las distintas especies.

Además, debemos comprobar que se cumplen las especificaciones para construir una función lineal discriminante:

1. La distribución conjunta de las variables en cada uno de los grupos debe ser una normal multivariante.
2. Las medias de cada grupo no son todas iguales.
3. Las matrices de varianzas y covarianzas entre los grupos deben ser iguales.

Por sencillez en la exposición, escogeremos dos variables para construir el discriminante.

## 2.1. Selección de las variables

En la Figura 1 observamos una matriz que recoge diversos gráficos que nos ayudaran a escoger las variables para construir la función discriminante.

En primer lugar, en la diagonal principal vemos que la distribución de cada variable parece ser una mezcla de distribuciones normales (a falta de confirmación con un test de normalidad). Por lo que parece razonable escoger dos variables que separen lo máximo posible el centro de mezclas. Por ejemplo, seleccionaremos *Sepal.Length* y *Petal.Length* donde las mezclas parecen estar relativamente separadas (distintos vectores de medias).

## 2.2. Comprobación de las especificaciones para construir una función discriminante lineal

Comprobamos ahora las especificaciones del análisis discriminante lineal. En primer lugar en el Código 1 comprobamos la normalidad multivariante de las variables en cada uno de los grupos.

Código 1: Test de Mardia sobre cada grupo

```
library('MVN') # libreria de R para la diagnosis de normalidad multi.

for(specie in species){
        normality_tests = mvn(filter(iris, iris['Species'] == specie)[,
            c(1,3)],
        cov = TRUE, mvnTest='mardia')
        print(normality_tests$multivariateNormality)
}

[1] "setosa"

        Test         Statistic            p value Result
1 Mardia Skewness 6.10611658571657 0.191362343322577    YES
2 Mardia Kurtosis 1.16825186615433 0.242705184818886    YES
3             MVN              <NA>               <NA>    YES

[2] "versicolor"

        Test           Statistic            p value Result
1 Mardia Skewness   6.63800201068544 0.156300181771164    YES
2 Mardia Kurtosis -0.763571602727455 0.445122577300795    YES
3             MVN              <NA>               <NA>    YES

[3] "virginica"
        Test           Statistic            p value Result
1 Mardia Skewness    3.8568432348427 0.425726551515805    YES
2 Mardia Kurtosis -0.529059167515372 0.596764405862679    YES
3             MVN              <NA>               <NA>    YES
```

Y comprobamos que efectivamente podemos asumir normalidad multivariante en cada una de las variables dentro de cada grupo.

Por otro lado, la hipótesis de homocedasticidad requiere de una generalización de los test univariantes de igualdad de varianzas al caso multivariante para varias poblaciones normales. Aquí he escogido el propuesto por Ahmad 2017 cuyo resultado se muestra en el Código 2.

Código 2: Test de homogeneidad de matrices de covarianzas

```
library(covTestR)

irisSpecies <- unique(iris$Species)
```

```
iris_ls <- lapply(irisSpecies,
function(x){as.matrix(iris[iris$Species == x, c(1,3)])}
)

names(iris_ls) <- irisSpecies

Ahmad2017(iris_ls)

        Ahmad 2017 Homogeneity of Covariance Matrices Test

data:  setosa, versicolor and virginica
Standard Normal = 0.82744, Mean = 0, Variance = 1, p-value = 0.204
alternative hypothesis: true difference in covariance matrices is not
    equal to 0
```

Y tenemos que podemos asumir igualdad de matrices de covarianzas.

Por último realizamos un test de igualdad de medias para comprobar que no todas los grupos tienen la misma media (el mismo centro). Lo vemos en el Código 3.

Código 3: Test de igualdad de medias

```
n = dim(iris)[1]
p=2
G=3
g = p*(G-1)

S = cov(iris[, c(1,3)])

S_vir = cov(filter(iris, iris['Species'] == 'virginica')[, c(1,3)])
S_set = cov(filter(iris, iris['Species'] == 'setosa')[, c(1,3)])
S_versi = cov(filter(iris, iris['Species'] == 'versicolor')[, c(1,3)])

S_w = (S_vir+S_versi+S_set)*(1/3)

stat = n*log(det(S)/det(S_w))  # estadistico de contraste
print(c('pvalor', pchisq(stat, g, lower.tail=FALSE))
[1] "pvalor"    "2.07455591527702e-102"
```

Y vemos que el pavalor nos permite rechazar que las medias de los grupos son todas iguales.

Concluimos que las variables *Sepal.Length* y *Petal.Length* cumplen con las especificaciones para construir un modelo de clasificación entre las distintas especies a través de una función discriminante lineal.

# 3. Planteamiento teórico del modelo e implementación

Vamos a construir ahora el sustento teórico del modelo de clasificación mediante una función discriminante lineal en condiciones de normalidad multivariante en los grupos ($G$) e igualdad de varianzas.

Al desconocer las medias y matriz de covarianzas poblacional de los grupos,

las estimaremos mediante sus estimadores muestrales usuales que son:

$$\bar{x}_g = \frac{1}{n_g}\sum_{g\in G} x_{gi}, \;\; S_w = \frac{1}{G}\sum_{g\in G}(x_{ig}-\bar{x}_g)(x_{ig}-\bar{x}_g)'$$

Ahora, en las condiciones que hemos probado anteriormente, el problema de discriminación se resuelve clasificando una observación $x_0$ en un grupo $g \in G$ segun la regla:

$$g_0 = \min_{g\in G}(x_0-\bar{x}_g)'\mathbf{S}_w^{-1}(x_0-\bar{x}_g) = \min_{g\in G} D(x_0, x_g)$$

donde $D$ es la distancia de Mahalanobis, con peso la matriz de covarianzas muestral, a cada una de las medias muestrales.

La probabilidad de error con esta regla viene definida por:

$$p_{error} = \frac{\text{total mal clasificados}}{\text{total bien clasificados}}$$

Para la implementación, siguiendo las recomendaciones del texto base, hemos dividido el conjunto de datos en una parte de entrenamiento y otra de validación (20 % de los datos validación) con un muestreo aleatorio simple del conjunto total.

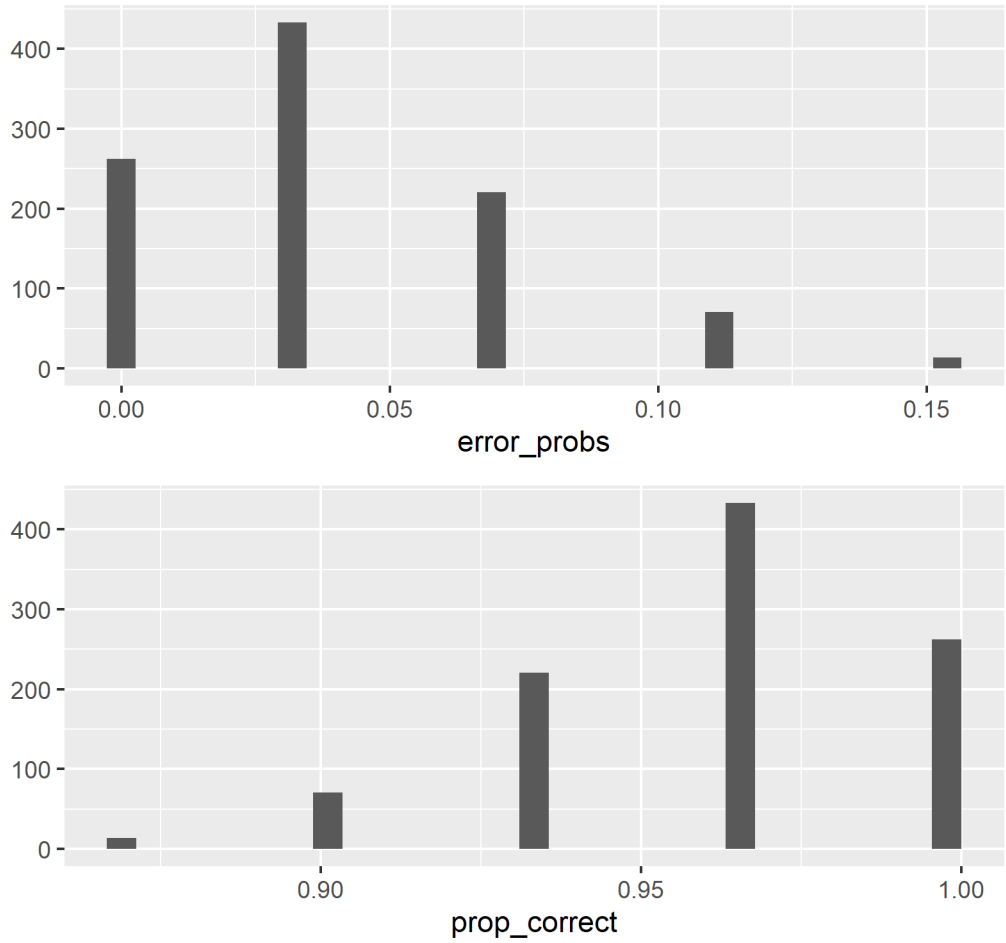
Código 4: Ajuste del modelo y simulación de experimentos

```
error_probs = c() # vector que almacena las probabilidades de error de
    cada ajuste
prop_correct = c() # vector que almacena la proporcion de clasificados
    correctamente en el conjunto de validacion

for(i in 0:1000){

        set.seed(i) # variamos la semilla aleatoria en cada experimento

        # division en entrenamiento y test

        sample <- sample.int(n = nrow(iris), size = floor(.8*nrow(iris))
            , replace = F)
        train <- iris[sample, ]
        test  <- iris[-sample, ]

        #calculo de los estadisticos muestrales

        means_vir = apply(filter(train, train['Species'] == 'virginica')
            [, c(1,3)], 2, mean)
        S_vir = cov(filter(train, train['Species'] == 'virginica')[, c
            (1,3)])

        means_set = apply(filter(train, train['Species'] == 'setosa')[,
            c(1,3)], 2, mean)
        S_set = cov(filter(train, train['Species'] == 'setosa')[, c(1,3)
            ])

        means_versi = apply(filter(train, train['Species'] == '
            versicolor')[, c(1,3)], 2, mean)
        S_versi = cov(filter(train, train['Species'] == 'versicolor')[,
            c(1,3)])
```

```
        S_hat = (1/3)*(S_vir + S_set + S_versi) # matriz de covarianzas
            muestral

        # funciones discriminantes de setosa, virginica y versicolor
        # (equivalente a la minima distancia de mahalanobis al centro de
            cada grupo)

        Mah_set = function(x){

                x = matrix(x, 2,1)
                return((t(x-means_set)%*%inv(S_hat)%*%(x-means_set))
                    [1,1])
        }

        Mah_vir = function(x){

                x = matrix(x, 2,1)
                return((t(x-means_vir)%*%inv(S_hat)%*%(x-means_vir))
                    [1,1])
        }

        Mah_versi = function(x){

                x = matrix(x, 2,1)
                return((t(x-means_versi)%*%inv(S_hat)%*%(x-means_versi
                    ))[1,1])
        }

        # regla de decision

        lda_classicator = function(x){

                specie = c('pred_versicolor', 'pred_setosa', 'pred_
                    virginica')
                distances = c(Mah_versi(x), Mah_set(x), Mah_vir(x))

                return(specie[which.min(distances)])
        }

        # clasificacion sobre los valors usados en el ajuste y sobre
            test (no observados)
        test$pred = apply(test[, c(1,3)], 1,min_dist)
        train$pred = apply(train[, c(1,3)], 1,min_dist)

        crosstab = matrix(ftable(test$Species, test$pred), 3, 3)

        correct = sum(diag(crosstab)) # correctamente clasificados
        conf = dim(test)[1]-correct # incorrectamente clasificados

        prob_error = conf/correct
        propor_correct = correct/dim(test)[1]

        error_probs = append(error_probs, prob_error)
        prop_correct = append(prop_correct, propor_correct)
}
```

Vemos en la Figura 2 las probabilidades de error calculadas en cada experimento junto a la proporción de clasificados correctamente. Podemos concluir que los resultados son bastante aceptables, puesto que en promedio tenemos una probabilidad de error alrededor de 0,05 con una proporción de clasificados correctamente mayor de 0,95, y aun que algunos experimentos han resultado en probabilidades de error algo mas altas, no tenemos evidencia para creer que el modelo no sea adecuado.

Figura 2: Histograma de las probabilidades de error y proporciones de clasificados correctamente durante los experimentos




## 4. Conclusión

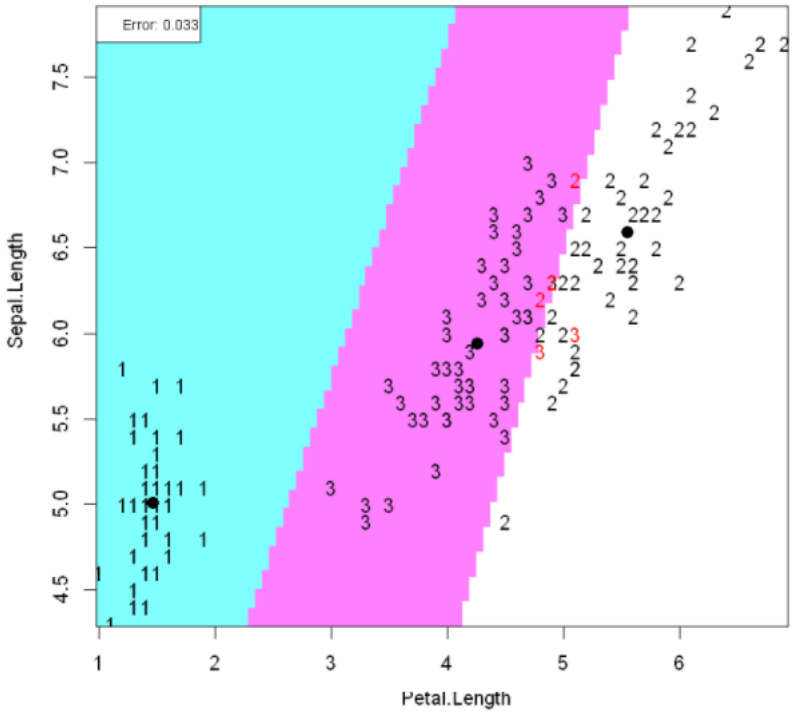
A la vista de los resultados obtenidos, si en algún momento observamos una nueva pieza floral dentro de las poblaciones donde se observó la muestra, tan solo nos basta conocer la longitud del sépalo y pétalo para poder clasificarla en una de las tres especies presentes en la muestra con fuertes garantías experimentales de que equivocarnos es poco probable.

Cabe mencionar también que el modelo se ha construido usando dos variables y los resultados han sido aceptables, así que cabría estudiar si se pueden obtener unos resultados similares usando tan solo una de ellas. Por supuesto,

no hay evidencia experimental que nos sugiera introducir más variables, ya que los errores altos de los experimentos pueden deberse a grupos de datos atípicos como observábamos en la Figura 1 al inicio.

Ajustando finalmente el modelo sobre todo el conjunto de datos podemos visualizar las regiones que delimitan las funciones discriminantes lineales [1].

Figura 3: División del espacio




# Bibliografía

[1] Fisher, Ronald A. "The use of multiple measurements in taxonomic problems. Annals of eugenics 7.2 (1936): 179-188.

[2] Kaggle. https://www.kaggle.com/uciml/iris

[1] Hemos cambiado los nombres de las clases para facilitar la lectura del gráfico, el 1 es setosa, 2 es virginica y 3 versicolor

Figura 1: Matriz de gráficos de dispersión, funciones de densidad univariante y diagramas de cajas