Estadística y Técnicas de Machine Learning con R

Técnicas de Machine Learning

true

22 de julio de 2018

Contents

Introducción

- El campo de estudio interesado en el desarrollo de algoritmos de computadora para transformar datos en acción inteligente se conoce como **aprendizaje automático (machine learning)**.
- Este campo se originó en un entorno donde los datos disponibles, los métodos estadísticos y el poder de cómputo evolucionaron rápida y simultáneamente.

ML has sido usado en:

- Predecir los resultados de las elecciones
- Identificar y filtrar los mensajes no deseados del correo electrónico
- Prever actividad criminal
- Automatice las señales de tráfico según las condiciones de la carretera
- Producir estimaciones financieras de tormentas y desastres naturales
- Examine la rotación de clientes
- Crear aviones de pilotaje automático y automóviles de conducción automática
- Identificar individuos con la capacidad de donar
- Dirigir publicidad a tipos específicos de consumidores

¿Cómo aprenden las máquinas?

> Una máquina aprende si es capaz de llevar la experiencia y utilizarla de manera que su desempeño mejore en experiencias similares en el futuro. (Tom M. Mitchell)

El proceso de aprendizaje puede considerarse comprendido en tres pasos:

- **Entrada de datos:** utiliza la observación, el almacenamiento de memoria y la recuperación para proporcionar una base fáctica para un mayor razonamiento.
- **Abstracción:** implica la traducción de datos en representaciones más amplias.
- **Generalización:** Utiliza datos resumidos para formar una base para la acción.

Pasos para aplicar ML

1. **Recopilación de datos:** si los datos están escritos en papel, grabados en archivos de texto y hojas de cálculo, o almacenados en una base de datos SQL, deberá reunirlos en un formato electrónico adecuado para el análisis. Esta información servirá como el aprendizaje material que usa un algoritmo para generar conocimiento procesable.
2. **Explorar y preparar los datos:** la calidad de cualquier aprendizaje automático proyecto se basa en gran medida en la calidad de los datos que utiliza. Este paso en la máquina el proceso de aprendizaje tiende a requerir una gran cantidad de intervención humana. Un estadística citada a menudo sugiere que el 80 por ciento del esfuerzo en el aprendizaje automático está dedicado a los datos. Gran parte de este tiempo se dedica a aprender más sobre los datos y sus matices durante una práctica llamada exploración de datos.
3. **Formación de un modelo sobre los datos:** cuando los datos han sido preparados para análisis, es probable que tenga una idea de lo que espera aprender de los datos. La tarea específica de aprendizaje automático informará la selección de un algoritmo apropiado, y el algoritmo representará los datos en la forma de un modelo.
4. **Evaluación del rendimiento del modelo:** porque cada modelo de aprendizaje automático resultados en una solución sesgada al problema de aprendizaje, es importante evaluar qué tan bien el algoritmo aprendió de su experiencia. Dependiente en el tipo de modelo utilizado, es posible que pueda evaluar la precisión de el modelo que usa un conjunto de datos de prueba, o puede necesitar desarrollar medidas de rendimiento específico para la aplicación prevista.
5. **Mejora del rendimiento del modelo:** si se necesita un mejor rendimiento, se convierte en necesario para utilizar estrategias más avanzadas para aumentar el rendimiento del modelo. A veces, puede ser necesario cambiar a un tipo diferente de modelo en conjunto. Es posible que necesite complementar sus datos con datos, o realizar trabajos preparatorios adicionales como en el paso dos de este proceso.

Clasificación del ML

- En el **aprendizaje supervisado** (SML), el algoritmo de aprendizaje se presenta con entradas de ejemplo etiquetadas, donde las etiquetas indican el resultado deseado. SML se compone de clasificación, donde el resultado es categórico, y regresión, donde el resultado es numérico.
- En el **aprendizaje no supervisado** (UML), no se proporcionan etiquetas, y el algoritmo de aprendizaje se centra únicamente en la detección de estructura en los datos de entrada no etiquetados.

Aprendizaje supervizado

Regresión

Se utiliza para estimar los valores reales (costo de las viviendas, número de llamadas, ventas totales, etc.) en función de la (s) variable (s) continua (s). Aquí, establecemos la relación entre las variables independientes y dependientes ajustando una mejor línea. Esta línea de mejor ajuste se conoce como línea de regresión y representada por una ecuación lineal $Y = aX + b$.

La mejor forma de entender la regresión lineal es revivir esta experiencia de la infancia. Digamos, le pides a un niño de quinto grado que organice a las personas de su clase aumentando el orden de peso sin pedirles su peso. ¿Qué crees que hará el niño? Es probable que mire (analice visualmente) a la altura y la estructura de las personas y las organice utilizando una combinación de estos parámetros visibles. ¡Esto es una regresión lineal en la vida real! El niño realmente ha descubierto que la altura y la construcción se correlacionan con el peso de una relación, que se parece a la ecuación anterior.

En esta ecuación

- Y es la variable dependiente
- a es la pendiente
- X es la variable independiente
- b es el intercepto

Estos coeficientes a y b se derivan de la minimización de la suma de la diferencia cuadrada de distancia entre los puntos de datos y la línea de regresión.

Mira el ejemplo de abajo. Aquí hemos identificado la mejor línea de ajuste con ecuación lineal $y = 0.2811x+13.9$. Ahora usando esta ecuación, podemos encontrar el peso, sabiendo la altura de una persona.

La regresión lineal es principalmente de dos tipos: regresión lineal simple y regresión lineal múltiple. La regresión lineal simple se caracteriza por una variable independiente. Y, la Regresión Lineal Múltiple (como su nombre indica) se caracteriza por múltiples (más de 1) variables independientes. Al encontrar la mejor línea de ajuste, puede ajustarse a una regresión polinómica o curvilínea. Y estos se conocen como regresión polinómica o curvilínea.

Paso 1: recopilación de datos

```
load("insurance.RData")
```

El archivo `insurance.csv` incluye 1338 beneficiarios actualmente inscritos en el plan de seguro, con características que indican las características del paciente, así como los gastos médicos totales cargados al plan para el año calendario.

- `age`: Este es un número entero que indica la edad del beneficiario principal (excluyendo los mayores de 64 años, ya que generalmente están cubiertos por el gobierno).
- `sex`: Este es el sexo del titular de la póliza, ya sea `male` o `female`.
- `bmi`: Este es el índice de masa corporal (IMC), que proporciona una idea de qué tan alto o bajo de peso tiene una persona en relación con su estatura. El IMC es igual al peso (en kilogramos) dividido por la altura (en metros) al cuadrado. Un IMC ideal está dentro del rango de 18.5 a 24.9.
- `children`: Este es un número entero que indica la cantidad de hijos / dependientes cubiertos por el plan de seguro.
- `smoker`: Esto es `yes` o `no` dependiendo de si el asegurado fuma tabaco regularmente.
- `region`: Este es el lugar de residencia del beneficiario en los EE.UU., Dividido en cuatro regiones geográficas: noreste, sureste, suroeste o noroeste.

Es importante **reflexionar** sobre **cómo estas variables pueden estar relacionadas** con los gastos médicos facturados. Por ejemplo, podríamos esperar que las personas mayores y los fumadores corran un mayor riesgo de grandes gastos médicos.

A diferencia de muchos otros métodos de aprendizaje automático, en el análisis de regresión, el usuario suele especificar las relaciones entre las características en lugar de detectarlas automáticamente.

Paso 2: Explorar y preparar los datos

```
str(insurance)
```

```
## 'data.frame':    1338 obs. of  7 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
```

```
##  $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
##  $ charges : num  16885 1726 4449 21984 3867 ...
```

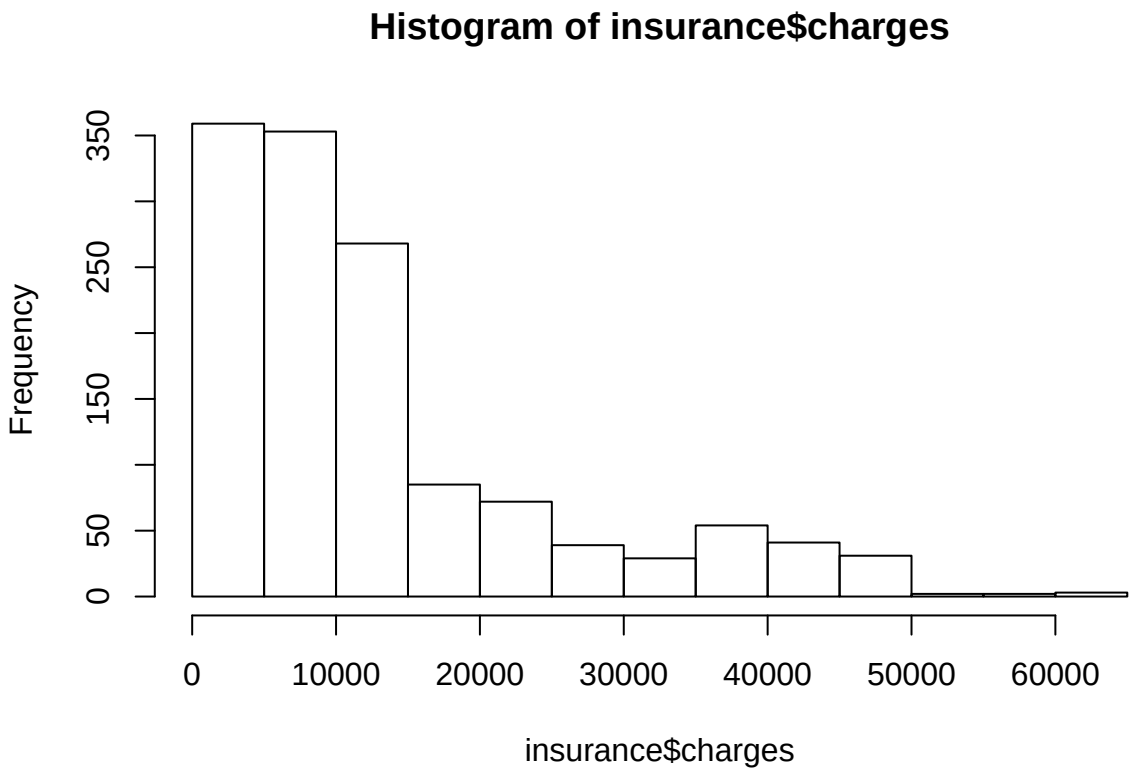
La variable dependiente es `charges`, veamos su distribución:

```
summary(insurance$charges)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1122    4740    9382   13270   16640   63770
```

Como la media es mayor a la mediana, la distribución de los cargos por seguros es sesgada a la derecha. Podemos confirmalo visualmente:

```
hist(insurance$charges)
```



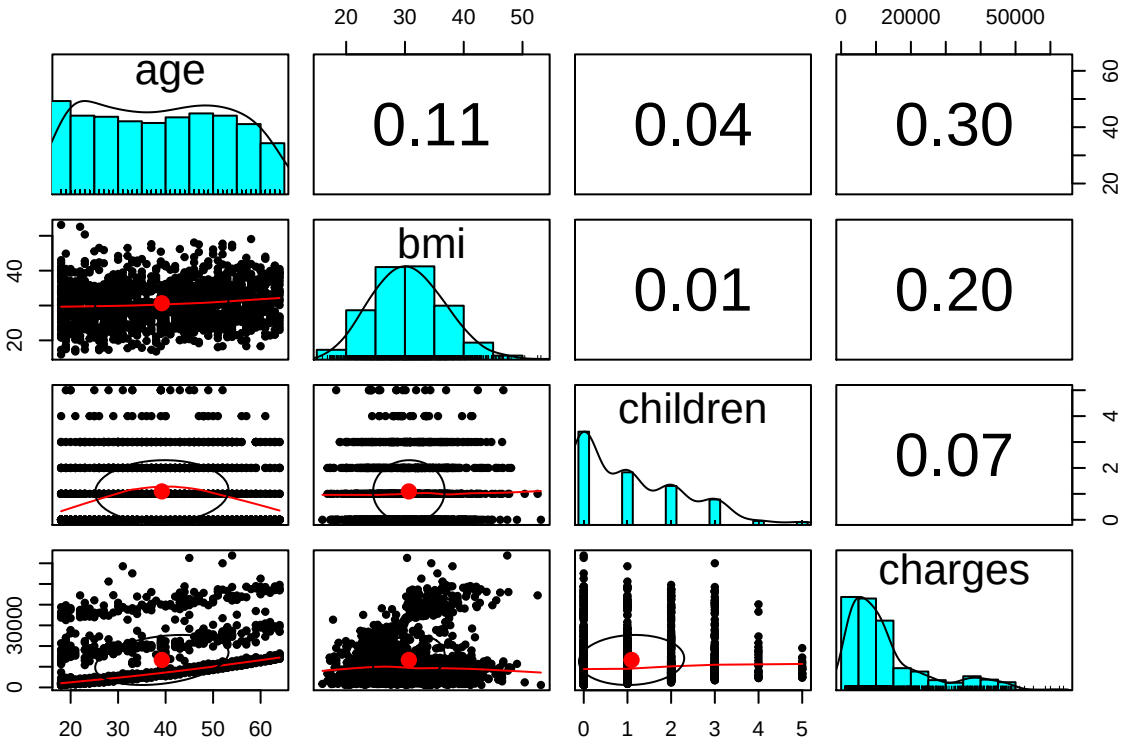
Sabemos que los datos están dividios en 4 regiones, veamos más de cerca su ditribución

```
table(insurance$region)
```

```
##
## northeast northwest southeast southwest
##       324       325       364       325
```

Vemos que los datos se distribuyen casi uniformemente. Veamos ahora la asociación lineal de las variables

```
library(psych)
pairs.panels(insurance[,c("age", "bmi", "children", "charges")])
```



Paso 3: entrenar un modelo en los datos

Ajustemos un modelo de regresión

```
ins_model <- lm(charges ~ age + children + bmi + sex + smoker + region, data = insurance)
# ins_model <- lm(charges ~ ., data = insurance)
ins_model
```

```
##
## Call:
## lm(formula = charges ~ age + children + bmi + sex + smoker +
##     region, data = insurance)
##
## Coefficients:
##     (Intercept)              age         children              bmi
##        -11938.5            256.9            475.5            339.2
##         sexmale        smokeryes  regionnorthwest  regionsoutheast
##          -131.3          23848.5           -353.0          -1035.0
## regionsouthwest
##          -960.1
```

Paso 4: evaluar el rendimiento del modelo

```
summary(ins_model)
```

```
##
## Call:
## lm(formula = charges ~ age + children + bmi + sex + smoker +
##     region, data = insurance)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11304.9  -2848.1   -982.1   1393.9  29992.8
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -11938.5      987.8 -12.086  < 2e-16 ***
## age                256.9       11.9  21.587  < 2e-16 ***
## children           475.5      137.8   3.451 0.000577 ***
## bmi                339.2       28.6  11.860  < 2e-16 ***
## sexmale           -131.3      332.9  -0.394 0.693348
## smokeryes        23848.5      413.1  57.723  < 2e-16 ***
## regionnorthwest   -353.0      476.3  -0.741 0.458769
## regionsoutheast  -1035.0      478.7  -2.162 0.030782 *
## regionsouthwest   -960.0      477.9  -2.009 0.044765 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6062 on 1329 degrees of freedom
## Multiple R-squared:  0.7509, Adjusted R-squared:  0.7494
## F-statistic: 500.8 on 8 and 1329 DF,  p-value: < 2.2e-16
```

- Al ver el resumen de los residuos, concluimos que al menos una de las obervaciones tiene un error de aproximadamente 30000.

Paso 5: mejorando el ajuste

Una de las formas de lograr este objetivo es suavizando el modelo, por ejemplo tomando un polinomio:

```
insurance$age2 <- insurance$age^2
```

También ayuda el discretizar algunas variables. Supongamos que creemos que una variable tiene un efecto dado algún segmento de su ditribución. Podemos crear una dummy con un cierto umbral para capturar este efecto:

```
insurance$bmi30 <- ifelse(insurance$bmi >= 30, 1, 0)
```

Otra forma de logar mejorar el modelo es aumentando interacciones en el modelo.

```
ins_model2 <- lm(charges ~ age + age2 + children + bmi + sex +
bmi30*smoker + region, data = insurance)
summary(ins_model2)
```

```
##
## Call:
## lm(formula = charges ~ age + age2 + children + bmi + sex + bmi30 *
##     smoker + region, data = insurance)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17296.4  -1656.0  -1263.3   -722.1  24160.2
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       134.2509  1362.7511   0.099 0.921539
```

```
## age               -32.6851    59.8242  -0.546 0.584915
## age2                3.7316     0.7463   5.000 6.50e-07 ***
## children          678.5612   105.8831   6.409 2.04e-10 ***
## bmi               120.0196    34.2660   3.503 0.000476 ***
## sexmale          -496.8245   244.3659  -2.033 0.042240 *
## bmi30           -1000.1403   422.8402  -2.365 0.018159 *
## smokeryes       13404.6866   439.9491  30.469  < 2e-16 ***
## regionnorthwest  -279.2038   349.2746  -0.799 0.424212
## regionsoutheast  -828.5467   351.6352  -2.356 0.018604 *
## regionsouthwest -1222.6437   350.5285  -3.488 0.000503 ***
## bmi30:smokeryes 19810.7533   604.6567  32.764  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4445 on 1326 degrees of freedom
## Multiple R-squared:  0.8664, Adjusted R-squared:  0.8653
## F-statistic: 781.7 on 11 and 1326 DF,  p-value: < 2.2e-16
```

Otra forma de mejorar el modelo es usando la función `step`

```
fit1 <- lm(charges ~ ., data = insurance)
fit2 <- lm(charges ~ 1, data = insurance)
step(fit1,direction="backward")
```

```
## Start:  AIC=23280.51
## charges ~ age + sex + bmi + children + smoker + region + age2 +
##     bmi30
##
##            Df  Sum of Sq        RSS   AIC
## - age       1 4.4942e+06 4.7408e+10 23279
## - sex       1 9.1647e+06 4.7413e+10 23279
## - region    3 1.8862e+08 4.7593e+10 23280
## <none>                   4.7404e+10 23280
## - bmi       1 3.9891e+08 4.7803e+10 23290
## - age2      1 4.6048e+08 4.7864e+10 23291
## - children  1 7.0043e+08 4.8104e+10 23298
## - bmi30     1 8.8621e+08 4.8290e+10 23303
## - smoker    1 1.2253e+11 1.6994e+11 24987
##
## Step:  AIC=23278.63
## charges ~ sex + bmi + children + smoker + region + age2 + bmi30
##
##            Df  Sum of Sq        RSS   AIC
## - sex       1 9.0630e+06 4.7417e+10 23277
## - region    3 1.8849e+08 4.7597e+10 23278
## <none>                   4.7408e+10 23279
## - bmi       1 3.9616e+08 4.7805e+10 23288
## - children  1 7.3360e+08 4.8142e+10 23297
## - bmi30     1 8.9822e+08 4.8307e+10 23302
## - age2      1 1.7626e+10 6.5034e+10 23700
## - smoker    1 1.2253e+11 1.6994e+11 24985
##
## Step:  AIC=23276.89
## charges ~ bmi + children + smoker + region + age2 + bmi30
##
```

```
##            Df  Sum of Sq        RSS   AIC
## - region    3 1.8827e+08 4.7606e+10 23276
## <none>                  4.7417e+10 23277
## - bmi       1 3.9448e+08 4.7812e+10 23286
## - children  1 7.3116e+08 4.8149e+10 23295
## - bmi30     1 8.9533e+08 4.8313e+10 23300
## - age2      1 1.7655e+10 6.5073e+10 23698
## - smoker    1 1.2306e+11 1.7048e+11 24987
##
## Step:  AIC=23276.19
## charges ~ bmi + children + smoker + age2 + bmi30
##
##            Df  Sum of Sq        RSS   AIC
## <none>                  4.7606e+10 23276
## - bmi       1 3.3084e+08 4.7937e+10 23284
## - children  1 7.2678e+08 4.8333e+10 23294
## - bmi30     1 9.3668e+08 4.8542e+10 23300
## - age2      1 1.7777e+10 6.5383e+10 23699
## - smoker    1 1.2362e+11 1.7123e+11 24987
```

```
##
## Call:
## lm(formula = charges ~ bmi + children + smoker + age2 + bmi30,
##     data = insurance)
##
## Coefficients:
## (Intercept)          bmi     children    smokeryes         age2
##   -3578.823      135.992      611.678    23828.987        3.261
##       bmi30
##    2788.688
```

```
step(fit2,direction="forward",scope=list(upper=fit1,lower=fit2))
```

```
## Start:  AIC=25160.18
## charges ~ 1
##
##            Df  Sum of Sq        RSS   AIC
## + smoker    1 1.2152e+11 7.4554e+10 23868
## + age2      1 1.7738e+10 1.7834e+11 25035
## + age       1 1.7530e+10 1.7854e+11 25037
## + bmi30     1 7.8063e+09 1.8827e+11 25108
## + bmi       1 7.7134e+09 1.8836e+11 25108
## + children  1 9.0660e+08 1.9517e+11 25156
## + region    3 1.3008e+09 1.9477e+11 25157
## + sex       1 6.4359e+08 1.9543e+11 25158
## <none>                  1.9607e+11 25160
##
## Step:  AIC=23868.38
## charges ~ smoker
##
##            Df  Sum of Sq        RSS   AIC
## + age2      1 2.0260e+10 5.4295e+10 23446
## + age       1 1.9928e+10 5.4626e+10 23454
## + bmi30     1 7.7565e+09 6.6798e+10 23723
## + bmi       1 7.4856e+09 6.7069e+10 23729
```

```
## + children  1 7.5272e+08 7.3802e+10 23857
## <none>                   7.4554e+10 23868
## + sex       1 1.4213e+06 7.4553e+10 23870
## + region    3 1.0752e+08 7.4447e+10 23872
##
## Step:  AIC=23446.1
## charges ~ smoker + age2
##
##            Df  Sum of Sq        RSS   AIC
## + bmi30     1 5622934584 4.8672e+10 23302
## + bmi       1 5026405143 4.9268e+10 23318
## + children  1  776816728 5.3518e+10 23429
## <none>                   5.4295e+10 23446
## + age       1   10202784 5.4284e+10 23448
## + sex       1    2089722 5.4293e+10 23448
## + region    3  131673514 5.4163e+10 23449
##
## Step:  AIC=23301.82
## charges ~ smoker + age2 + bmi30
##
##            Df Sum of Sq        RSS   AIC
## + children  1 735108434 4.7937e+10 23284
## + bmi       1 339172285 4.8333e+10 23294
## <none>                  4.8672e+10 23302
## + age       1  47604629 4.8624e+10 23302
## + sex       1   5103662 4.8667e+10 23304
## + region    3 113242561 4.8558e+10 23305
##
## Step:  AIC=23283.46
## charges ~ smoker + age2 + bmi30 + children
##
##          Df Sum of Sq        RSS   AIC
## + bmi     1 330842015 4.7606e+10 23276
## <none>                4.7937e+10 23284
## + sex     1   7267450 4.7929e+10 23285
## + age     1   1784079 4.7935e+10 23285
## + region  3 124634416 4.7812e+10 23286
##
## Step:  AIC=23276.19
## charges ~ smoker + age2 + bmi30 + children + bmi
##
##          Df Sum of Sq        RSS   AIC
## <none>                4.7606e+10 23276
## + region  3 188269713 4.7417e+10 23277
## + sex     1   8842056 4.7597e+10 23278
## + age     1   4268540 4.7601e+10 23278
##
## Call:
## lm(formula = charges ~ smoker + age2 + bmi30 + children + bmi,
##     data = insurance)
##
## Coefficients:
## (Intercept)    smokeryes         age2        bmi30     children
```

```
##   -3578.823    23828.987        3.261     2788.688      611.678
##         bmi
##     135.992
```

Análisis discriminante

El análisis discriminante lineal (LDA) y el discriminante lineal de Fisher relacionado son métodos utilizados en estadística, reconocimiento de patrones y aprendizaje automático para encontrar una combinación lineal de características que caracteriza o separa dos o más clases de objetos o eventos. La combinación resultante se puede usar como un clasificador lineal o, más comúnmente, para la reducción de dimensionalidad antes de la clasificación posterior.

Considere un conjunto de observaciones x (también llamadas características, atributos, variables o medidas) para cada muestra de un objeto o evento con una clase conocida $y \in \{0, 1\}$. Este conjunto de muestras se llama conjunto de entrenamiento. El problema de clasificación es encontrar un buen predictor para la clase y de cualquier muestra de la misma distribución (no necesariamente del conjunto de entrenamiento), dado solo una observación x.

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.

Posibles aplicaciones

- Predicción de bancarrota: en la predicción de bancarrota basada en razones contables y otras variables financieras, el análisis discriminante lineal fue el primer método estadístico aplicado para explicar sistemáticamente qué empresas entraron en bancarrota vs. sobrevivieron.
- Comercialización: en marketing, el análisis discriminante solía utilizarse para determinar los factores que distinguen diferentes tipos de clientes y/o productos sobre la base de encuestas u otras formas de datos recopilados.
- Estudios biomédicos: la principal aplicación del análisis discriminante en medicina es la evaluación del estado de gravedad de un paciente y el pronóstico del desenlace de la enfermedad. Por ejemplo, durante el análisis retrospectivo, los pacientes se dividen en grupos según la gravedad de la enfermedad, forma leve, moderada y grave. Luego, se estudian los resultados de los análisis clínicos y de laboratorio para revelar las variables que son estadísticamente diferentes en los grupos estudiados. Usando estas variables, se construyen funciones discriminantes que ayudan a clasificar objetivamente la enfermedad en un futuro paciente en una forma leve, moderada o severa.

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 (por ejemplo, bancarrota-no bancarrota). 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

- La *variable dependiente* (grupos) debe ser 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.

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:

$$D_i = a + W_1 X_{1,i} + W_2 X_{2,i} + \ldots + W_n X_{n,i}$$

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 $X1, \ldots, Xn$ 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.

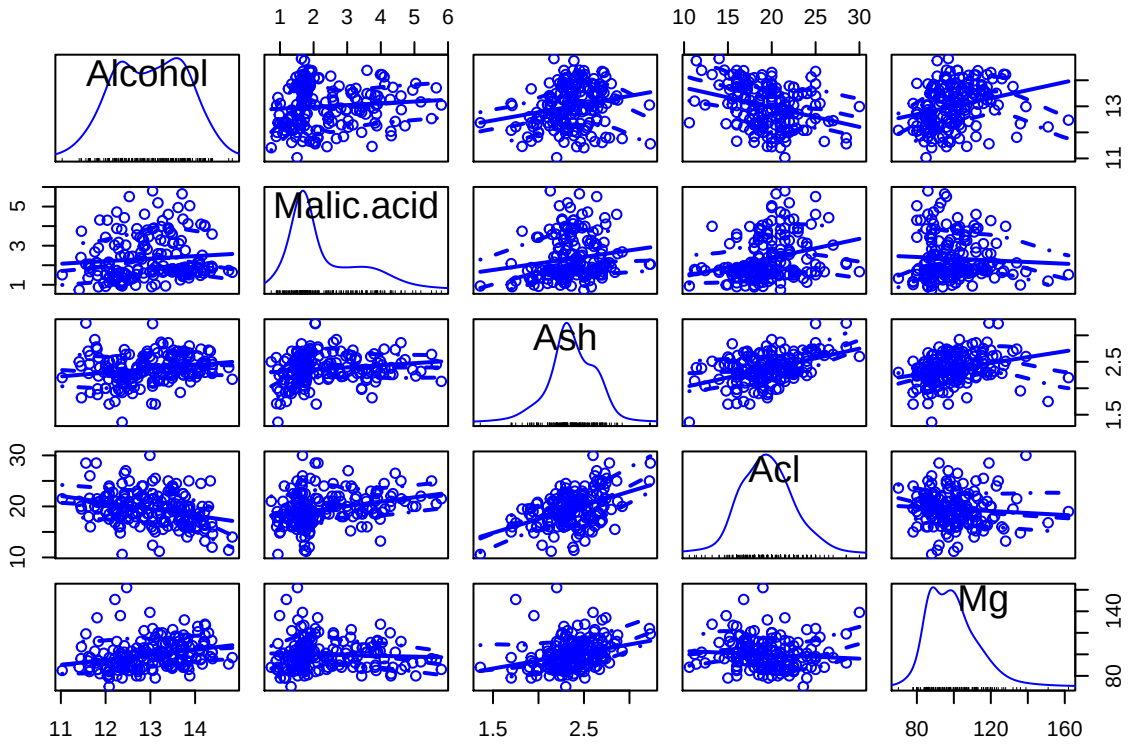
Ejemplo 1: clasificación de vinos

En este primer caso de estudio, el conjunto de datos del vino, tenemos 13 concentraciones químicas que describen muestras de vino de tres cultivos.

```
library(car)
# install.packages('rattle')
uu <- "https://gist.githubusercontent.com/tijptjik/9408623/raw/b237fa5848349a14a14e5d4107dc7897c21951f5/
wine <- read.csv(url(uu))
head(wine)
```

```
##   Wine Alcohol Malic.acid  Ash  Acl  Mg Phenols Flavanoids
## 1    1   14.23       1.71 2.43 15.6 127    2.80       3.06
## 2    1   13.20       1.78 2.14 11.2 100    2.65       2.76
## 3    1   13.16       2.36 2.67 18.6 101    2.80       3.24
## 4    1   14.37       1.95 2.50 16.8 113    3.85       3.49
## 5    1   13.24       2.59 2.87 21.0 118    2.80       2.69
## 6    1   14.20       1.76 2.45 15.2 112    3.27       3.39
##   Nonflavanoid.phenols Proanth Color.int  Hue   OD Proline
## 1                 0.28    2.29      5.64 1.04 3.92    1065
## 2                 0.26    1.28      4.38 1.05 3.40    1050
## 3                 0.30    2.81      5.68 1.03 3.17    1185
## 4                 0.24    2.18      7.80 0.86 3.45    1480
## 5                 0.39    1.82      4.32 1.04 2.93     735
## 6                 0.34    1.97      6.75 1.05 2.85    1450
```

```
scatterplotMatrix(wine[2:6])
```



El propósito del análisis discriminante lineal (LDA) en este ejemplo es encontrar las combinaciones lineales de las variables originales (las 13 concentraciones químicas aquí) que proporcionan la mejor separación posible entre los grupos (variedades de vino aquí) en nuestro conjunto de datos. El análisis discriminante lineal también se conoce como "análisis discriminante canónico", o simplemente "análisis discriminante".

Supuestos:

Homogeneidad de varianzas multivariante

```
library(vegan)
# seleccionamos las variables ambientales a analizar
env.pars2 <- as.matrix(wine[, 2:14])
# verificamos la homogeneidad multivariada de las matrices de covarianza intra-grupo
env.pars2.d1 <- dist(env.pars2)
env.MHV <- betadisper(env.pars2.d1, wine$Wine)
anova(env.MHV)
```

```
## Analysis of Variance Table
##
## Response: Distances
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## Groups      2  190082   95041  8.3286 0.0003507 ***
## Residuals 175 1997003   11411
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
permutest(env.MHV)
```

```
##
## 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      2  190082   95041 8.3286    999  0.001 ***
## Residuals 175 1997003   11411
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Conclusión: rechazo la hipótesis nula de homogeneidad intra-grupo. Se podría hacer transformaciones logarítmicas para enfrentar este asunto.

Normalidad multivariante

```
library(mvnormtest)
mshapiro.test(t(env.pars2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Z
## W = 0.83696, p-value = 7.846e-13
```

Rechazamos la Ho de normalidad multivariante

Multicolinealidad

```r
as.dist(cor(env.pars2))
```

```
##                            Alcohol   Malic.acid          Ash          Acl
## Malic.acid            0.094396941
## Ash                   0.211544596  0.164045470
## Acl                  -0.310235137  0.288500403  0.443367187
## Mg                    0.270798226 -0.054575096  0.286586691 -0.083333089
## Phenols               0.289101123 -0.335166997  0.128979538 -0.321113317
## Flavanoids            0.236814928 -0.411006588  0.115077279 -0.351369860
## Nonflavanoid.phenols -0.155929467  0.292977133  0.186230446  0.361921719
## Proanth               0.136697912 -0.220746187  0.009651935 -0.197326836
## Color.int             0.546364195  0.248985344  0.258887259  0.018731981
## Hue                  -0.071747197 -0.561295689 -0.074666889 -0.273955223
## OD                    0.072343187 -0.368710428  0.003911231 -0.276768549
## Proline               0.643720037 -0.192010565  0.223626264 -0.440596931
##                              Mg      Phenols   Flavanoids
## Malic.acid
## Ash
## Acl
## Mg
## Phenols               0.214401235
## Flavanoids            0.195783770  0.864563500
## Nonflavanoid.phenols -0.256294049 -0.449935301 -0.537899612
## Proanth               0.236440610  0.612413084  0.652691769
## Color.int             0.199950006 -0.055136418 -0.172379398
## Hue                   0.055398196  0.433681335  0.543478566
## OD                    0.066003936  0.699949365  0.787193902
## Proline               0.393350849  0.498114880  0.494193127
##                      Nonflavanoid.phenols      Proanth    Color.int
## Malic.acid
## Ash
## Acl
## Mg
## Phenols
## Flavanoids
## Nonflavanoid.phenols
## Proanth                      -0.365845099
## Color.int                     0.139057013 -0.025249931
## Hue                          -0.262639631  0.295544253 -0.521813193
## OD                           -0.503269596  0.519067096 -0.428814942
## Proline                      -0.311385188  0.330416700  0.316100113
##                               Hue           OD
## Malic.acid
## Ash
## Acl
## Mg
## Phenols
## Flavanoids
## Nonflavanoid.phenols
## Proanth
## Color.int
## Hue
## OD                    0.565468293
## Proline               0.236183447  0.312761075
```

```r
library(MASS)
wine.lda <- lda(Wine ~ ., data=wine)
wine.lda
```

```
## Call:
## lda(Wine ~ ., data = wine)
##
## Prior probabilities of groups:
##         1         2         3
## 0.3314607 0.3988764 0.2696629
##
## Group means:
##    Alcohol Malic.acid      Ash      Acl       Mg  Phenols Flavanoids
## 1 13.74475   2.010678 2.455593 17.03729 106.3390 2.840169  2.9823729
## 2 12.27873   1.932676 2.244789 20.23803  94.5493 2.258873  2.0808451
## 3 13.15375   3.333750 2.437083 21.41667  99.3125 1.678750  0.7814583
##   Nonflavanoid.phenols  Proanth Color.int       Hue       OD   Proline
## 1             0.290000 1.899322  5.528305 1.0620339 3.157797 1115.7119
## 2             0.363662 1.630282  3.086620 1.0562817 2.785352  519.5070
## 3             0.447500 1.153542  7.396250 0.6827083 1.683542  629.8958
##
## Coefficients of linear discriminants:
##                                LD1           LD2
## Alcohol              -0.403399781  0.8717930699
## Malic.acid            0.165254596  0.3053797325
## Ash                  -0.369075256  2.3458497486
## Acl                   0.154797889 -0.1463807654
## Mg                   -0.002163496 -0.0004627565
## Phenols               0.618052068 -0.0322128171
## Flavanoids           -1.661191235 -0.4919980543
## Nonflavanoid.phenols -1.495818440 -1.6309537953
## Proanth               0.134092628 -0.3070875776
## Color.int             0.355055710  0.2532306865
## Hue                  -0.818036073 -1.5156344987
## OD                   -1.157559376  0.0511839665
## Proline              -0.002691206  0.0028529846
##
## Proportion of trace:
##    LD1    LD2
## 0.6875 0.3125
```

Esto significa que la primera función discriminante es una combinación lineal de las variables: $-0.403 * Alcohol + 0.165 * Malic \cdots - 0.003 * Proline$. Por conveniencia, el valor de cada función discriminante (por ejemplo, la primera función discriminante) se escala de modo que su valor medio sea cero y su varianza sea uno.

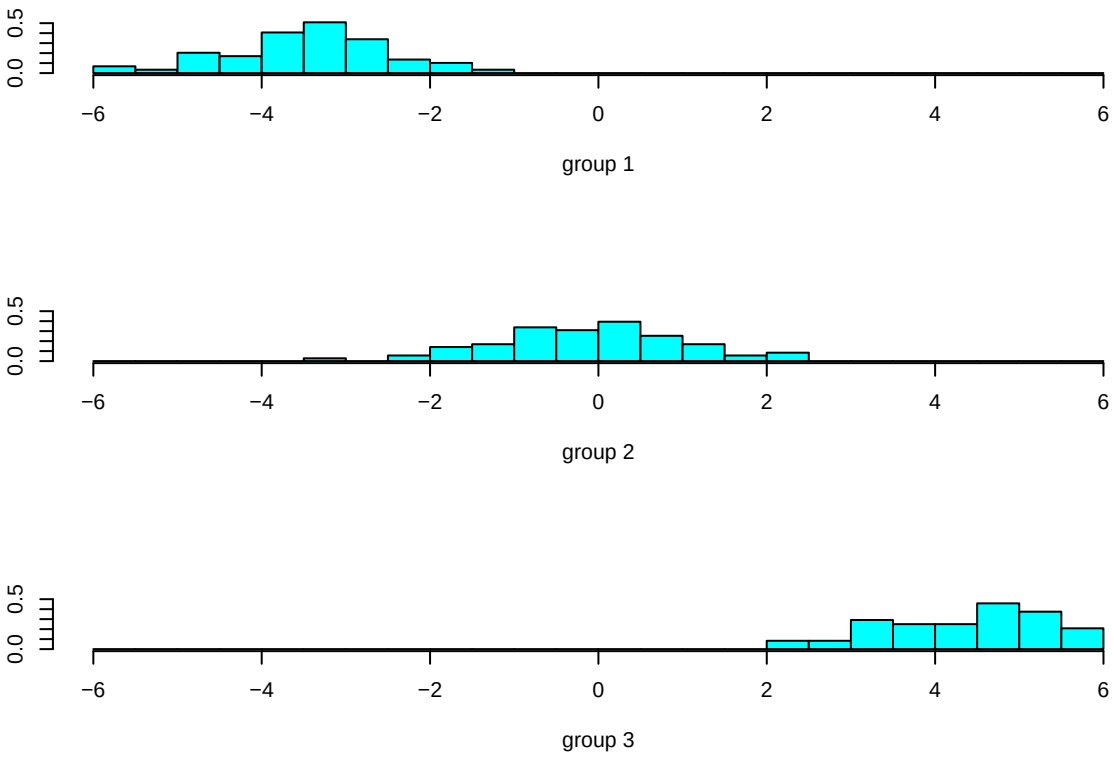
La "proporción de traza" que se imprime cuando escribe "wine.lda" (la variable devuelta por la función `lda()`) es la separación porcentual lograda por cada función discriminante. Por ejemplo, para los datos del vino obtenemos los mismos valores que acabamos de calcular (68.75% y 31.25%).

Histrogramas de resultado

Una buena forma de mostrar los resultados de un análisis discriminante lineal (LDA) es hacer un histograma apilado de los valores de la función discriminante para las muestras de diferentes grupos (diferentes variedades de vino en nuestro ejemplo).

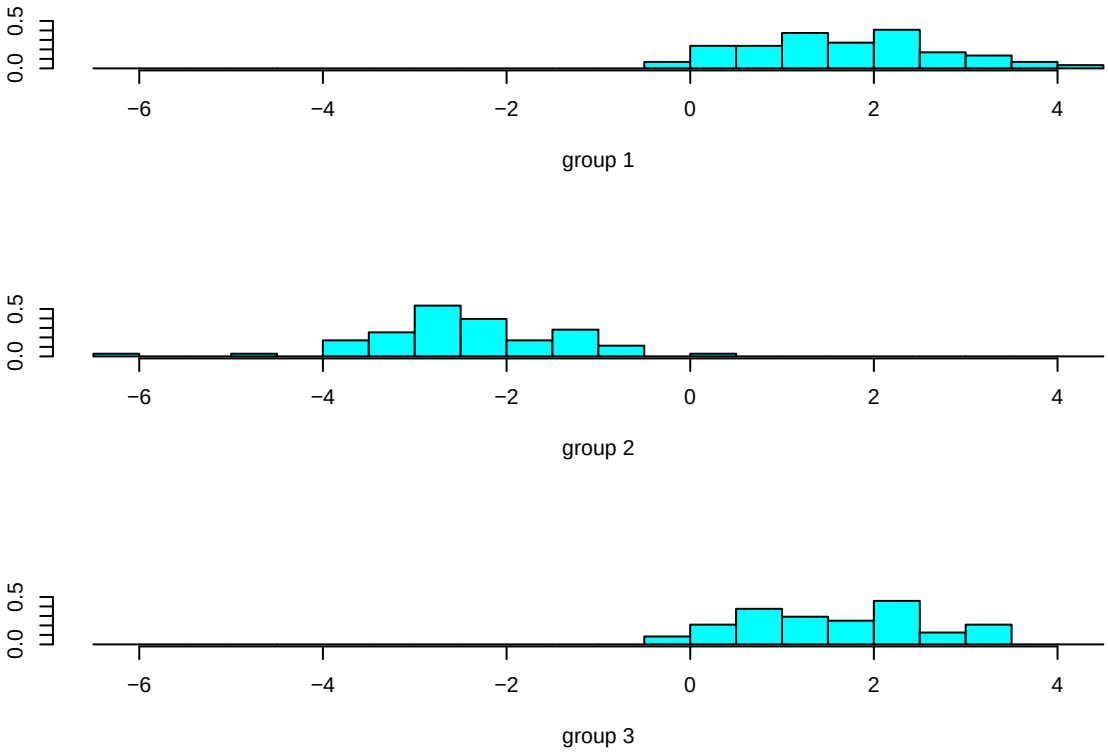
Podemos hacer esto usando la función `ldahist()` en R. Por ejemplo, para hacer un histograma apilado de los valores de la primera función discriminante para muestras de vino de los tres diferentes cultivares de vino, escribimos:

```
wine.lda.values <- predict(wine.lda)
ldahist(data = wine.lda.values$x[,1], g=wine$Wine)
```



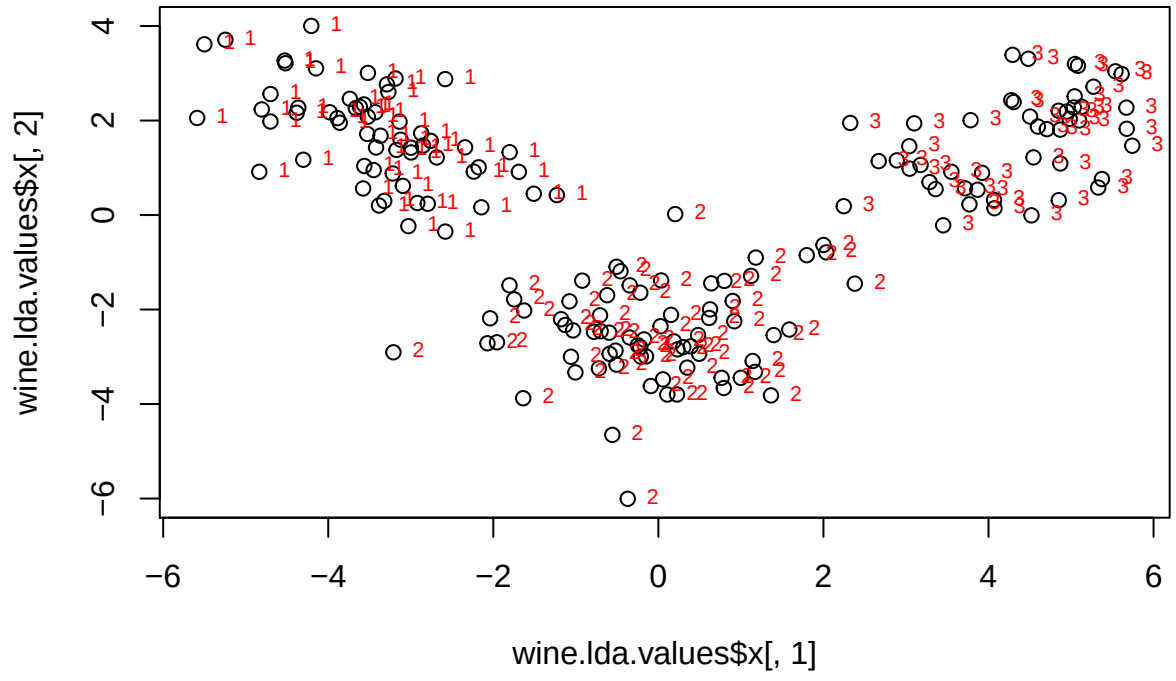
usando la segunda función discriminante:

```
ldahist(data = wine.lda.values$x[,2], g=wine$Wine)
```



Gráficos de las funciones discriminantes

```
plot(wine.lda.values$x[,1],wine.lda.values$x[,2]) # se realiza el grafico
text(wine.lda.values$x[,1],wine.lda.values$x[,2],wine$Wine,cex=0.7,pos=4,col="red") # agregamos etiique
```




```
spe.class <- predict(wine.lda)$class
(spe.table <-table(wine$Wine, spe.class))
```

```
##    spe.class
```

```
##      1  2  3
##   1 59  0  0
##   2  0 71  0
##   3  0  0 48
```

Ejemplo 2: Admisiones

El conjunto de datos proporciona datos de admisión para los solicitantes a las escuelas de posgrado en los negocios. El objetivo es usar los puntajes de GPA y GMAT para predecir la probabilidad de admisión (admitir, no admitir y límite).

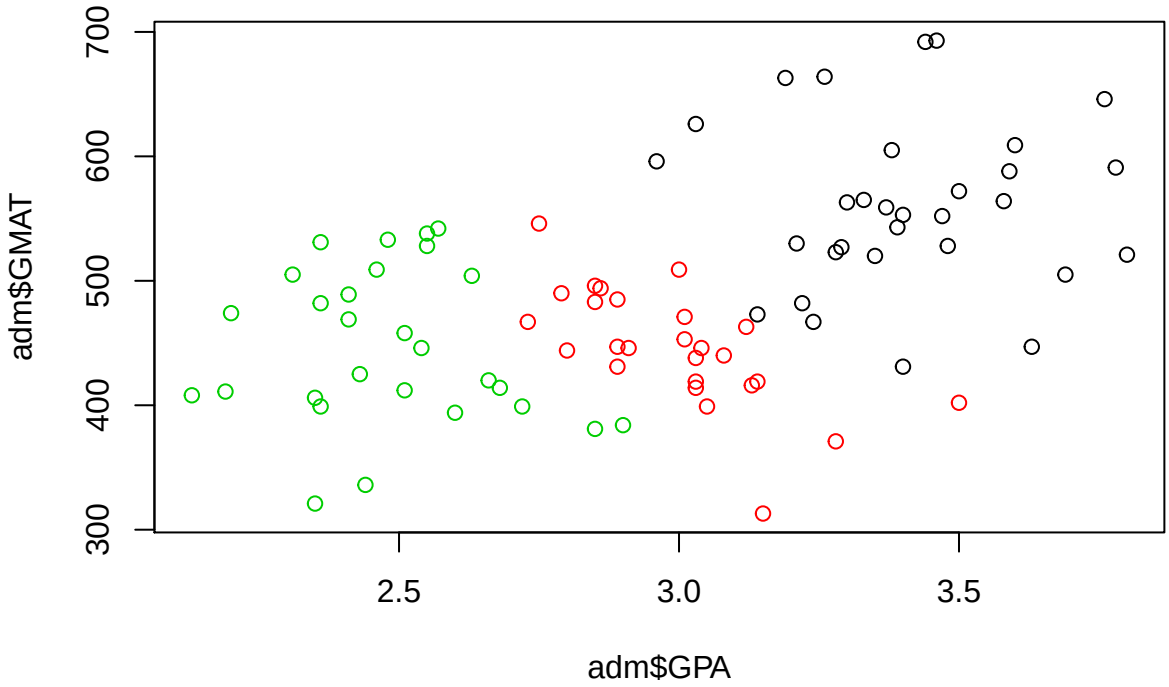
```
url <- 'http://www.biz.uiowa.edu/faculty/jledolter/DataMining/admission.csv'
admit <- read.csv(url)

head(admit)
```

```
##    GPA GMAT    De
## 1 2.96  596 admit
## 2 3.14  473 admit
## 3 3.22  482 admit
## 4 3.29  527 admit
## 5 3.69  505 admit
## 6 3.46  693 admit
```

Realizamos un gráfico de los datos:

```
adm <- data.frame(admit)
plot(adm$GPA,adm$GMAT,col=adm$De)
```



Supuestos:

Homogeneidad de varianzas multivariante

```r
library(vegan)
# seleccionamos las variables ambientales a analizar
env.pars2 <- as.matrix(adm[, 1:2])
# verificamos la homogeneidad multivariada de las matrices de covarianza intra-grupo
env.pars2.d1 <- dist(env.pars2)
env.MHV <- betadisper(env.pars2.d1, adm$De)
anova(env.MHV)
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df Sum Sq Mean Sq F value  Pr(>F)
## Groups     2   6224  3112.0  2.4009 0.09698 .
## Residuals 82 106285  1296.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
permutest(env.MHV)
```

```
##
## 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     2   6224  3112.0 2.4009    999  0.084 .
## Residuals 82 106285  1296.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Conclusión: no rechazo la hipótesis nula de homogeneidad intra-grupo.

Normalidad multivariante

```r
library(mvnormtest)
mshapiro.test(t(env.pars2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Z
## W = 0.98854, p-value = 0.6623
```

No rechazamos la Ho de normalidad multivariante

Multicolinealidad

```r
as.dist(cor(env.pars2))
```

```
##            GPA
## GMAT 0.4606332
```

```r
library(MASS)
m1 <- lda(De~.,adm)
m1
```

```
## Call:
## lda(De ~ ., data = adm)
```

```
##
## Prior probabilities of groups:
##     admit    border  notadmit
## 0.3647059 0.3058824 0.3294118
##
## Group means:
##               GPA     GMAT
## admit    3.403871 561.2258
## border   2.992692 446.2308
## notadmit 2.482500 447.0714
##
## Coefficients of linear discriminants:
##              LD1         LD2
## GPA  5.008766354  1.87668220
## GMAT 0.008568593 -0.01445106
##
## Proportion of trace:
##    LD1    LD2
## 0.9673 0.0327
```

Comenta los resultados.

Realizamos una predicción:

```
predict(m1,newdata=data.frame(GPA=3.21,GMAT=497))
```

```
## $class
## [1] admit
## Levels: admit border notadmit
##
## $posterior
##       admit    border     notadmit
## 1 0.5180421 0.4816015 0.0003563717
##
## $x
##        LD1      LD2
## 1 1.252409 0.318194
```

Análisis discrimante 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.

```
m2 <- qda(De~.,adm)
m2
```

```
## Call:
## qda(De ~ ., data = adm)
##
## Prior probabilities of groups:
##     admit    border  notadmit
## 0.3647059 0.3058824 0.3294118
##
## Group means:
##               GPA     GMAT
## admit    3.403871 561.2258
## border   2.992692 446.2308
## notadmit 2.482500 447.0714
```

Realizamos la predicción

```
predict(m2,newdata=data.frame(GPA=3.21,GMAT=497))
```

```
## $class
## [1] admit
## Levels: admit border notadmit
##
## $posterior
##       admit    border     notadmit
## 1 0.9226763 0.0768693 0.0004544468
```

¿Qué modelo es el mejor?

Para responder a esta pregunta, evaluamos el análisis discriminante lineal seleccionando aleatoriamente 60 de 85 estudiantes, estimando los parámetros en los datos de entrenamiento y clasificando a los 25 estudiantes restantes de la muestra retenida. Repetimos esto 100 veces

```
n <- 85
nt <- 60
neval <-n-nt
rep <- 100

### LDA
set.seed(123456789)
errlin <- dim(rep)
for (k in 1:rep) {
train <- sample(1:n,nt)
## linear discriminant analysis
m1 <- lda(De~.,adm[train,])
predict(m1,adm[-train,])$class
tablin <- table(adm$De[-train],predict(m1,adm[-train,])$class)
errlin[k] <- (neval-sum(diag(tablin)))/neval
}
merrlin <- mean(errlin) #media del error lineal
merrlin
```

```
## [1] 0.102
```

Ahora en el QDA:

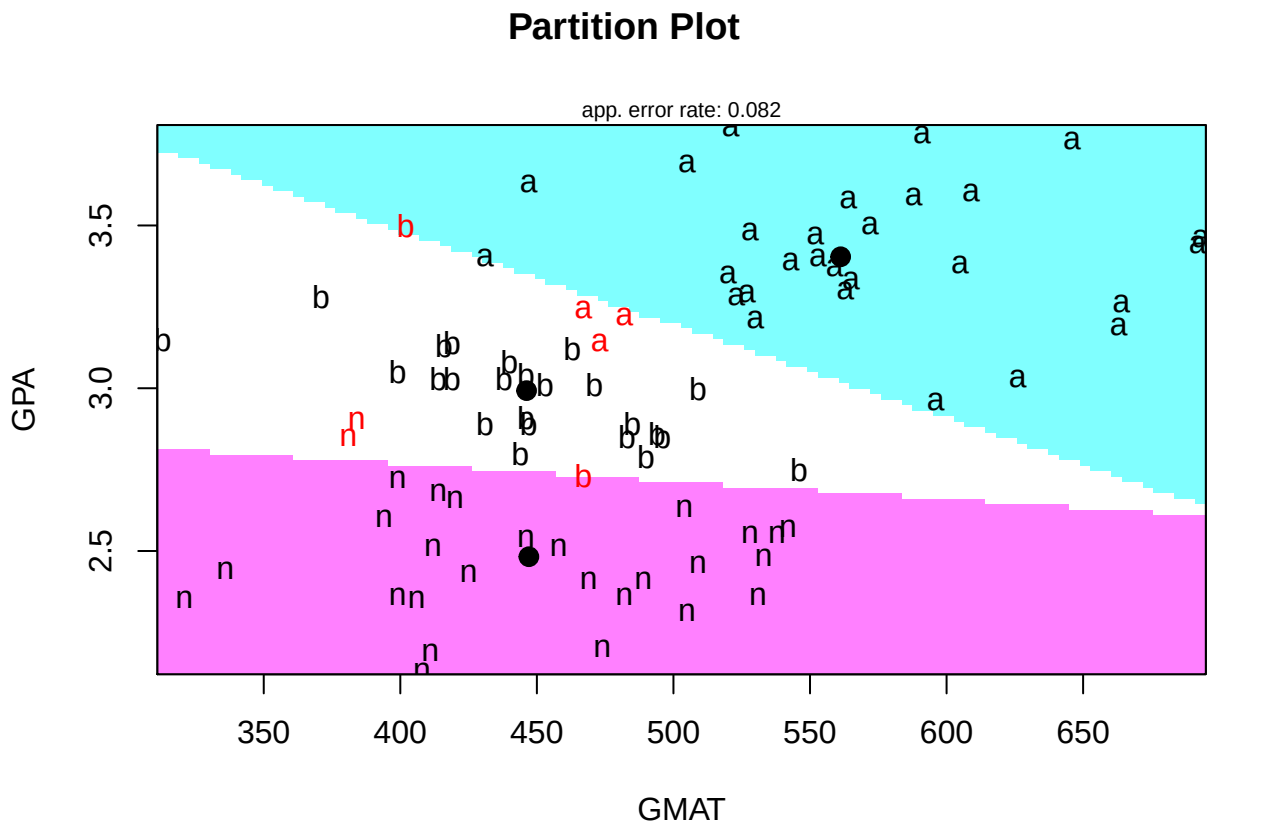
```
### QDA
set.seed(123456789)
errqda <- dim(rep)
for (k in 1:rep) {
train <- sample(1:n,nt)
## quadratic discriminant analysis
m1 <- qda(De~.,adm[train,])
predict(m1,adm[-train,])$class
tablin <- table(adm$De[-train],predict(m1,adm[-train,])$class)
errqda[k] <- (neval-sum(diag(tablin)))/neval
}
merrqda <- mean(errlin)
merrqda
```

```
## [1] 0.102
```

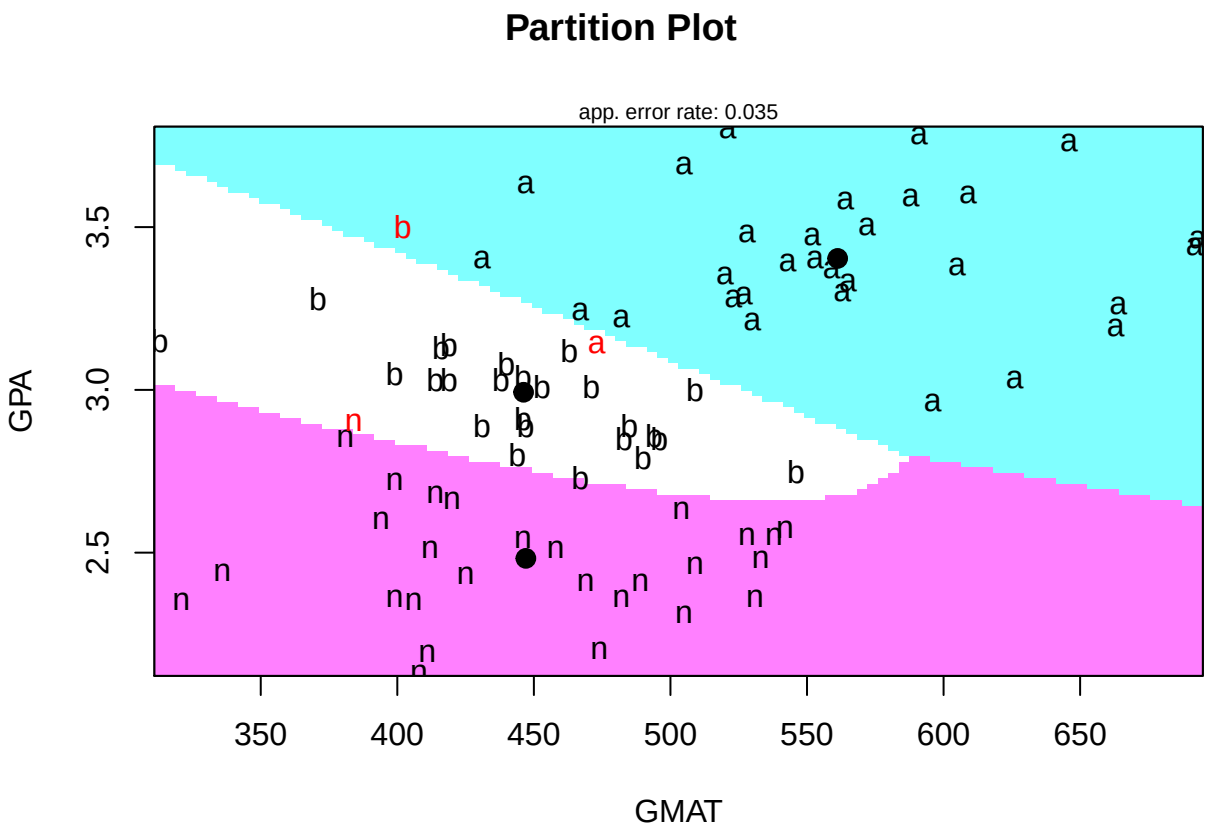
Logramos una tasa de clasificación errónea del 10.2% en ambos casos. R también nos da algunas herramientas

de visualización. Por ejemplo en la librería `klaR`:

```
# Gráficos exploratorios para LDA or QDA
#install.packages('klaR')
library(klaR)
partimat(De~.,data=adm,method="lda")
```




```
partimat(De~.,data=adm,method="qda")
```



Ejemplo 3: Score de crédito de un banco alemán

El conjunto de datos de crédito alemán se obtuvo del Repositorio de aprendizaje automático UCI. El conjunto de datos, que contiene atributos y resultados sobre 1000 solicitudes de préstamo, fue proporcionado en 1994 por el Profesor Dr. Hans Hofmann del Institut fuer Statistik und Oekonometrie de la Universidad de Hamburgo. Ha servido como un importante conjunto de datos de prueba para varios algoritmos de puntuación de crédito. Una descripción de las variables se da en `germancreditDescription.docx` de `DataLectures`. Comenzamos cargando los datos:

```
## read data
credit <- read.csv("http://www.biz.uiowa.edu/faculty/jledolter/DataMining/germancredit.csv")
head(credit,2) # Mira la codificación en el lugar indicado
```

```
##   Default checkingstatus1 duration history purpose amount savings employ
## 1       0             A11        6     A34     A43   1169     A65    A75
## 2       1             A12       48     A32     A43   5951     A61    A73
##   installment status others residence property age otherplans housing
## 1           4    A93   A101         4     A121  67       A143    A152
## 2           2    A92   A101         2     A121  22       A143    A152
##   cards  job liable tele foreign
## 1     2 A173      1 A192    A201
## 2     1 A173      1 A191    A201
```

Como se puede ver, solo las variables: duración, cantidad, plazos y edad son numéricas. Con los restantes (indicadores) los supuestos de una distribución normal serían, en el mejor de los casos, débiles; por lo tanto, estas variables no se consideran aquí.

```r
cred1 <- credit[, c("Default","duration","amount","installment","age")]
head(cred1)
```

```
##   Default duration amount installment age
## 1       0        6   1169           4  67
## 2       1       48   5951           2  22
## 3       0       12   2096           2  49
## 4       0       42   7882           2  45
## 5       1       24   4870           3  53
## 6       0       36   9055           2  35
```

```r
summary(cred1)
```

```
##     Default       duration        amount       installment
##  Min.   :0.0   Min.   : 4.0   Min.   :  250   Min.   :1.000
##  1st Qu.:0.0   1st Qu.:12.0   1st Qu.: 1366   1st Qu.:2.000
##  Median :0.0   Median :18.0   Median : 2320   Median :3.000
##  Mean   :0.3   Mean   :20.9   Mean   : 3271   Mean   :2.973
##  3rd Qu.:1.0   3rd Qu.:24.0   3rd Qu.: 3972   3rd Qu.:4.000
##  Max.   :1.0   Max.   :72.0   Max.   :18424   Max.   :4.000
##       age
##  Min.   :19.00
##  1st Qu.:27.00
##  Median :33.00
##  Mean   :35.55
##  3rd Qu.:42.00
##  Max.   :75.00
```

Transformemos los datos en un data.frame

```r
cred1 <- data.frame(cred1)
```

- Realiza las pruebas de los supuestos y comenta los resultados
- Estima y compara lds con qda
- Estima la matriz de confusión
- ¿Usarías este modelo para una aplicación real?

Referencias