Algunas técnicas básicas para entender los datos del fenómeno a estudiar

Juan A. Botía Blaya
juanbot@um.es

October 21, 2009

1 Introducción

Esta primera práctica la vamos a dedicar a aprender cómo familiarizarnos con conjuntos de datos. La asignatura trata de cómo analizar de manera inteligente (con técnicas distintas a las que encontramos en la estadística convencional) conjuntos de datos que responden a muestras (i.e. observaciones) de un mismo fenómeno. En esta primera práctica, veremos cómo con simples herramientas de visualización de datos y proceso estadístico, podemos obtener, de manera sencilla, informes basados en estadística descriptiva, que nos ayudarán a entender mejor el fenómeno que estamos estudiando.

Este tipo de informe no es el objetivo último del análisis inteligente de datos. Debemos verlo como el primer paso. Como una manera de tener más información para entender el problema de manera inicial y poder atacarlo con las mayores garantías posibles.

2 El problema

La flor que aparece a la izquierda de la figura 1 es una Iris, del tipo siberiano[1]. Supongamos que queremos estudiar esta flor, a partir de una plantación de tres variedades de la misma, Setosa, Versicolor y Virgínica. Para ello se nos ha facilitado una serie de tuplas de datos de cada flor disponibles en un invernadero que usaremos como datos fuente. Si denotamos con

$$D = \{(\bar{x}, y) | \bar{x} \in R^4\},$$

en donde las dos primeras características de $\bar{x}$ se refieren a la longitud y anchura del pétalo, y las otras dos a la longitud y anchura del sépalo, junto con que y se refiere al tipo de flor, esa es toda la información que nos dan.

El problema genérico se trata de responder a la pregunta de si es posible distinguir un tipo de Iris de las otras dos, simplemente mirando a las dimensiones de pétalo y sépalo correspondientes. O formulado de otra forma, si cada una de estas tres variedades se diferencia de manera significativa del resto por sus pétalos y sépalos.

3 Algunas estadísticas descriptivas sobre el cjto. de datos

Lo primero que debemos tener claro es que estamos ante un problema de clasificación. Es decir, debemos generar una hipótesis h, a partir de los datos D, tal que esta nos permita clasificar nuevas observaciones $\bar{x}$ como pertenecientes a uno de los tres tipos de Iris conocidas. La denominamos hipótesis para hacer énfasis en el hecho de que no nos referimos a un modelo concreto, ya sea de red neuronal o árbol de decisión, por poner dos ejemplos de sobra conocidos. De momento no nos interesa saber qué tipo concreto tendrá el modelo.

[1] http://home.att.net/~nnthom266/1999/iris.htm

Figure 1: Detalle de una Iris siberiana (izquierda) y pétalo y sépalo de una flor (derecha).

Al ser un problema de clasificación, lo primero que deberíamos hacer es preguntarnos por la proporción de ejemplares de cada clase que tenemos. ¿Cómo hacer esto de manera sencilla?

Aquí es donde entra R. Desde la línea de comandos de nuestro puesto, hacemos

```
host> R
```

y obtendremos un mensaje de saludo parecido a este, dependiendo de la versión concreta que estemos utilizando

```
R : Copyright 2006, The R Foundation for Statistical Computing
Version 2.3.1 (2006-06-01)
ISBN 3-900051-07-0

R es un software libre y viene sin GARANTIA ALGUNA.
Usted puede redistribuirlo bajo ciertas circunstancias.
Escriba 'license()' o 'licence()' para detalles de distribucion.

R es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para obtener mas informacion y
'citation()' para saber como citar R o paquetes de R en publicaciones.

Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

>
```

Asumimos ahora que el conjunto de datos (el fichero de texto `iris.data`) está accesible y que las tuplas del mismo tienen la forma[2]

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-versicolor
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-virginica
```

[2]Mirar en http://www.ics.uci.edu/~mlearn/databases/ donde podreis encontrar este ejemplo y otros típicos de problemas de aprendizaje.

La primera tupla corresponde a los valores 5.1, 3.5, 1.4 y 0.2 para $\bar{x}$ e `Iris-setosa` para la etiqueta de clase y.

El primer paso es intentar que R lea ese fichero y lo haga disponible en memoria principal en forma de variable para que lo podamos manipular desde ahí. Si hacemos (suponiendo que el conjunto de datos esté en el directorio desde el que hemos invocado a R),

```
> iris <- read.table("iris.data")
```

ahora, en la variable `iris` tenemos almacenado el cjto. de datos. Para ver el contenido de la variable hacemos

```
> iris
```

Y obtendremos algo como esto

```
                                V1
1       5.1,3.5,1.4,0.2,Iris-setosa
2       4.9,3.0,1.4,0.2,Iris-setosa
3       4.7,3.2,1.3,0.2,Iris-setosa
4       4.6,3.1,1.5,0.2,Iris-setosa
5       5.0,3.6,1.4,0.2,Iris-setosa
...
```

que nos muestra todo el conjunto de datos tal y como está contenido en la variable `iris`. Obsérvese que en la primera línea aparece una etiqueta `V1`, y como primera columna una secuencia de números. La etiqueta hace referencia al nombre de la columna almacenada, lo cual quiere decir que no se han separado en características diferentes los cinco valores que debe haber por tupla. Esto es así ya que el separador que asume por defecto la función `read.table` es el espacio. En este caso, es una coma lo que distingue unos valores de otros. Por lo tanto, si echamos mano del manual de referencia de R, veremos que en al entrada para `read.table` aparece un modificador `sep`, que aplicamos tal que así

```
> iris <- read.table("iris.data",sep=",")
```

con lo que si ahora volvemos a visualizar las primeras cinco filas del data frame, tendremos

```
> iris[1:5,]
```

obtenemos

```
    V1  V2  V3  V4          V5
1   5.1 3.5 1.4 0.2 Iris-setosa
2   4.9 3.0 1.4 0.2 Iris-setosa
3   4.7 3.2 1.3 0.2 Iris-setosa
4   4.6 3.1 1.5 0.2 Iris-setosa
5   5.0 3.6 1.4 0.2 Iris-setosa
...
```

con lo que comprobamos que se han separado correctamente los valores en cinco columnas. Para trabajar más cómodamente con el conjunto, vamos a ponerle un nombre a las columnas. Así que el comando final para leer los datos será

```
iris <- read.table("iris.txt", header=F, sep=",",
        col.names=c('longitud sepalo', 'anchura sepalo',
        'longitud petalo', 'anchura petalo', 'clase'))
```

Simplemente le decimos a R que los nombres de las columnas de los datos que va a leer son los que le indicamos. Con `header=F` le decimos que los datos no vienen con encabezamiento. Con este comando, ahora tenemos

```
> iris[1:5,]
  longitud.sepalo anchura.sepalo longitud.petalo anchura.petalo       clase
1             5.1            3.5             1.4            0.2 Iris-setosa
2             4.9            3.0             1.4            0.2 Iris-setosa
3             4.7            3.2             1.3            0.2 Iris-setosa
4             4.6            3.1             1.5            0.2 Iris-setosa
5             5.0            3.6             1.4            0.2 Iris-setosa
```

Si lo que ahora queremos es obtener un resumen de una columna tipo factor, como es la de clase, hacemos

```
> summary(iris[[5]])
    Iris-setosa Iris-versicolor  Iris-virginica
             50              50              50
```

Si por otro lado, utilizamos `summary` para columnas numéricas, tendremos

```
> summary(iris[[1]])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.300   5.100   5.800   5.843   6.400   7.900
> summary(iris[[2]])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.800   3.000   3.054   3.300   4.400
> summary(iris[[3]])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.600   4.350   3.759   5.100   6.900
> summary(iris[[4]])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.100   0.300   1.300   1.199   1.800   2.500
```

que corresponde a una lista de seis parámetros de estadística descriptiva más bien básicos y que nos sirven para darnos una idea superficial aunque rápida de las características de nuestros datos. El primero de ellos es el valor mínimo, el segundo el límite por debajo del cual están el 25% de valores más bajos si los ordenamos de menor a mayor, es el valor que se encuentra a la mitad de esa lista (no la media), el cuarto es la media, el quinto el tercer cuartil o el valor por debajo del cual está el 75% de valores menores que el mismo y, por último, el valor máximo.

Aunque también podíamos haberlo hecho de una vez

```
> summary(iris)
 longitud.sepalo anchura.sepalo  longitud.petalo anchura.petalo
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.054   Mean   :3.759   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
             clase
 Iris-setosa    :50
 Iris-versicolor:50
 Iris-virginica :50
```

Es interesante notar que, por ejemplo, la diferencia entre media y mediana nos puede dar una idea del nivel de *skewness* de la muestra. Por ejemplo, si representamos un histograma para la longitud del sépalo, con el comando

```
> hist(iris$longitudsepalo,prob="T")
```

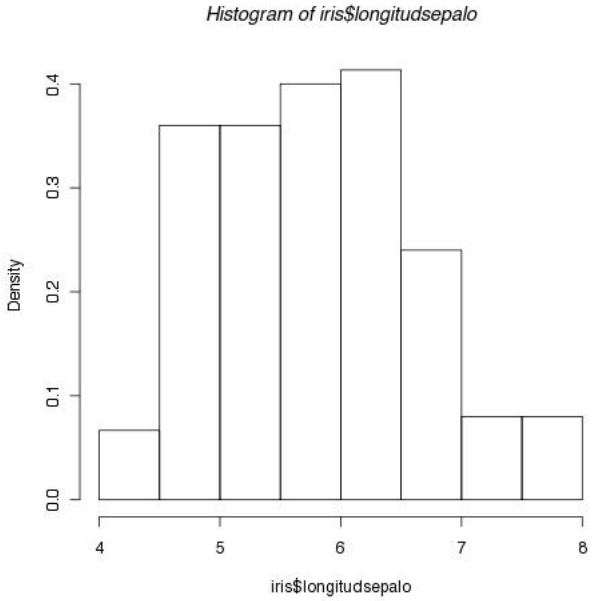

Figure 2: Histograma de longitud del sépalo

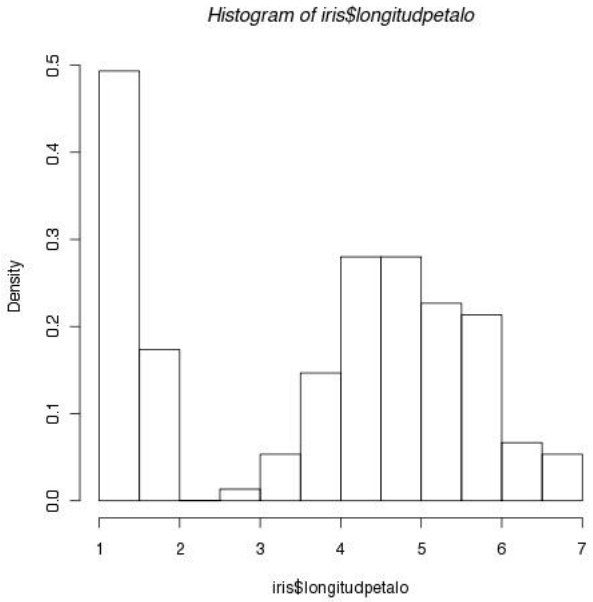

Figure 3: Histograma de longitud del Pétalo

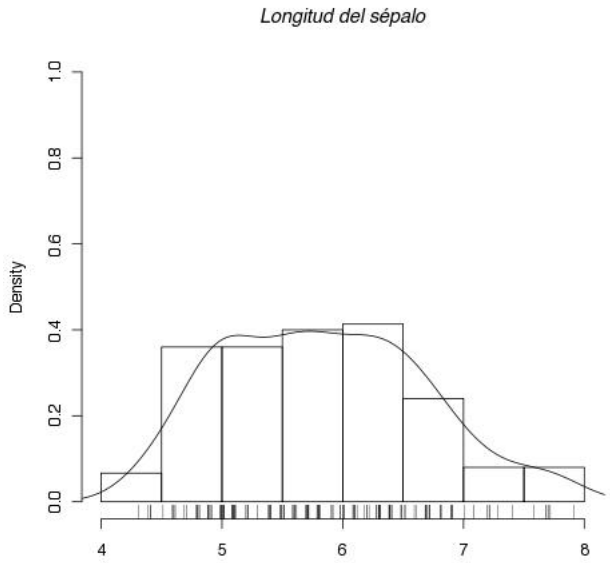

Figure 4: Histograma enriquecido de longitud del sépalo

tendremos la figura 2, con la que podemos comprobar que se aproxima ligeramente a una normal (media y mediana son similares). Si ahora echamos la vista a la longitud del pétalo, podemos comprobar con la figura 3 que la distribución está *torcida*. A su vez, comprobamos que media y mediana son diferentes.

Pero podemos representar, mediante una versión del histograma bastante enriquecido, el gráfico de la figura 4. y lo hacemos mediante la secuencia de comandos siguiente

```
> hist(iris$longitudsepalo, xlab="",
    main="Longitud del sépalo", ylim=0:1,prob=T)
> lines(density(iris$longitudsepalo,na.rm=T))
> rug(jitter(iris$longitudsepalo))
```

en el que podemos, mediante el uso de kernels, una estimación de la pdf. Además, también podemos ver mediante el tercer comando, la representación de los valores reales del atributo, bajo el eje x. Con ello podemos comprobar la existencia de algún outlier. Aquí vemos que no hay ningún valor que destace significativamente del resto (lo cual es normal ya que este cjto. no es precisamente muy irregular).

Resulta bastante más conveniente representar toda esta información visualmente mediante un R, así

```
> boxplot(iris[[1]],iris[[2]],iris[[3]],iris[[4]],names = c("Long. Sépalo",
"Anchura Sépalo", "Long. Pétalo", "Anchura Pétalo"))
```

y tendremos el diagrama tipo Whisker-Plot que aparece en la figura 5. a la derecha, en el que aparecen el máximo y mínimo que no están fuera de rango, el percentil 25% (i.e. Q_1) y el 75% (i.e. Q_3), la mediana y los valores fuera de rango (i.e. *outliers*). La mediana se obtiene con el valor en medio de la lista de valores, si el número es impar, o con la media aritmética de los dos valores en el centro. Un dato se considera un outlier si está más allá de de 3/2 de la diferencia entre los valores Q_3 y Q_1.

¿Qué podemos decir a la luz de los Whiskerplot de la derecha de la figura 5? Individualmente, podemos ver que en la anchura del sépalo hay valores bastante extremos, lo que podría dificultar el proceso de aprendizaje. Por otro lado, tanto la longitud como la anchura del sépalo tienen una mediana más o menos centrada lo que puede llevar a pensar en una distribución normal de los datos. No podemos decir lo mismo de los vectores relativos al pétalo. Esta afirmación queda totalmente contrastada si utilizamos gráficas del tipo $Q-Q$ como las que aparecen en la figura 6. En estas gráficas podemos ver una representación de los percentiles de cada uno de los atributos, con respecto a los de una normal centrada en cero. Como referencia, se incluye

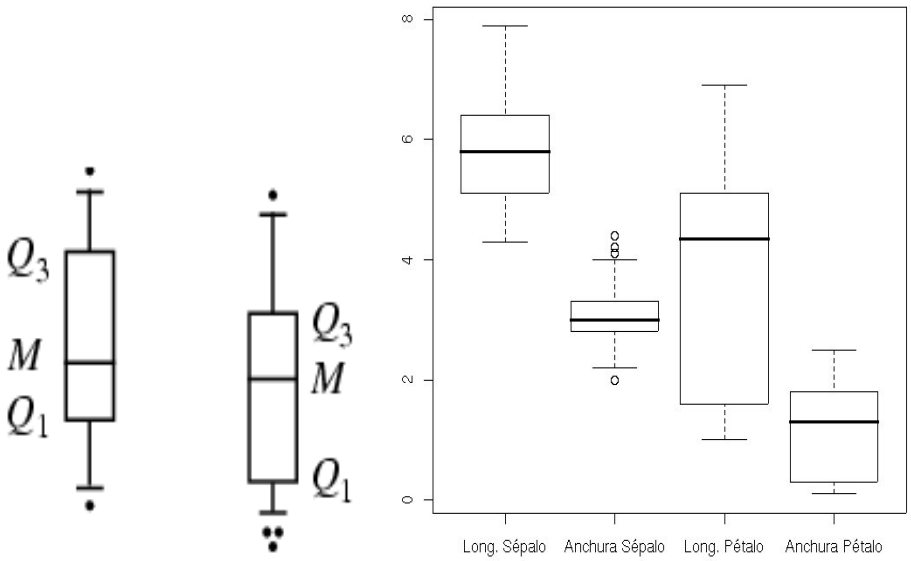

Figure 5: Diagramas Whisker-Plot típicos para dos vectores de datos (izquierda) y los correspondientes a los cuatro atributos de entrada del conjunto Iris (derecha).

una linea con una orientación de 45^0 con respecto al eje orizontal. Cada punto (x, y) hace referencia a un percentil concreto del 0 al 100, de tal forma que si el del percentil 25% es (a, b) significa que el de la normal es a y el del vector de nuestro conjunto de datos es b. Cuanto más se ajusten a la línea representada, más se parecerá a una distribución normal. Dichas gráficas las vamos a obtener con los siguientes comandos:

```
> qqnorm(iris[[1]], main="Gráfica Q-Q para Long. Sépalo")
> qqline(iris[[1]], main="Gráfica Q-Q para Long. Sépalo")
> qqline(iris[[2]], main="Gráfica Q-Q para Ancho Sépalo")
> qqline(iris[[2]], main="Gráfica Q-Q para Ancho Sépalo")
> qqnorm(iris[[3]], main="Gráfica Q-Q para Long. Pétalo")
> qqline(iris[[3]], main="Gráfica Q-Q para Long. Pétalo")
> qqnorm(iris[[4]], main="Gráfica Q-Q para Ancho Pétalo")
> qqline(iris[[4]], main="Gráfica Q-Q para Ancho Pétalo")
```

Hay que tener en cuenta que los percentiles quedan a la izquierda de la línea trazada como referencia, se dice que la distribución de probabilidad está desplazada a la izquierda. Una distribución desplazada a izquierda la tenemos en la figura 7.

Hasta ahora hemos tratado cada uno de los vectores independientemente, con respecto al resto. Pero también podemos relacionarlos entre ellos para ver correlaciones y demás detalles interesantes. Podemos representar, en una simple gráfica de puntos en dos dimensiones, los cinco vectores por pares, como en la figura 8.

Esta panorámica es muy interesante y útil para ver cómo se relacionan las variables por pares. Obsérvese que la diagonal principal de la matriz formada por las gráficas está formada por las etiquetas de los vectores que aparecen en el eje horizontal, para todas las gráficas de la columa. Análogamente, aparecen en el eje vertical, para todas las gráficas de la fila. Por ejemplo, la gráfica inferior izquierda hace referencia a la gráfica (v_1, v_5). Podemos comprobar, para esa última fila, como los atributos v_1, v_2, v_3 y v_4, de izquierda a derecha y respectivamente, sirven para clasificar, en cierta medida, los puntos. Otras gráficas llaman la atención por la alta correlación entre sus atributos (e.g. v_3 y v_4).

Sin embargo, si lo que queremos es realmente ver cómo influye cada uno de los atributos en el problema de clasificación, atendiendo a cada una de las tres clases, podemos usar las gráficas de la figura 9. Son gráficas que muestran las distribuciones de probabilidad, para cada clase, representada por un color diferente. En este

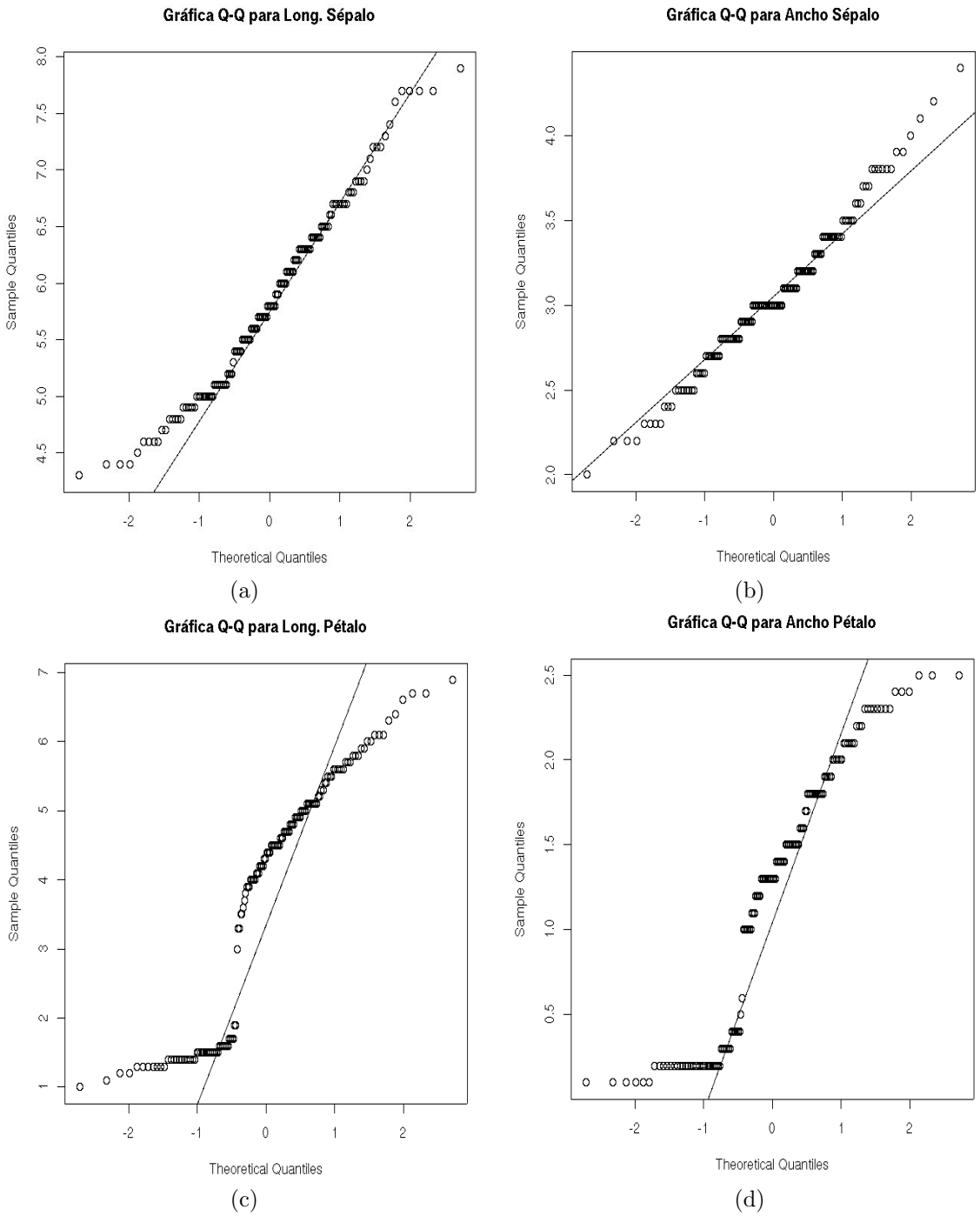

Figure 6: Comprobación visual para determinar si los vectores se ajustan a una normal

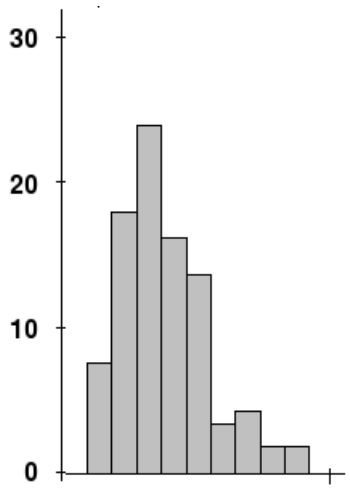

Figure 7: Distribución desplazada a la izquierda

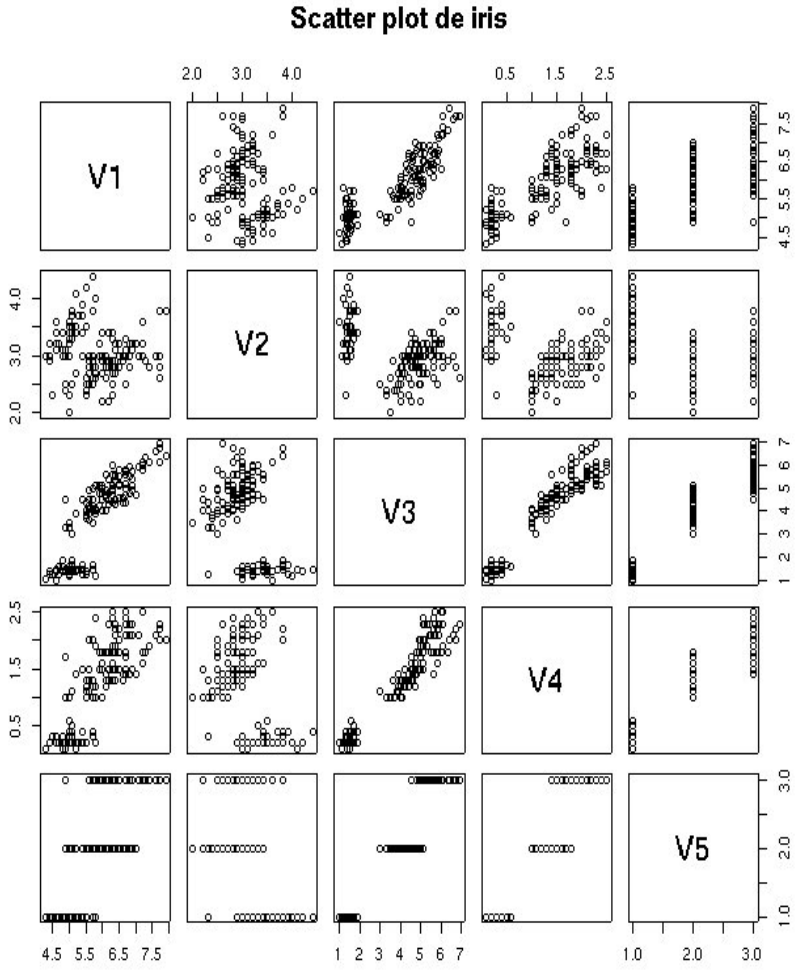

Figure 8: Representación del scatter plot para el conjunto iris

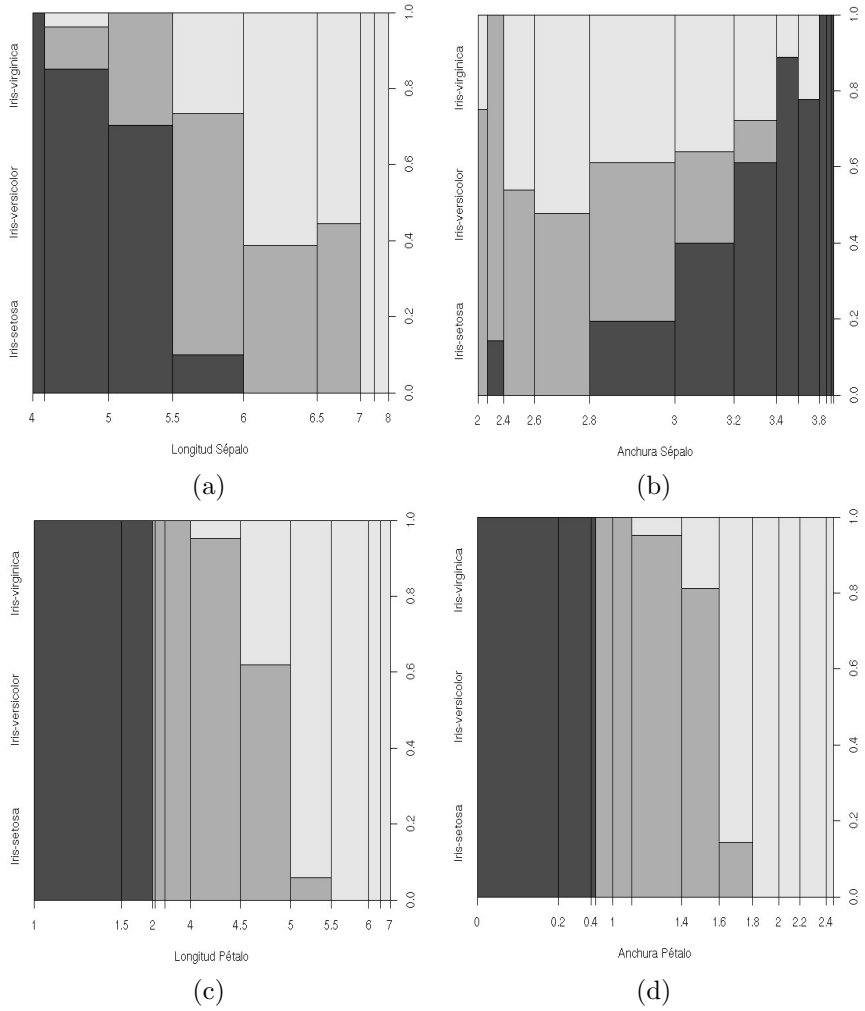

Figure 9: Gráficas de distribución de probabilidad de las tres clases del conjunto Iris, para cada uno de los cuatro vectores de entrada.

caso, tres grises. Los diferentes valores del vector numérico correspondiente se muestran en el eje horizontal y las probabilidades de cada clase, según el rango de valores utilizado en el eje horizontal, aparecen en vertical. Obśervese como parece que, a priori, los atributos x_3 y x_4 discriminan bastante bien las tres clases (alto número de intervalos en el eje horizontal, con probabilidad 1).

4 Trabajando con datos categóricos

A menudo, en el contexto del aprendizaje nos vamos a encontrar con conjuntos de datos en los que tanto $\bar{x}$ como y son categóricos. Es, por ejemplo, el caso del conjuto de datos `breast-cancer`[3]. En este conjunto de datos hay 10 atributos categóricos, y se corresponde con un problema de clasificación binaria. En este caso, un comando del tipo

```
> breast <- read.table("breast-cancer.data",sep=",")
> breast
```

mostrará el siguiente contenido

[3]http://www.ics.uci.edu/∼mlearn/databases/breast-cancer/

```
                      V1    V2      V3    V4    V5  V6 V7    V8        V9 V10
1   no-recurrence-events 30-39 premeno 30-34   0-2  no  3  left  left_low  no
2   no-recurrence-events 40-49 premeno 20-24   0-2  no  2 right  right_up  no
3   no-recurrence-events 40-49 premeno 20-24   0-2  no  2  left  left_low  no
4   no-recurrence-events 60-69    ge40 15-19   0-2  no  2 right   left_up  no
...
```

y si queremos obtener un resumen estadístico de datos categóricos como estos, obtendremos lo siguiente:

```
> summary(breast)
```

lo que obtendremos será una descripción básica de la distribución de las distintas categorías, para cada atributo (obsérvese que el séptimo atributo, aunque es categórico, con valores 1, 2 y 3, se interpreta como numérico):

```
                    V1          V2          V3          V4         V5
no-recurrence-events:201   20-29: 1   ge40   :129   30-34  :60   0-2  :213
recurrence-events   : 85   30-39:36   lt40   :  7   25-29  :54   12-14:  3
                           40-49:90   premeno:150   20-24  :50   15-17:  6
                           50-59:96                 15-19  :30   24-26:  1
                           60-69:57                 10-14  :28   3-5  : 36
                           70-79: 6                 40-44  :22   6-8  : 17
                                                    (Other):42   9-11 : 10
  V6            V7           V8           V9       V10
?  :  8   Min.   :1.000   left :152   ?        :  1   no :218
no :222   1st Qu.:2.000   right:134   central  : 21   yes: 68
yes: 56   Median :2.000               left_low :110
          Mean   :2.049               left_up  : 97
          3rd Qu.:3.000               right_low: 24
          Max.   :3.000               right_up : 33
```

5 Tratamiento de valores fuera de rango

Para este apartado, vamos a usar el excelente aunque inacabado libro de Luis Torgo que podeis encontrar en la Web de la asignatura. En este, se ofrecen algunos ejemplos sencillos de cómo tratar los valores fuera de rango de un conjunto con outliers típico. Este muestra, para cada ejemplar, una muestra de agua de un determinado rio, que nos ayudará a, mediante la medición de determinados compuestos químicos, y la proporción de determinados tipos de alga en el agua, la probabilidad de aparición de focos de crecimiento de algas dañinas para el rio.

Cargamos los datos (hay tres ficheros, de aprendizaje, de evaluación y el tercero con soluciones reales).

```
>algas <- read.table("algas_a.txt", header=F, dec='.',
         col.names=c('season','size','speed','mxPH','mnO2','C1','NO3',
         'NH4','oPO4','PO4','Chla','a1','a2','a3','a4','a5','a6','a7'),
         na.strings=c("XXXXXXX"))
>algas[1:5,]

  season  size  speed mxPH mnO2     C1    NO3     NH4    oPO4     PO4 Chla  a1
1 winter small medium 8.00  9.8 60.800  6.238 578.000 105.000 170.000 50.0 0.0
2 spring small medium 8.35  8.0 57.750  1.288 370.000 428.750 558.750  1.3 1.4
3 autumn small medium 8.10 11.4 40.020  5.330 346.667 125.667 187.057 15.6 3.3
4 spring small medium 8.07  4.8 77.364  2.302  98.182  61.182 138.700  1.4 3.1
5 autumn small medium 8.06  9.0 55.350 10.416 233.700  58.222  97.580 10.5 9.2
```

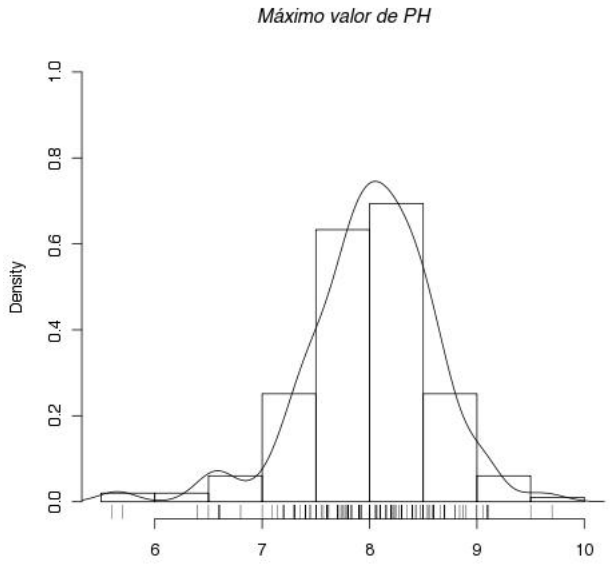

Figure 10: Histograma enriquecido para lecturas de PH máximo

```
    a2   a3  a4   a5  a6  a7
1  0.0  0.0 0.0 34.2 8.3 0.0
2  7.6  4.8 1.9  6.7 0.0 2.1
3 53.6  1.9 0.0  0.0 0.0 9.7
4 41.0 18.9 0.0  1.4 0.0 1.4
5  2.9  7.5 0.0  7.5 4.1 1.0
```

Obsérvese que, dado que el cjto. tiene valores nulos y aparecen con esa serie de caracteres 'X' en el cjto. R los tratará como nulos cuando los lea.

Si nos fijamos en el atributo que nos indica en máximo valor de ph medido en el agua, y hacemos uso de la representación de histograma enriquecido que hemos usado antes, que aparece ahora en la figura 10.

```
> hist(algas$mxPH, xlab="",
    main="Máximo valor de PH", ylim=0:1,prob=T)
> lines(density(algas$mxPH,na.rm=T))
> rug(jitter(algas$mxPH))
```

podremos comprobar ahora que si nos fijamos en los valores para ese atributo de los distintos ejemplares, podremos comprobar que hay otuliers en valores muy pequeños y muy grandes de ph. Si ahora utilizamos

```
>boxplot(algas$oPO4,boxwex=0.15,ylab='Orthophosphate(oPO4)')
>rug(jitter(algae$oPO4),side=2)
>abline(h=mean(algae$oPO4,na.rm=T),lty=2)
```

para conseguir un Wisker plot, con el que además, representamos los ejemplares en el eje de ordenadas, y la media mediante una línea horizontal punteada obtendremos la figura 11. en la que podemos ver claramente que hay una gran cantidad de outliers con valores excesivamente grandes. Otro dato que nos confirma esto es que la media está por encima de la mediana, indicada por la caja dispuesta verticalmente. El valor de la media está distorsionado debido a la presencia de esos valores. Si queremos identificar qué puntos del frame de datos tienen un valor exageradamente grande, podemos hacerlo con una sencilla indexación

```
> algas[algas$oPO4 > 300,]
     season   size  speed mxPH mnO2      C1   NO3     NH4    oPO4     PO4
```

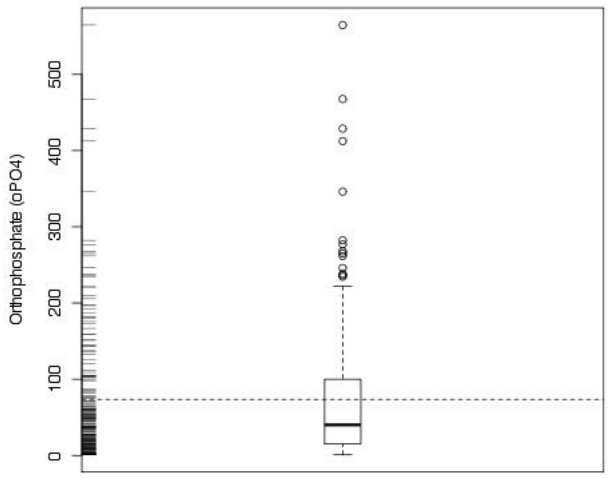

Figure 11: Whisker plot, media y jitter par el ortofosfato

```
2    spring  small medium 8.35  8.0  57.750 1.288  370.00 428.750 558.750
20   spring  small medium 7.79  3.2  64.000 2.822 8777.60 564.600 771.600
21   winter  small medium 7.83 10.7  88.000 4.825 1729.00 467.500 586.000
NA     <NA>   <NA>   <NA>   NA   NA      NA    NA       NA       NA       NA
88   winter medium medium 7.80  3.6  48.667 4.030 5738.33 412.333 607.167
172  summer  large medium 7.91  6.2 151.833 3.923 1081.66 346.167 388.167
NA.1   <NA>   <NA>   <NA>   NA   NA      NA    NA       NA       NA       NA
       Chla  a1   a2  a3   a4  a5  a6  a7
2     1.300 1.4  7.6 4.8  1.9 6.7 0.0 2.1
20    4.500 0.0  0.0 0.0 44.6 0.0 0.0 1.4
21   16.000 0.0  0.0 0.0  6.8 6.1 0.0 0.0
NA       NA  NA   NA  NA   NA  NA  NA  NA
88    4.300 0.0  0.0 2.6  2.4 5.0 0.0 2.4
172   5.083 1.7 12.0 4.9  2.7 0.0 5.9 1.7
NA.1     NA  NA   NA  NA   NA  NA  NA  NA
```

y tendremos todas aquellas muestras con un valor exageradamente grande.

Ahora supongamos que queremos ahondar en la relación entre dos variables. Digamos, por ejemplo, cómo se distribuyen las diferentes concentraciones de alga `a1` para los tres tipos de rios que hay

```
> bwplot(size ~ a1,data=algas,ylab="Tamaño del rio",xlab="Alga A1")
```

y obtenemos el sencillo Whisker plot discriminado para cada uno de los valores, que aparece en la figura 12. con la que podemos deducir que, para esta alga, tendremos altas concentraciones de la misma en rios pequeños. Lo cual puede ser bastante valioso como conocimiento en nuestro análisis. Obsérvese que hemos contrastado un valor discreto (i.e. un factor) con una variable real. Podemos hacerlo también con dos variables reales, siempre que una de ellas la discreticemos.

Vamos a ver cómo discretizar una variable. Por ejemplo, la variable de concentración mínima de oxígeno, `mnO2`.

```
> min02 <- equal.count(na.omit(algas$mnO2),number=4, overlap=1/5)
> min02
```

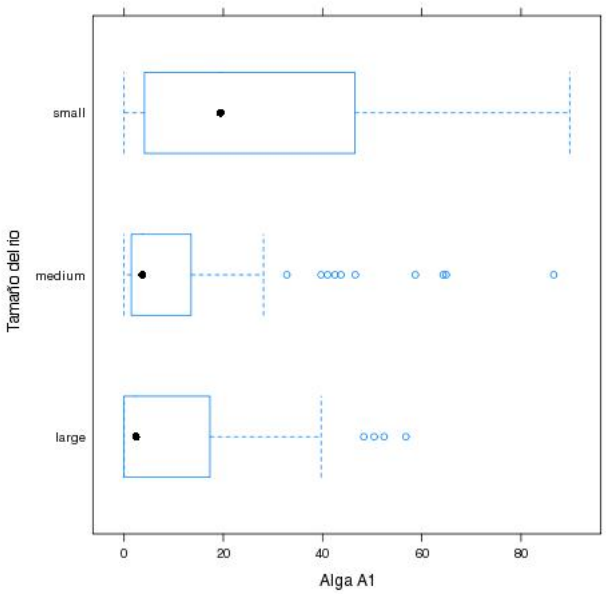

Figure 12: Concentraciones del alga a1, según el tamaño del rio

```
Data:
  [1]  9.80  8.00 11.40  4.80  9.00 13.10 10.30 10.60  3.40  9.90 10.20 11.70
 [13]  9.60 11.80  9.60 11.50 12.00  9.80 10.40  3.20 10.70  9.20 10.30  8.50
 [25]  9.40 10.70  8.40 11.10  9.80 11.30 12.50 10.30 11.30  9.90  7.80  8.40
 ....

Intervals:
     min    max count
1  1.495  8.205    60
2  7.595  9.905    62
3  9.695 10.805    60
4 10.695 13.405    61

Overlap between adjacent intervals:
[1] 14 16 15
```

Se han creado, utilizando el comando `equal.count()` cuatro bins, como los vistos en clase, del atributo, al que previamente se le han eliminado los elementos nulos. El algoritmo utilizado ha sido el `equal count` que tiene un valor de solape de 1/5, lo cual significa que seguramente habrá valores que se dupliquen en intervalos contiguos ya que los bins han de crearse de tal forma que tengan un número igual de ejemplares. Si ahora representamos la variable discreta `season` con respecto a la concentración del tipo 3 de alga, con respecto a posibles valores mínimos de O2, tenemos

```
> stripplot(season ~ a3|mnO2,data=algas[!is.na(algas$mnO2),])
```

la figura 13 Obsérvese que el valor de concentraciones del tipo de alga `a3` se muestran con respecto a la variable `season`. Esto se ha hecho para cada bin diferente de la variable `minO2`. Se han dispuesto las gráficas para los distintos bins desde izquierda a derecha y de abajo a arriba. Obsérvese el uso de la función `is.na(algas$mnO2)` para evitar que los valores nulos introdujeran ruido en el análisis.

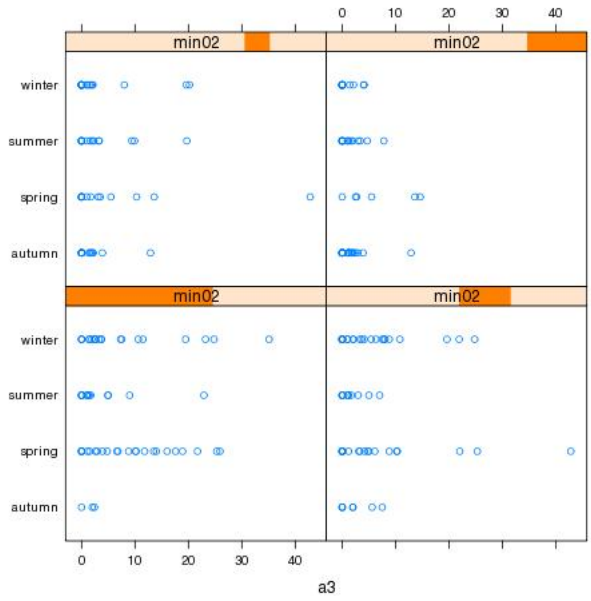

Figure 13: Concentraciones del alga a3, según temporada y niveles de O2 mínimos

6 Tratamiento de valores nulos

Ya hemos visto en clase cómo se pueden manejar los valores nulos. La forma más inmediata de tratarlos es deshacerse de ellos. Sin embargo, dichos nulos, o las tuplas que los incluyen pueden acarrear información importante que de esta forma se perdería. Podemos sustituirlos con los valores más frecuentes, en todo caso. También podemos hacer uso de información proveniente de correlaciones entre variables, la modelamo, y reproducimos el nulo. También podemos explorar la similitud entre diferentes casos. Todos estos casos los vamos a ver en este apartado.

6.1 Eliminación de observaciones con nulos

Para tomar medida de cuán serio es el problema de los valores nulos, debemos contar los casos, y lo hacemos con

```
> algas[!complete.cases(algas),]
    season   size  speed mxPH mnO2    Cl   NO3 NH4    oPO4     PO4  Chla   a1
28  autumn  small   high 6.80 11.1 9.000 0.630  20   4.000      NA  2.70 30.3
38  spring  small   high 8.00   NA 1.450 0.810  10   2.500   3.000  0.30 75.8
48  winter  small    low   NA 12.6 9.000 0.230  10   5.000   6.000  1.10 35.5
55  winter  small   high 6.60 10.8    NA 3.245  10   1.000   6.500    NA 24.3
56  spring  small medium 5.60 11.8    NA 2.220   5   1.000   1.000    NA 82.7
57  autumn  small medium 5.70 10.8    NA 2.550  10   1.000   4.000    NA 16.8
58  spring  small   high 6.60  9.5    NA 1.320  20   1.000   6.000    NA 46.8
59  summer  small   high 6.60 10.8    NA 2.640  10   2.000  11.000    NA 46.9
60  autumn  small medium 6.60 11.3    NA 4.170  10   1.000   6.000    NA 47.1
61  spring  small medium 6.50 10.4    NA 5.970  10   2.000  14.000    NA 66.9
62  summer  small medium 6.40   NA    NA    NA  NA      NA  14.000    NA 19.4
63  autumn  small   high 7.83 11.7 4.083 1.328  18   3.333   6.667    NA 14.4
116 winter medium   high 9.70 10.8 0.222 0.406  10  22.444  10.111    NA 41.0
161 spring  large    low 9.00  5.8    NA 0.900 142 102.000 186.000 68.05  1.7
184 winter  large   high 8.00 10.9 9.055 0.825  40  21.083  56.091    NA 16.8
```

```
199 winter  large medium 8.00  7.6    NA    NA  NA      NA      NA    NA  0.0
      a2  a3   a4  a5  a6  a7
28   1.9 0.0  0.0 2.1 1.4 2.1
38   0.0 0.0  0.0 0.0 0.0 0.0
48   0.0 0.0  0.0 0.0 0.0 0.0
55   0.0 0.0  0.0 0.0 0.0 0.0
56   0.0 0.0  0.0 0.0 0.0 0.0
57   4.6 3.9 11.5 0.0 0.0 0.0
58   0.0 0.0 28.8 0.0 0.0 0.0
59   0.0 0.0 13.4 0.0 0.0 0.0
60   0.0 0.0  0.0 0.0 1.2 0.0
61   0.0 0.0  0.0 0.0 0.0 0.0
62   0.0 0.0  2.0 0.0 3.9 1.7
63   0.0 0.0  0.0 0.0 0.0 0.0
116  1.5 0.0  0.0 0.0 0.0 0.0
161 20.6 1.5  2.2 0.0 0.0 0.0
184 19.6 4.0  0.0 0.0 0.0 0.0
199 12.5 3.7  1.0 0.0 0.0 4.9
> nrow(algas[!complete.cases(algas),])
[1] 16
```

La función `complete.cases()` devuelve un `TRUE` para cada fila con un caso completo (i.e. sin valores nulos). Si lo negamos, tendremos un `TRUE` para cada caso con nulos. Con el indexado lógico, obtenemos aquellas filas con nulos. Finalmente, con `nrow()` contamos las filas. Tenemos un total de 16, de 200. Un porcentaje importante que conviene tratar.

En el caso de que decidieramos deshacernos de los nulos, con hacer

```
algas <- na.omit(algas)
```

sería suficiente. Si queremos conservarlos, aun podemos limpiar aquellos ejemplares con un número de nulos alto tales que las hacen inservibles. Las tuplas 62 y 199 tienen un total de seis valores nulos. Por tanto, podemos hacer

```
algas <- algas[-c(62,199),]
```

6.2 Sustitución mediante valores representativos

Es interesante en determinados casos utilizar valores centrales de atributos con valores nulos, para sustituir a éstos. Como ya sabemos por las clases de teoría, dependiendo de lo similar que una pdf de una muestra sea a una normal, la media puede ser una buena medida para ello. Si, por el contrario, la muestra tiene una distribución desplazada *skewed*, la mediana es mejor opción.

Por tanto, antes de tomar una decisión, habría que echar un vistazo a los datos, para determinar la pdf de la muestra. En este ejemplo concreto, el ejemplar 48 no tiene valor en la variable `mxPH`. Dado que dicha variable tiene una distribución parecida a la normal, hacemos

```
> algae[48,'mxPH'] <- mean(algae$mxPH,na.rm=T)
```

con la que en la celda correspondiente, utilizamos la media de la columna correspondiente, sin utilizar valores nulos para su cálculo. Otras veces necesitaremos trabajar de manera intensiva con una columna. Por tanto, en esos casos un acceso directo no va a ser la mejor manera de proceder. Es el caso de la variable `Chla`, que tiene una distribución *skewed*, así que utilizaremos la mediana.

```
> algae[is.na(algae$Chla),'Chla'] <- median(algae$Chla,na.rm=T)
```

Nótese que estos métodos introducen un sesgo muy importante (i.e. todos se sustituyen por un valor más o menos representativo). Lo cual puede influenciar el análisis posterior. Son interesantes ya que su coste computacional es muy bajo. Sin embargo, métodos sin sesgo serían deseables cuando el tamaño del problema lo permita.

6.3 Sustitución mediante estudio de correlaciones

La idea es simple. Si uno de los atributos con valores nulos tiene una fuerte correlación con otro, podemos aprovechar ese hecho para así generar sustitutos para los valores nulos, mediante una técnica que introduce poco sesgo y tiene un bajo coste computacional.

Si recordamos el atributo `mxPH` y la muestra 48, si encontráramos un atributo altamente correlado con él, esa muestra, su valor nulo, podría ser sustituido sin problemas. Podemos probar a hacer

```
> cor(algas[,4:18], use="complete.obs")
            mxPH        mnO2          C1         NO3         NH4         oPO4
mxPH  1.00000000 -0.10269374  0.14709539 -0.17213024 -0.15429757  0.090229085
mnO2 -0.10269374  1.00000000 -0.26324536  0.11790769 -0.07826816 -0.393752688
C1    0.14709539 -0.26324536  1.00000000  0.21095831  0.06598336  0.379255958
NO3  -0.17213024  0.11790769  0.21095831  1.00000000  0.72467766  0.133014517
NH4  -0.15429757 -0.07826816  0.06598336  0.72467766  1.00000000  0.219311206
oPO4  0.09022909 -0.39375269  0.37925596  0.13301452  0.21931121  1.000000000
PO4   0.10132957 -0.46396073  0.44519118  0.15702971  0.19939575  0.911964602
Chla  0.43182377 -0.13121671  0.14295776  0.14549290  0.09120406  0.106914784
a1   -0.16262986  0.24998372 -0.35923946 -0.24723921 -0.12360578 -0.394574479
a2    0.33501740 -0.06848199  0.07845402  0.01997079 -0.03790296  0.123811068
a3   -0.02716034 -0.23522831  0.07653027 -0.09182236 -0.11290467  0.005704557
a4   -0.18435348 -0.37982999  0.14147281 -0.01448875  0.27452000  0.382481433
a5   -0.10731230  0.21001174  0.14534877  0.21213579  0.01544458  0.122027482
a6   -0.17273795  0.18862656  0.16904394  0.54404455  0.40119275  0.003340366
a7   -0.17027088 -0.10455106 -0.04494524  0.07505030 -0.02539279  0.026150420
             PO4        Chla          a1           a2           a3          a4
mxPH  0.10132957  0.43182377 -0.16262986  0.335017401 -0.027160336 -0.18435348
mnO2 -0.46396073 -0.13121671  0.24998372 -0.068481989 -0.235228307 -0.37982999
C1    0.44519118  0.14295776 -0.35923946  0.078454019  0.076530269  0.14147281
NO3   0.15702971  0.14549290 -0.24723921  0.019970786 -0.091822358 -0.01448875
NH4   0.19939575  0.09120406 -0.12360578 -0.037902958 -0.112904666  0.27452000
oPO4  0.91196460  0.10691478 -0.39457448  0.123811068  0.005704557  0.38248143
PO4   1.00000000  0.24849223 -0.45816781  0.132667891  0.032193981  0.40883951
Chla  0.24849223  1.00000000 -0.26601088  0.366724647 -0.063301128 -0.08600540
a1   -0.45816781 -0.26601088  1.00000000 -0.262665485 -0.108177581 -0.09338072
a2    0.13266789  0.36672465 -0.26266549  1.000000000  0.009759915 -0.17628704
a3    0.03219398 -0.06330113 -0.10817758  0.009759915  1.000000000  0.03336910
a4    0.40883951 -0.08600540 -0.09338072 -0.176287038  0.033369102  1.00000000
a5    0.15548900 -0.07342837 -0.26972709 -0.186758940 -0.141610948 -0.10131827
a6    0.05320294  0.01032550 -0.26156402 -0.133518480 -0.196900051 -0.08488426
a7    0.07978353  0.01760782 -0.19306384  0.036206205  0.039060248  0.07114638
              a5           a6          a7
mxPH -0.10731230 -0.172737947 -0.17027088
mnO2  0.21001174  0.188626555 -0.10455106
C1    0.14534877  0.169043945 -0.04494524
NO3   0.21213579  0.544044553  0.07505030
NH4   0.01544458  0.401192749 -0.02539279
oPO4  0.12202748  0.003340366  0.02615042
PO4   0.15548900  0.053202942  0.07978353
Chla -0.07342837  0.010325497  0.01760782
a1   -0.26972709 -0.261564023 -0.19306384
a2   -0.18675894 -0.133518480  0.03620621
a3   -0.14161095 -0.196900051  0.03906025
a4   -0.10131827 -0.084884259  0.07114638
a5    1.00000000  0.388608955 -0.05149346
```

```
a6    0.38860896  1.000000000 -0.03033428
a7   -0.05149346 -0.030334277  1.00000000
```

con lo que podemos comprobar que, al ser tantos atributos, la matriz no es demasiado informativa. Podríamos buscar un atributo realmente correlado pero es más cómodo usar la siguiente forma

```
> symnum(cor(algae[,4:18],use="complete.obs"))
     mP mO C1 NO NH o P Ch a1 a2 a3 a4 a5 a6 a7
mxPH 1
mnO2    1
C1         1
NO3           1
NH4           ,  1
oPO4    .  .        1
PO4     .  .        * 1
Chla .                  1
a1          .        . .    1
a2   .                  .     1
a3                             1
a4      .           . .           1
a5                                   1
a6             .  .                  .  1
a7                                         1
attr(,"legend")
[1] 0 ‘ ’ 0.3 ‘.’ 0.6 ‘,’ 0.8 ‘+’ 0.9 ‘*’ 0.95 ‘B’ 1
```

con la que obtenemos una representación simbólica de esta matriz. Si nos fijamos en la leyenda que se nos proporciona, y buscamos marcas como comas o asteríscos (no hay signo más o letra B), encontramos que las parejas `NH4, NO` y `PO4, o` están correladas. Nos quedamos con la correlación más fuerte. Si utilizamos los comandos

```
> algas[is.na(algas$PO4),]
    season  size  speed mxPH mnO2 C1  NO3 NH4 oPO4 PO4 Chla   a1   a2  a3 a4
28  autumn small   high  6.8 11.1  9 0.63  20    4  NA  2.7 30.3  1.9 0.0  0
199 winter large medium  8.0  7.6 NA   NA  NA   NA  NA   NA  0.0 12.5 3.7  1
     a5  a6  a7
28  2.1 1.4 2.1
199 0.0 0.0 4.9
> algas[is.na(algas$o),]
    season  size  speed mxPH mnO2 C1 NO3 NH4 oPO4 PO4 Chla   a1   a2  a3 a4 a5
62  summer small medium  6.4   NA NA  NA  NA   NA  14   NA 19.4  0.0 0.0  2  0
199 winter large medium  8.0  7.6 NA  NA  NA   NA  NA   NA  0.0 12.5 3.7  1  0
     a6  a7
62  3.9 1.7
199 0.0 4.9
```

para reparar el atributo `PO4`, vemos que solamente la instancia 28 es subsceptible de ello ya que, para ambos, el resto de instancias con nulos tienen excesivos y hay que eliminarlas.

Ahora, podemos encontrar un modelo de correlación lineal entre las dos viriables, de una forma extremadamente fácil, con

```
> lm(oPO4 ~ PO4,data=algas)

Call:
lm(formula = oPO4 ~ PO4, data = algas)

Coefficients:
```

```
(Intercept)          PO4
   -15.6142       0.6466
```

y lo que estamos haciendo es pedirle a R que genere un modelo lineal con `lm()`, en el que se aproxime `oPO4` con `PO4`, utilizando los datos fuente. Así, el modelo lineal que se obtiene es el siguiente

$$oPO4 = 0.6466 \times P04 - 15.6142.$$

Por tanto, solo nos resta aplicar dicho modelo a la sustitución del valor nulo en el ejemplar 28, de la siguiente forma

```
algas[28,'PO4'] <- (algas[28,'oPO4']+15.6142)/0.6466)
```

7 Ejercicios

Intentar reproducir el análisis simple que hemos realizado en clase para los conjuntos de datos `wine.data`, `waveform.data`, `covtype.data`.