Técnicas de análisis cuantitativas y cualitativas
Prácticas con R y RStudio

Máster universitario en gestión sostenible de la tierra y el territorio

Universidad de Santiago de Compostela

Eduardo Corbelle Rico

Curso 2014–2015

Documento compilado el 13 de noviembre de 2014.

```
print(sessionInfo(), locale = FALSE)

## R version 2.15.1 (2012-06-22)
## Platform: i486-pc-linux-gnu (32-bit)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets
## [6] methods   base
##
## other attached packages:
## [1] knitr_1.5
##
## loaded via a namespace (and not attached):
## [1] evaluate_0.5.1 formatR_0.10   highr_0.3
## [4] stringr_0.6.2  tools_2.15.1
```

Índice general

Práctica 1

Introducción a R y RStudio

En esta práctica realizaremos una breve introducción al trabajo con R a través del entorno de desarrollo integrado RStudio. Ambas aplicaciones están disponibles para diferentes sistemas operativos y son distribuídas bajo la GNU *General Public License* (GPL), por lo que se trata de aplicaciones libres. El objetivo de esta práctica es comenzar a familiarizarnos con su uso, repasar algunos conceptos elementales de estadística descriptiva e iniciarnos en el paradigma de la *investigación reproducible*.

1.1. Toma de datos

Realizaremos la práctica sobre un conjunto de datos obtenidos de las personas presentes en el aula. Nos interesa obtener, como mínimo, una variable numérica y una variable categórica. Para introducirlos en R trataremos de crear un documento de texto simple (con el block de notas, por ejemplo, o una aplicación similar) que guardaremos con la extensión *.csv* (*comma-separated values*). El aspecto de este fichero debería ser similar a:

```
Altura, Sexo
156, hombre
167, mujer
180, mujer
176, hombre
164, hombre
172, mujer
```

Es importante fijarse en que estamos separando los dos campos (altura y sexo) mediante comas, por lo que de haber necesitado introducir decimales deberíamos haber empleado el punto decimal. Una vez escrito y guardado el fichero, podemos importar los datos en R a partir de él utilizando la orden:

```
datos1 <- read.csv("./Datos/DatosPractica1.csv")
```

1.2. Manipulación de los datos

Una vez importados, los datos deberían aparecer listados en la ventana *Environment* de RStudio. Alternativamente, también podemos solicitar un listado del espacio de trabajo en la consola de R:

```
ls()

## [1] "datos1"
```

Para comprobar que el proceso de importación ha sido correcto, y que todos los campos han sido asignados al tipo de datos que le corresponden podemos solicitar a R que nos indique la estructura del objeto recién importado:

```
str(datos1)

## 'data.frame': 6 obs. of  2 variables:
##  $ Altura: int  156 167 180 176 164 172
##  $ Sexo  : Factor w/ 2 levels " hombre"," mujer": 1 2 2 1 1 2

summary(datos1)

##      Altura         Sexo
##  Min.   :156    hombre:3
##  1st Qu.:165    mujer :3
##  Median :170
##  Mean   :169
##  3rd Qu.:175
##  Max.   :180
```

Como podemos apreciar, R ha asignado la clase *data.frame* al objeto que acabamos de crear, identificado correctamente la existencia de dos campos, y asignado el tipo de datos correcto a cada uno: números enteros (*integer*) para el campo *Altura* y variable categórica (*factor*) para el campo *Sexo*.

Existen diferentes maneras de acceder a los datos de la tabla, para invocar sobre ellos diferentes funciones, por ejemplo:

```
datos1$Altura

## [1] 156 167 180 176 164 172

class(datos1$Altura)

## [1] "integer"

mean(datos1$Altura)

## [1] 169.2

median(datos1$Altura)

## [1] 169.5

quantile(datos1$Altura, probs = 0.5)

##   50%
## 169.5
```

```
max(datos1$Altura)

## [1] 180

min(datos1$Altura)

## [1] 156

sum(datos1$Altura)

## [1] 1015

sd(datos1$Altura)

## [1] 8.681

var(datos1$Altura)

## [1] 75.37

summary(datos1$Altura)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     156     165     170     169     175     180
```

Los datos de la variable categórica *Sexo* pueden ser accedidos de igual modo:

```
datos1$Sexo

## [1]  hombre  mujer   mujer   hombre  hombre  mujer
## Levels:  hombre  mujer

class(datos1$Sexo)

## [1] "factor"

levels(datos1$Sexo)

## [1] " hombre" " mujer"

table(datos1$Sexo)

##
##  hombre   mujer
##       3       3
```

Una manera útil de localizar una observación concreta dentro de un vector de datos es el uso de un subíndice. Por ejemplo, la siguiente expresión nos devuelve el valor de la segunda observación de la variable *Sexo*:

```
datos1$Sexo[2]

## [1]  mujer
## Levels:  hombre  mujer
```

Por otra parte, podemos emplear expresiones lógicas para realizar consultas sobre los datos. Por ejemplo, para localizar las observaciones con altura mayor o igual a 170 cm:

```
datos1$Altura >= 170

## [1] FALSE FALSE  TRUE  TRUE FALSE  TRUE

which(datos1$Altura >= 170)

## [1] 3 4 6

datos1$Altura[which(datos1$Altura >= 170)]

## [1] 180 176 172
```

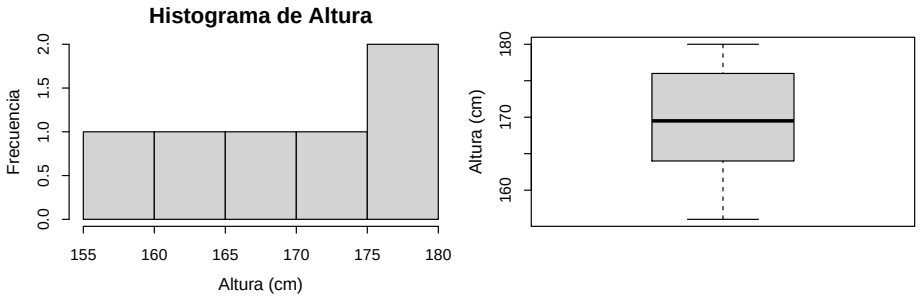
El uso de subíndices puede hacerse extensivo a otras clases de objetos. En una *data.frame* (o una matriz), dado que se trata de un objeto con dos dimensiones, el subíndice tendrá que tener dos valores separados por una coma. El valor inicial indica el número de fila, y el valor final el número de columna. Cuando no se especifica ningún valor en una de las dos posiciones, la expresión devuelve todos los valores de una fila o columna, respectivamente.

```
datos1[1, ]  # Devuelve la primera fila de datos

##   Altura    Sexo
## 1    156  hombre

datos1[, 1]  # Devuelve la primera columna de datos

## [1] 156 167 180 176 164 172

datos1[3, 2]  # Devuelve la tercera observación (fila) de la segunda columna

## [1]  mujer
## Levels:  hombre  mujer

datos1[1:3, ]

##   Altura    Sexo
## 1    156  hombre
## 2    167   mujer
## 3    180   mujer

datos1[c(2, 5), ]

##   Altura    Sexo
## 2    167   mujer
## 5    164  hombre
```

1.2.1. Representación gráfica

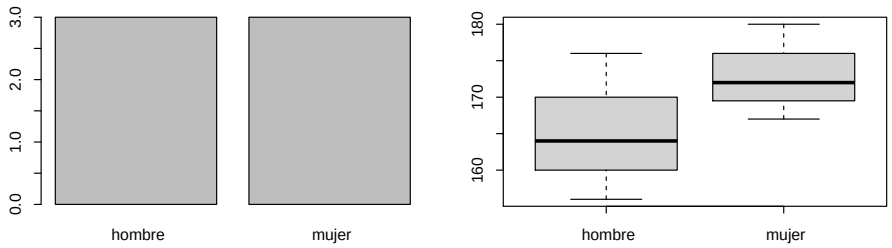
Podemos representar gráficamente la variable *Altura*, por ejemplo, mediante un histograma, o mediante un diagrama de cajas.

```
hist(datos1$Altura, col = "lightgrey", xlab = "Altura (cm)",
    ylab = "Frecuencia", main = "Histograma de Altura")
boxplot(datos1$Altura, col = "lightgrey", ylab = "Altura (cm)")
```



La variable *Sexo* podría ser representada mediante un diagrama de barras. También podemos analizar el comportamiento de la variable *Altura* en función de la variable *Sexo*, mediante dos diagrama de cajas.

```
frecuencias <- table(datos1$Sexo)
barplot(frecuencias)
boxplot(Altura ~ Sexo, data = datos1, col = "lightgrey")
```



Práctica 2

Contrastes de hipótesis

2.1. Sobre la media de una población

Los datos de la práctica anterior corresponden a un censo de las personas que estaban en el aula. No suponen necesariamente, por tanto, una muestra representativa de las personas que trabajan en el campus. Aspectos relacionados con la edad de los presentes, por ejemplo, pueden haber sesgado los valores de altura y sexo recogidos, lo que nos llevaría a conclusiones erróneas sobre esta última población citada. Aún así, en el caso poco probable de que aceptásemos usar estos datos como muestra representativa de la población del campus, podríamos realizar un test de hipotesis para la media de la variable *Altura*. Por ejemplo, si deseamos probar la hipótesis alternativa de que la altura media de la población del campus es mayor de 170 cm:

```
t.test(datos1$Altura, mu = 170, alternative = "greater")

##
##  One Sample t-test
##
## data:  datos1$Altura
## t = -0.2351, df = 5, p-value = 0.5883
## alternative hypothesis: true mean is greater than 170
## 95 percent confidence interval:
##  162 Inf
## sample estimates:
## mean of x
##     169.2
```

Observamos en este caso que, aunque la altura media de la muestra es efectivamente inferior a 170 cm, el contraste no permite descartar la hipótesis nula.

Pudiera ser también que deseásemos comparar la altura media de las subpoblaciones de hombres y mujeres. Por ejemplo, siendo la hipótesis alternativa que la altura de ambas subpoblaciones es diferente:

```
t.test(Altura ~ Sexo, data = datos1, alternative = "two.sided")

##
##  Welch Two Sample t-test
##
```

```
## data:  Altura by Sexo
## t = -1.105, df = 3.438, p-value = 0.3403
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -28.23  12.90
## sample estimates:
## mean in group  hombre  mean in group  mujer
##                 165.3                 173.0
```

De nuevo, aunque la altura media de las dos submuestras es diferente, la diferencia observada no es suficiente para descartar la posibilidad de que sea simplemente debida al azar. En realidad, si hubiéramos observado una diferencia similar pero la muestra fuese mucho mayor (es decir, si el experimento fuese más *potente*), quizá el p-valor asociado al test hubiera sido suficientemente bajo como para rechazar la hipótesis nula de igualdad entre las dos medias.

2.2. Para la proporción de una población

Imaginemos ahora que somos clientes de una fábrica de tornillos. La fábrica asegura que la proporción de tornillos no defectuosos en su producción es del 80 %, pero nuestra experiencia nos hace sospechar que la proporción de defectos es mayor. Con ánimo de comprobarlo, decidimos extraer una muestra aleatoria de 50 tornillos, cuya inspección visual revela que 15 son defectuosos. ¿Es suficiente para enviar una reclamación a la fábrica?

```
prop.test(35, 50, p = 0.8, alternative = "less")

##
##  1-sample proportions test with continuity correction
##
## data:  35 out of 50, null probability 0.8
## X-squared = 2.531, df = 1, p-value = 0.05581
## alternative hypothesis: true p is less than 0.8
## 95 percent confidence interval:
##  0.0000 0.8026
## sample estimates:
##   p
## 0.7
```

Como se puede apreciar en el resultado del test, si bien la proporción de tornillos no defectuosos en la muestra es inferior al 80 % (de hecho, es del 70 %), se trata de una situación que podría ser atribuíble al azar con una probabilidad de casi 6 %, aun cuando la proporción real (de la población) fuese la declarada por la fábrica. Puede parecer suficiente evidencia para iniciar una reclamación, pero habitualmente querríamos tener una mayor seguridad para poder afirmarlo.

Por lo tanto, y con el ánimo de hacer el test más potente, extraemos una muestra adicional (también aleatoria) de 50 tornillos más, de los cuales resultan en esta ocasión 13 defectuosos. Tenemos ahora una muestra total de 100, de los cuales $100 - (15 + 13) = 72$ no resultaron defectuosos:

```
prop.test(72, 100, p = 0.8, alternative = "less")

##
##  1-sample proportions test with continuity correction
##
## data:  72 out of 100, null probability 0.8
## X-squared = 3.516, df = 1, p-value = 0.0304
## alternative hypothesis: true p is less than 0.8
## 95 percent confidence interval:
##  0.0000 0.7918
## sample estimates:
##    p
## 0.72
```

Podemos observar cómo aunque la diferencia entre la proporción declarada por la fábrica y la que observamos en la muestra se ha reducido, ahora tenemos mayor certeza para afirmar que la hipótesis nula ($H_0 : P \geq 0{,}8$) es probablemente falsa.

Enviamos, por lo tanto, una reclamación a la fábrica de tornillos. La respuesta no se hace esperar y afirma que el departamento de control de calidad de la fábrica extrajo su propia muestra de 100 tornillos, de la cual resultaron 79 no defectuosos. Sorprendidos, pensamos que quizá la muestra extraída en la fábrica provenga de un lote (población) diferente, con una diferente proporción de defectos. Tratamos de comprobarlo:

```
prop.test(c(72, 79), c(100, 100), alternative = "two.sided")

##
##  2-sample test for equality of proportions with
##  continuity correction
##
## data:  c(72, 79) out of c(100, 100)
## X-squared = 0.9731, df = 1, p-value = 0.3239
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.19882  0.05882
## sample estimates:
## prop 1 prop 2
##   0.72   0.79
```

A la vista de los resultados del test, no podemos afirmar que se trate de dos lotes con diferente proporción de defectos. Pero ahora tenemos una muestra todavía más potente: 151 tornillos no defectuosos de un total de 200:

```
prop.test(151, 200, p = 0.8, alternative = "less")

##
##  1-sample proportions test with continuity correction
##
## data:  151 out of 200, null probability 0.8
## X-squared = 2.258, df = 1, p-value = 0.06647
## alternative hypothesis: true p is less than 0.8
```

```
## 95 percent confidence interval:
##  0.0000 0.8037
## sample estimates:
##     p
## 0.755
```

¿Qué podemos concluir a la vista de los resultados?

Práctica 3

Pruebas χ^2

3.1. Obtención de los datos

Los datos a utilizar en esta práctica proceden del paquete MASS. Por lo general, este es un paquete incluido en la instalación por defecto de R, de modo que no es necesario instalarlo.[1] Lo que sí precisamos hacer es cargarlo, para después cargar los datos de ejemplo *caith*:

```
library(MASS)
data(caith)
```

Los datos corresponden a 5387 personas del condado escocés de Caithness, clasificadas por su color de ojos y color de pelo. Para obtener más información sobre los datos y su origen, podemos invocar la ayuda:

```
help(caith)
```

Es un buen hábito explorar primero la estructura en la que se nos proporcionan los datos. Podemos hacer esto pidiendo a R que nos indique la clase de objeto en la que están almacenados, o su estructura:

```
class(caith)

## [1] "data.frame"

str(caith)

## 'data.frame': 4 obs. of  5 variables:
##  $ fair  : int  326 688 343 98
##  $ red   : int  38 116 84 48
##  $ medium: int  241 584 909 403
##  $ dark  : int  110 188 412 681
##  $ black : int  3 4 26 85
```

Como se puede observar, los datos están almacenados en formato de *data frame*, lo que es poco usual teniendo en cuenta el tipo de información que contienen (probablemente sería preferible un objeto *table*, más apropiado para una tabla de contingencia

[1] En caso de ser preciso podríamos hacerlo con *install.packages("MASS")*.

como la que nos ocupa). En todo caso, los nombres de los niveles de las dos variables no resultan demasiado intuitivos en su versión inglesa, de modo que podemos cambiarlos. Dado que los datos están guardados en forma de *data frame*, los nombres de los niveles están almacenados como nombres de fila y columna.

```
# Cambiamos los niveles de color de pelo (nombres de
# columna)
colnames(caith) <- c("rubio", "pelirrojo", "castaño", "moreno",
    "negro")
# Cambiamos los niveles de color de ojos (nombres de fila)
rownames(caith) <- c("azules", "claros", "castaños", "oscuros")
```

Podemos visualizar la tabla mediante el simple procedimiento de invocar su nombre. Es de destacar que, aunque se trate de un objeto tipo *data frame*, su aspecto es idéntico al que tendría una *table*.

```
caith
```

```
##          rubio pelirrojo castaño moreno negro
## azules     326        38     241    110     3
## claros     688       116     584    188     4
## castaños   343        84     909    412    26
## oscuros     98        48     403    681    85
```

3.2. Análisis

Tratándose de una tabla de contingencia, puede que visualizar las frecuencias relativas (respecto del total de la tabla) o marginales (respecto del total de fila o columna), sea más informativo que las frecuencias absolutas que acabamos de visualizar. Para ello si tendremos que convertir el objeto *caith* a un objeto de tipo *table*, a través de un objeto tipo matriz:

```
caith.table <- as.table(as.matrix(caith))
class(caith.table)
```

```
## [1] "table"
```

```
# Frecuencias relativas
prop.table(caith.table)
```

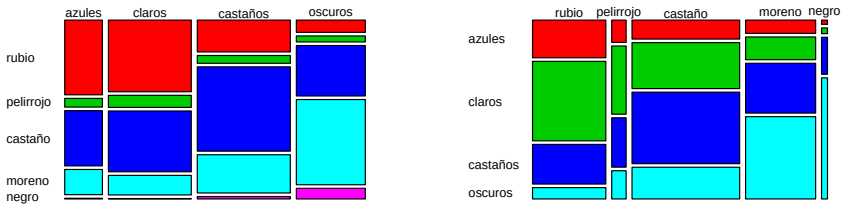
```
##              rubio pelirrojo   castaño    moreno     negro
## azules   0.0605161 0.0070540 0.0447373 0.0204195 0.0005569
## claros   0.1277149 0.0215333 0.1084091 0.0348988 0.0007425
## castaños 0.0636718 0.0155931 0.1687396 0.0764804 0.0048264
## oscuros  0.0181919 0.0089103 0.0748097 0.1264154 0.0157787
```

```
# Frecuencias marginales por filas
prop.table(caith.table, 1)
```

```
##             rubio pelirrojo  castaño   moreno    negro
## azules   0.454039  0.052925 0.335655 0.153203 0.004178
## claros   0.435443  0.073418 0.369620 0.118987 0.002532
## castaños 0.193348  0.047351 0.512401 0.232244 0.014656
## oscuros  0.074525  0.036502 0.306464 0.517871 0.064639

# Frecuencias marginales por columna
prop.table(caith.table, 2)

##             rubio pelirrojo castaño  moreno   negro
## azules   0.22405   0.13287 0.11277 0.07908 0.02542
## claros   0.47285   0.40559 0.27328 0.13515 0.03390
## castaños 0.23574   0.29371 0.42536 0.29619 0.22034
## oscuros  0.06735   0.16783 0.18858 0.48958 0.72034
```

Llegados a este punto es probable que nos resulte interesante visualizar los datos en un gráfico. La función estándar *plot* detecta automáticamente el tipo de objeto sobre el que se aplica (en este caso una *table*) para generar gráficos de barras como los siguientes. En estos, la anchura de cada columna es proporcional al total de observaciones en cada una, y cada barra se divide de modo proporcional a la frecuencia marginal de cada celda respecto de su columna. Podemos fijarnos en que, para obtener el segundo gráfico, aplicamos la misma función sobre la transpuesta de la tabla original.

```
plot(caith.table, col = 2:6, las = 1, main = "")
plot(t(caith.table), col = 2:5, las = 1, main = "")
```



En cualquiera de los dos gráficos podemos apreciar cómo determinados colores de pelo están más comúnmente asociados a determinados colores de ojos, o viceversa.

3.2.1. Contraste de independencia de caracteres

Asumiendo que los datos contenidos en la tabla puedan ser tomados como una muestra representativa de alguna población (¿quizá de la población de Escocia? ¿O simplemente del condado? En todo caso, no es algo que vayamos a decidir aquí), cabe preguntarse si la relación entre las dos variables (color de ojos y color de pelo) que podemos apreciar en los gráficos anteriores es exclusiva de la muestra (un efecto del azar) o por el contrario se trata de una asociación presente en la población original.

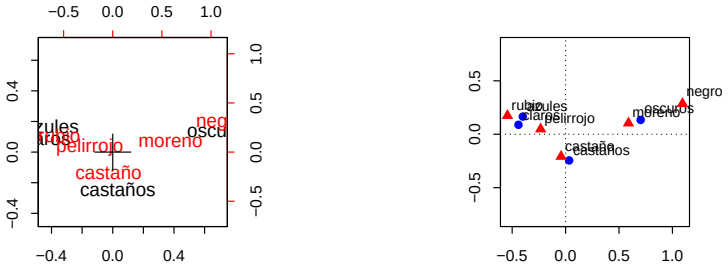
Para ello utilizaremos un contraste de χ^2, que en este caso funciona como un contraste de independencia de caracteres. Podemos observar cómo el nivel de significación resultante del contraste es prácticamente cero, lo que nos obliga a rechazar la hipótesis nula de independencia entre ambas variables.

```
chisq.test(caith)

##
##  Pearson's Chi-squared test
##
## data:  caith
## X-squared = 1240, df = 12, p-value < 2.2e-16
```

3.2.2. Análisis de correspondencias

Aunque en los gráficos anteriores se puede apreciar relativamente bien qué niveles de las dos variables van normalmente asociados (pelo rubio y ojos azules, por ejemplo), puede resultar útil realizar un análisis de correspondencias. Entre las varias posibilidades que tenemos para hacerlo podemos citar la función *corresp* disponible en el paquete MASS o la función *ca* en el paquete del mismo nombre.

```
plot(corresp(caith, nf = 2))
library(ca)
plot(ca(caith))
```



En cualquiera de los dos gráficos podemos observar qué niveles aparecen más frecuentemente asociados en función de su mayor proximidad en el gráfico.

Práctica 4

Modelos de regresión lineal

4.1. Obtención de los datos

En esta práctica emplearemos los datos de ejemplo *clouds*, relativos a un experimento de generación de lluvia artificial, disponibles en el paquete HSAUR. Para poder acceder a los datos es preciso tener instalado previamente el paquete, mediante la orden

```
install.packages("HSAUR")
```

O simplemente, si trabajamos en RStudio, yendo a *Packages→Install Packages*.

Una vez instalado el paquete, o si ya lo teníamos instalado, podemos cargarlo mediante la orden:

```
library(HSAUR)

## Loading required package: tools
```

o activando la pestaña correspondiente en RStudio. A continuación podemos cargar los datos con la orden[1]

```
data(clouds)
```

Para conocer algo más sobre el origen, estructura y contenido de los datos, se puede invocar la ayuda de R:

```
help(clouds)  # O, alternativamente, '?clouds'.
```

Como siempre, es buena idea comenzar explorando el número y tipo de variables disponibles en la tabla, solicitando a R información sobre la estructura del *data frame*:

```
str(clouds)

## 'data.frame': 24 obs. of  7 variables:
##  $ seeding   : Factor w/ 2 levels "no","yes": 1 2 2 1 2 1 1 1 1 2 ...
##  $ time      : int  0 1 3 4 6 9 18 25 27 28 ...
```

[1] Si no hubiera sido posible instalar el paquete HSAUR, es posible cargar los datos a partir del fichero *clouds.csv* disponible en la plataforma docente.

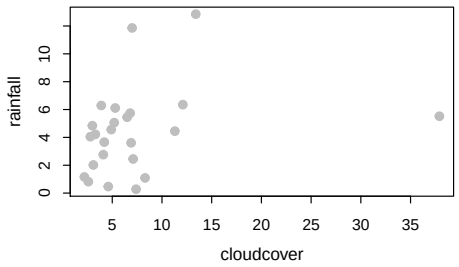
```
##  $ sne       : num  1.75 2.7 4.1 2.35 4.25 1.6 1.3 3.35 2.85 2.2 ...
##  $ cloudcover: num  13.4 37.9 3.9 5.3 7.1 6.9 4.6 4.9 12.1 5.2 ...
##  $ prewetness: num  0.274 1.267 0.198 0.526 0.25 ...
##  $ echomotion: Factor w/ 2 levels "moving","stationary": 2 1 2 1 1 2 1 1 1 1 ...
##  $ rainfall  : num  12.85 5.52 6.29 6.11 2.45 ...
```

Es un buen hábito comprobar en la descripción de la estructura si todas las variables han sido identificadas por R con la clase de información correcta. Es decir, si las variables categóricas han sido correctamente asignadas como *factor* o si las variables numéricas aparecen como tales. En caso contrario, es posible que algo haya fallado en el proceso de importación de los datos.

4.2. Ajuste del modelo

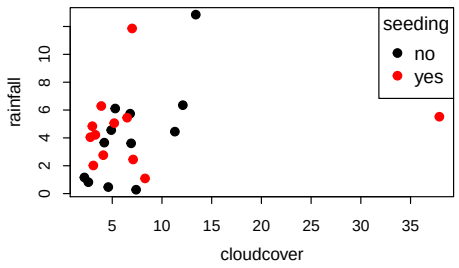
El primer paso consistirá en explorar de modo gráfico la relación entre las variables numéricas *cloudcover* y *rainfall*. El modo lógico de realizarlo es a través de un diagrama de dispersión:

```
plot(rainfall ~ cloudcover, data = clouds, pch = 19, col = "grey")
# El mismo gráfico también se puede generar con:
# plot(clouds$cloudcover,clouds$rainfall)
```



En todo caso, en el gráfico anterior están mezclados los eventos de lluvia artificial y no artificial, que podemos separar por colores empleando la variable *seeding* para establecer los colores de cada punto:

```
plot(rainfall ~ cloudcover, data = clouds, pch = 19, col = seeding)
legend("topright", legend = levels(clouds$seeding), col = 1:2,
    pch = 19, title = "seeding")
```



Si por alguna razón sólo nos interesase establecer la relación entre las dos variables (*cloudcover* y *rainfall*) en los casos en los que no se ha provocado la lluvia artificial (la

variable *seeding* toma el valor *no*), podemos proceder a realizar una selección. En este caso, lo haremos por exclusión, seleccionando el conjunto contrario:

```
artif <- which(clouds$seeding == "yes")
```

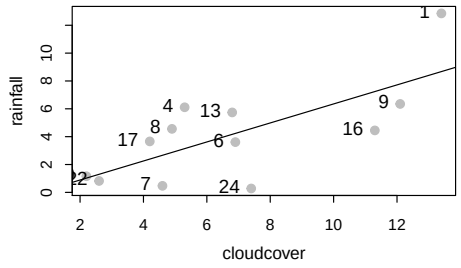
Observemos que en la orden anterior pedimos a R que nos devuelva los números de fila de las observaciones que cumplen una determinada condición (en este caso, que la variable *seeding* tome el valor *si*).

De este modo, podemos ajustar un modelo lineal para estos datos exclusivamente. Obsérvese que empleamos un signo negativo en la especificación del parámetro *subset*, para indicar que deseamos ajustar el modelo sobre todas las observaciones excepto las que tengan esos números de linea:

```
modelo1 <- lm(rainfall ~ cloudcover, data = clouds, subset = -artif)
# Invocar el nombre del modelo una vez ajustado nos
# proporciona información básica:
modelo1

##
## Call:
## lm(formula = rainfall ~ cloudcover, data = clouds, subset = -artif)
##
## Coefficients:
## (Intercept)   cloudcover
##      -0.478        0.683

# Y utilizar el comando summary nos proporciona información
# detallada
summary(modelo1)

##
## Call:
## lm(formula = rainfall ~ cloudcover, data = clouds, subset = -artif)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.296 -1.625 -0.171  1.603  4.176
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.478      1.627   -0.29   0.7748
## cloudcover     0.683      0.212    3.22   0.0092 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.59 on 10 degrees of freedom
## Multiple R-squared: 0.509,Adjusted R-squared: 0.459
## F-statistic: 10.3 on 1 and 10 DF,  p-value: 0.00922
```

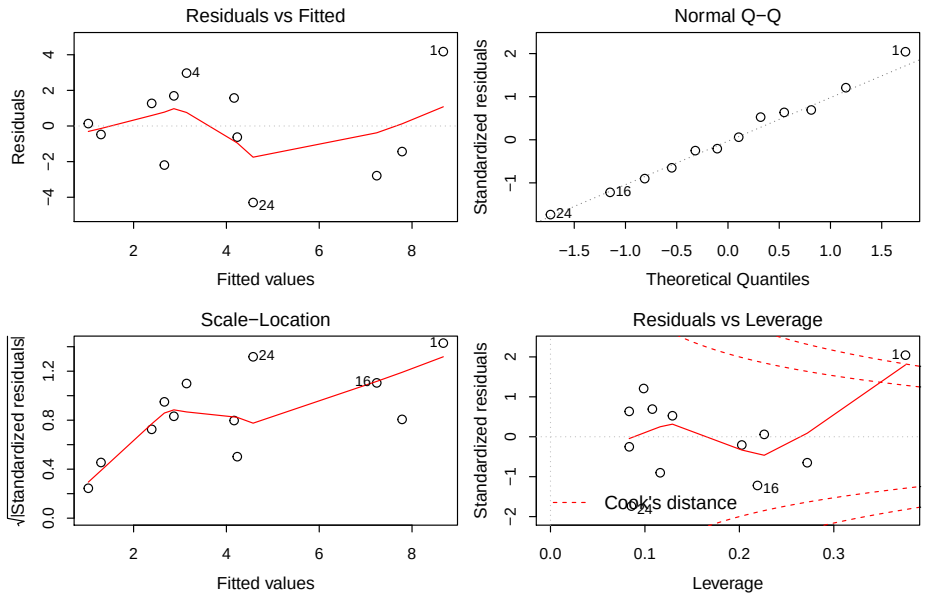
Podemos representar la recta definida por el modelo con la siguiente expresión:

```
plot(rainfall ~ cloudcover, data = clouds, pch = 19, col = "grey",
    subset = -artif)
text(x = clouds$cloudcover[-artif], y = clouds$rainfall[-artif],
    labels = rownames(clouds[-artif, ]), pos = 2)
abline(modelo1)
```



Una vez obtenido el modelo, debemos comprobar que los supuestos de partida de la regresión lineal se cumplen. Para ello, empleamos la orden *plot*, lo que nos irá dando una serie de cuatro gráficos:

```
plot(modelo1)
```



En el primero de los gráficos observamos que el valor medio de los residuos (indicado por la línea de ajuste local en color rojo) es próximo a cero a lo largo del ajuste, lo que sugiere que el residuo es independiente de los valores ajustados.

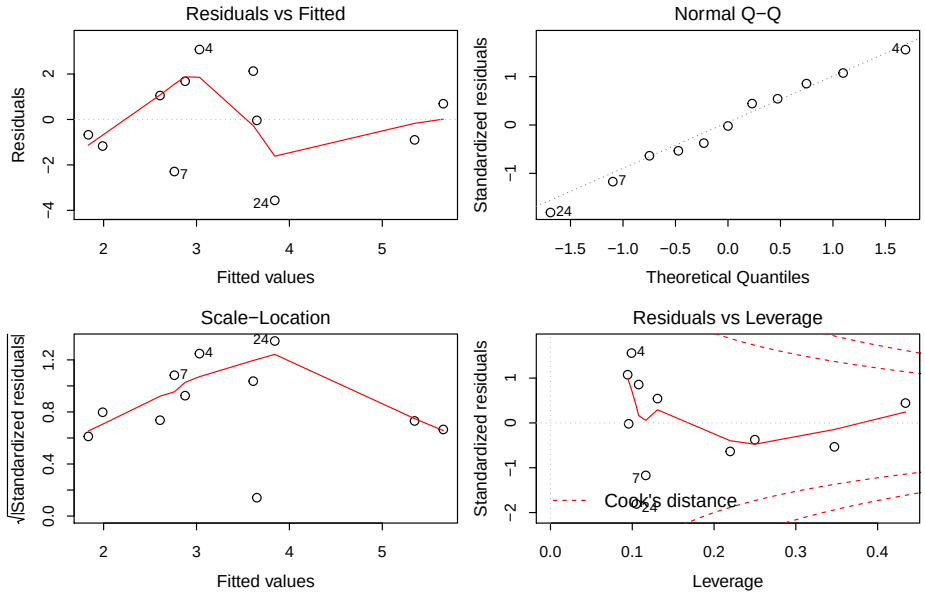
El segundo de los gráficos presenta un diagrama de cuantiles para los residuos de ajuste. Podemos observar que los puntos se sitúan razonablemente cerca de la línea diagonal, lo que permite suponer que los residuos siguen una distribución normal.

El tercero de los gráficos muestra la variabilidad de los residuos en función de los valores ajustados. Como podemos observar en este caso, los valores se hacen más altos hacia la derecha del gráfico, lo que sugiere que la varianza de los residuos no es independiente de los valores ajustados y que probablemente no se cumpla la condición de homocedasticidad.

Finalmente, el cuarto gráfico permite localizar puntos de apalancamiento utilizando la distancia de Cook como referencia. En este caso, podemos observar cómo la observa-

ción número 1 ejerce un considerable apalancamiento sobre el modelo. Si observamos el gráfico generado más atrás, veremos que la observación 1 es la situada en la esquina superior derecha. Por su posición alejada del conjunto de las observaciones, resulta razonable pensar que ejerce una considerable influencia sobre la forma del modelo. En el supuesto de que pensemos que sería recomendable eliminar esta observación, podríamos ajustar un nuevo modelo:

```
modelo1b <- lm(rainfall ~ cloudcover, data = clouds, subset = -c(1,
    artif))
summary(modelo1b)

##
## Call:
## lm(formula = rainfall ~ cloudcover, data = clouds, subset = -c(1,
##     artif))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.562 -1.034 -0.039  1.368  3.078
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.986      1.430    0.69    0.508
## cloudcover     0.386      0.207    1.86    0.095 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.08 on 9 degrees of freedom
## Multiple R-squared: 0.279,Adjusted R-squared: 0.199
## F-statistic: 3.48 on 1 and 9 DF,  p-value: 0.0951

plot(modelo1b)
```



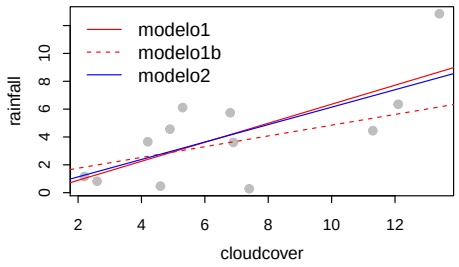
Podemos observar como en este caso la eliminación de la observación número 1 del conjunto de puntos a utilizar en el ajuste ha hecho que mejore el cumplimiento del supuesto de homocedasticidad. También es apreciable el cambio en los coeficientes estimados, particularmente en la pendiente.

Finalmente, también es posible recurrir a un modelo robusto para tratar de reducir la influencia de la observación 1 sin tener que eliminarla. Podemos hacerlo, por ejemplo, con la ayuda de la función *rlm* del paquete MASS:

```
library(MASS)
modelo2 <- rlm(rainfall ~ cloudcover, data = clouds, subset = -artif)
summary(modelo2)

##
## Call: rlm(formula = rainfall ~ cloudcover, data = clouds, subset = -artif)
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.233 -1.402 -0.344  1.605  4.581
##
## Coefficients:
##             Value  Std. Error t value
## (Intercept) -0.119  1.732     -0.069
## cloudcover   0.626  0.226      2.769
##
## Residual standard error: 2.38 on 10 degrees of freedom
```

Podemos representar gráficamente los tres modelos. ¿Cuál elegiríamos?

```
plot(rainfall ~ cloudcover, data = clouds, subset = -artif, pch = 19,
    col = "grey")
abline(modelo1, col = "red")
abline(modelo1b, col = "red", lty = 2)
abline(modelo2, col = "blue")
legend("topleft", legend = c("modelo1", "modelo1b", "modelo2"),
    col = c("red", "red", "blue"), lty = c(1, 2, 1), bty = "n")
```



Práctica 5

Análisis de varianza

5.1. Obtención de los datos

Al igual que en la práctica 4, en este caso emplearemos un conjunto de datos de ejemplo disponible en el paquete HSAUR. Si no hemos instalado previamente este paquete podemos hacerlo como se describe en aquella práctica. Al igual que en ese caso, procederemos a cargar el paquete y los datos con las órdenes:[1]

```
library(HSAUR)
data(weightgain)
```

En este caso, los datos proceden de un experimento relacionado con el engorde de ratas de laboratorio, empleando diferentes tipos de dieta. En particular, los investigadores querían analizar el efecto de dos factores: el contenido en proteínas (variable *type*) y el origen de las proteínas (variable *source*). Podemos obtener más información sobre estos detalles mediante la ayuda:

```
help(weightgain)
```

Asimismo, es recomendable verificar que la estructura de los datos es la esperada, y que las variables aparecen indentificadas correctamente.

```
str(weightgain)

## 'data.frame': 40 obs. of  3 variables:
##  $ source    : Factor w/ 2 levels "Beef","Cereal": 1 1 1 1 1 1 1 1 1 1 ...
##  $ type      : Factor w/ 2 levels "High","Low": 2 2 2 2 2 2 2 2 2 2 ...
##  $ weightgain: int  90 76 90 64 86 51 72 90 95 78 ...
```

Otras formas posibles para comprobar la estructura interna de los datos es solicitar un encabezado de la tabla, o un resumen:

```
head(weightgain)

##   source type weightgain
## 1   Beef  Low         90
```

[1] Los datos están también disponibles en la plataforma docente, en el fichero *weightgain.csv*.

```
## 2   Beef  Low         76
## 3   Beef  Low         90
## 4   Beef  Low         64
## 5   Beef  Low         86
## 6   Beef  Low         51

summary(weightgain)

##     source     type      weightgain
##  Beef  :20   High:20   Min.   : 51.0
##  Cereal:20   Low :20   1st Qu.: 75.5
##                        Median : 88.5
##                        Mean   : 87.2
##                        3rd Qu.: 98.0
##                        Max.   :118.0
```

5.2. Ajuste de los modelos

En este caso, comprobaremos en primer lugar si el diseño es equilibrado. Es decir, si el número de observaciones es el mismo para cada combinación de los niveles de los dos factores. Podemos emplear para ello una tabla de contingencia de las dos variables categóricas:

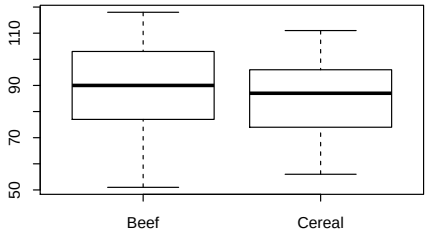
```
table(weightgain$source, weightgain$type)

##
##          High Low
##   Beef     10  10
##   Cereal   10  10
```

Imaginemos por un momento que sólo nos interesa analizar la influencia de uno de los factores, por ejemplo la variable *source*, sobre el incremento de peso de los animales. Al tratarse de un factor con sólo dos niveles, podríamos resolver el problema mediante un contraste de hipótesis para la diferencia de medias entre dos poblaciones distintas:

```
t.test(weightgain ~ source, data = weightgain)

##
##  Welch Two Sample t-test
##
## data:  weightgain by source
## t = 0.9057, df = 36.99, p-value = 0.3709
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -5.814 15.214
## sample estimates:
##   mean in group Beef mean in group Cereal
##                 89.6                 84.9
```

El resultado del test sugiere que la diferencia entre las medias de las dos muestras (89.6 y 84.9) no es suficientemente grande como para pensar que se corresponde con una diferencia en las medias de las dos poblaciones. De hecho, el p-valor asociado al test es bastante elevado. Una comprobación gráfica nos puede ayudar a entender este resultado: como podemos apreciar en el gráfico siguiente, los valores de las dos muestras se solapan considerablemente.

```
boxplot(weightgain$weightgain ~ weightgain$source)
```



5.2.1. Anova para un factor

Obtendremos un resultado equivalente al anterior si realizamos un análisis de varianza para un único factor. Podemos observar en este caso como la suma de residuos que podemos atribuir al factor en estudio (221) es un valor muy reducido respecto de la suma de residuos que no podemos explicar (10233), lo que explica el valor relativamente alto del nivel de significación del test.

```
anova1 <- aov(weightgain ~ source, data = weightgain)
summary(anova1)

##             Df Sum Sq Mean Sq F value Pr(>F)
## source       1    221     221    0.82   0.37
## Residuals   38  10233     269
```

Podemos comprobar los supuestos de normalidad e igualdad de varianza entre grupos mediante un test de Shapiro-Wilks y un test F de igualdad de varianzas, respectivamente:

```
# Podemos comprobar la normalidad del conjunto de
# observaciones:
shapiro.test(weightgain$weightgain)

##
##  Shapiro-Wilk normality test
##
## data:  weightgain$weightgain
## W = 0.9858, p-value = 0.8882

# O de cada uno de los grupos por separado:
tapply(weightgain$weightgain, weightgain$source, shapiro.test)
```

```
## $Beef
##
##  Shapiro-Wilk normality test
##
## data:  X[[1L]]
## W = 0.9804, p-value = 0.9391
##
##
## $Cereal
##
##  Shapiro-Wilk normality test
##
## data:  X[[2L]]
## W = 0.9713, p-value = 0.783
```

```
# Y la igualdad de varianzas:
var.test(weightgain ~ source, data = weightgain)
```

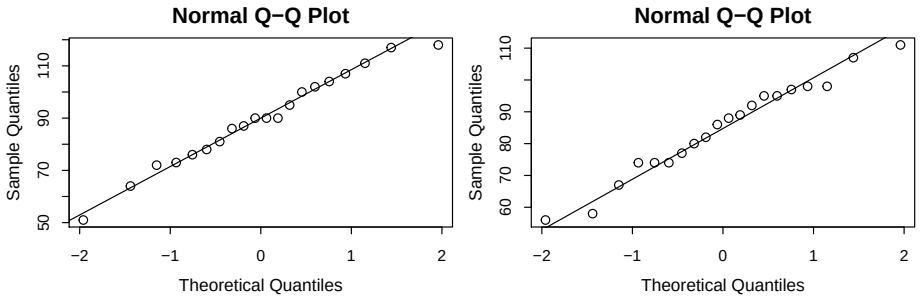
```
##
##  F test to compare two variances
##
## data:  weightgain by source
## F = 1.395, num df = 19, denom df = 19, p-value =
## 0.4746
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.5523 3.5254
## sample estimates:
## ratio of variances
##              1.395
```

```
bartlett.test(weightgain ~ source, data = weightgain)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  weightgain by source
## Bartlett's K-squared = 0.5114, df = 1, p-value =
## 0.4745
```

En el ejemplo anterior empleamos también un test de Bartlett, que es útil cuando el número de grupos a comparar es mayor de dos (situación en la que el test F no nos sirve). Los resultados de los diferentes tests permiten aceptar la hipótesis nula de normalidad en el test de Shapiro-Wilks, y la de igualdad de varianzas tanto en el test F como en el de Bartlett. En todo caso, la comprobación de normalidad puede ser realizada también (y es recomendable hacerlo para complementar el análisis numérico) mediante un diagrama de cuantiles:

```
qqnorm(weightgain$weightgain[weightgain$source == "Beef"])
qqline(weightgain$weightgain[weightgain$source == "Beef"])
qqnorm(weightgain$weightgain[weightgain$source == "Cereal"])
qqline(weightgain$weightgain[weightgain$source == "Cereal"])
```



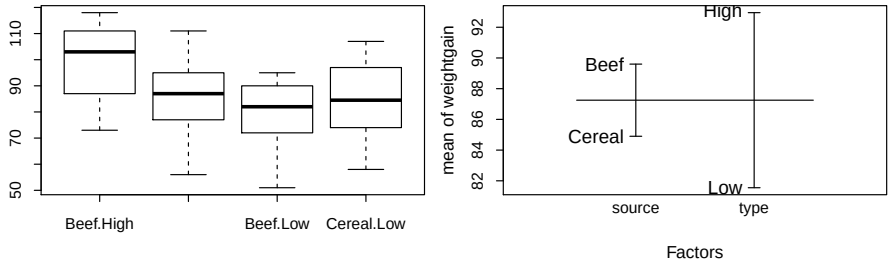
Aunque ya hemos visto que en este caso es razonable aceptar la hipótesis de homocedasticidad (igualdad de varianza entre grupos), aprovecharemos para introducir aquí el test de Welch, que sería el que deberíamos utilizar en caso de heterocedasticidad entre grupos:

```
oneway.test(weightgain ~ source, data = weightgain)

##
##  One-way analysis of means (not assuming equal
##  variances)
##
## data:  weightgain and source
## F = 0.8203, num df = 1.00, denom df = 36.99, p-value
## = 0.3709
```

5.2.2. Anova de dos factores

Comenzamos en este caso con un análisis gráfico mediante diagramas de cajas, y explorando los valores de la media y desviación típica de cada combinación de los niveles de los dos factores:

```
tapply(weightgain$weightgain, list(weightgain$source, weightgain$type),
    mean)

##         High  Low
## Beef   100.0 79.2
## Cereal  85.9 83.9

tapply(weightgain$weightgain, list(weightgain$source, weightgain$type),
    sd)

##         High   Low
## Beef   15.14 13.89
## Cereal 15.02 15.71

boxplot(weightgain ~ source + type, data = weightgain)
plot.design(weightgain)
```



El comando *plot.design* nos presenta en un mismo gráfico la variación en la media atribuible a cada uno de los dos factores en estudio. En este caso, podemos observar cómo la diferencia asociada al factor *type* es considerablemente mayor que la asociada al factor *source*. Para realizar un análisis de varianza para estos dos factores emplearemos la siguiente expresión:

```
anova2 <- aov(weightgain ~ source + type, data = weightgain)
anova2

## Call:
##    aov(formula = weightgain ~ source + type, data = weightgain)
##
## Terms:
##                   source type Residuals
## Sum of Squares       221 1300      8933
## Deg. of Freedom        1    1        37
##
## Residual standard error: 15.54
## Estimated effects may be unbalanced

summary(anova2)

##             Df Sum Sq Mean Sq F value Pr(>F)
## source       1    221     221    0.91  0.345
## type         1   1300    1300    5.38  0.026 *
## Residuals   37   8933     241
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

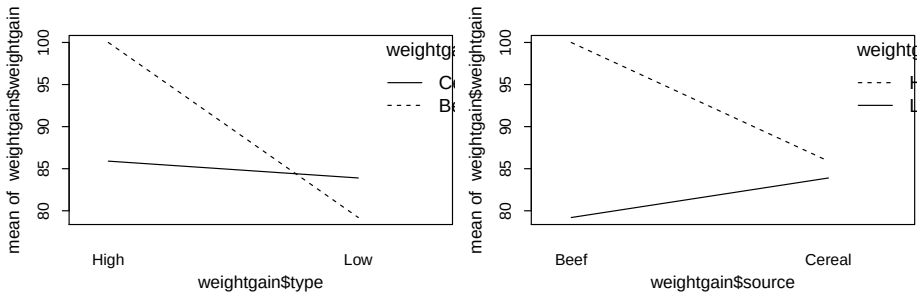
Podemos observar cómo el nivel de significación asociado a la variable *type* es menor que el umbral habitualmente utilizado del 5 %. En este caso, el análisis de dos factores nos indica que sólo podemos atribuir cambios en la media de la variable *weightgain* al factor *type*. Los cambios en la media de la variable *weightgain* que se producen en este conjunto de datos debido al factor *source* (observables en los gráficos anteriores) serían, por lo tanto, demasiado pequeños como para suponer la existencia de un efecto real debido a este factor, y fácilmente atribuíbles al azar.

En todo caso, hasta este momento, hemos utilizado un modelo de efectos principales (*main effects*) debidos a cada uno de los factores en estudio, pero queda por explorar la posible existencia de efectos de interacción entre ambos factores.

```
anova2b <- aov(weightgain ~ source * type, data = weightgain)
anova2b

## Call:
##    aov(formula = weightgain ~ source * type, data = weightgain)
##
## Terms:
##                  source type source:type Residuals
## Sum of Squares      221 1300         884      8049
## Deg. of Freedom       1    1           1        36
##
## Residual standard error: 14.95
## Estimated effects may be unbalanced

summary(anova2b)

##             Df Sum Sq Mean Sq F value Pr(>F)
## source       1    221     221    0.99  0.327
## type         1   1300    1300    5.81  0.021 *
## source:type  1    884     884    3.95  0.054 .
## Residuals   36   8049     224
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Podemos observar que para introducir la componente de interacción hemos utilizado la sintaxis *source*type* en lugar de *source+type*. Alternativamente, podríamos haber utilizado una mención explícita de los efectos principales y de interacción por separado: *source+type+source:type*. En los resultados de la Anova se puede apreciar cómo la componente de interacción está en el límite de ser considerada significativa por los estándares habituales (el nivel de significación es muy próximo al 5 %). Ello parece indicar que existe un cierto efecto de interacción entre los dos factores, que podemos explorar en los gráficos de interacción siguientes:

```
interaction.plot(weightgain$type, weightgain$source, weightgain$weightgain)
interaction.plot(weightgain$source, weightgain$type, weightgain$weightgain)
```



En este caso, probablemente el segundo de los dos gráficos de interacción es el más fácil de interpretar: ya habíamos dicho que es a la variable *type* a la que podemos atribuir un efecto sobre la media de la variable numérica *weightgain*, lo que se aprecia en en hecho de que la línea de puntos se encuentre siempre por encima de la línea continua en el segundo de los gráficos (de acuerdo con ello, los animales engordarían más cuando la

dieta es rica en proteínas que cuando es baja). No obstante, aunque no pudimos atribuir un efecto a la variable *source* por sí misma, esta parece influir en el comportamiento de la anterior: la diferencia entre dietas ricas y pobres en proteínas varía en función del origen de estas, de modo que parece ser mucho menor cuando la proteína procede de cereales y mayor cuando procede de carne.

Test de comparaciones múltiples

Finalmente, si deseamos conocer que grupos son diferentes entre sí, podemos someter los resultados de la Anova a un test de comparaciones múltiples, como por ejemplo:

```
TukeyHSD(anova2b, conf.level = 0.95)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = weightgain ~ source * type, data = weightgain)
##
## $source
##             diff    lwr  upr  p adj
## Cereal-Beef -4.7 -14.29 4.89 0.3269
##
## $type
##           diff    lwr   upr  p adj
## Low-High -11.4 -20.99 -1.81 0.0211
##
## $`source:type`
##                           diff    lwr   upr  p adj
## Cereal:High-Beef:High  -14.1 -32.11  3.91 0.1698
## Beef:Low-Beef:High     -20.8 -38.81 -2.79 0.0183
## Cereal:Low-Beef:High   -16.1 -34.11  1.91 0.0937
## Beef:Low-Cereal:High    -6.7 -24.71 11.31 0.7493
## Cereal:Low-Cereal:High  -2.0 -20.01 16.01 0.9905
## Cereal:Low-Beef:Low      4.7 -13.31 22.71 0.8953
```

El resultado del test nos indica un intervalo de confianza (que hemos establecido al 95 %) para la diferencia entre las medias de las cuatro combinaciones de los niveles de los dos factores (en este caso son las combinaciones de cuatro elementos, tomados de dos en dos: un total de seis posibles). Naturalmente, el intervalo está comprendido entre el límite inferior (*lower limit*, *lwr*) y el superior (*upper limit*, *upr*), y entenderemos que existen diferencias cuando este intervalo no contenga al valor 0. En este caso, el test sólo nos indica diferencias entre *Beef:Low* y *Beef:High*, lo que es coherente con los diagramas de cajas obtenidos al principio del subapartado, y con los gráficos de interacción. Es decir, el conjunto de pruebas que hemos hecho hasta el momento nos indica que sólo la variable *type* tiene un efecto apreciable sobre el incremento de peso, y sólo (este es el efecto de interacción) cuando la variable *source* (que por sí misma no tiene efectos) toma el valor *Beef*.

Práctica 6

Análisis de conglomerados

6.1. Obtención de los datos

En esta práctica utilizaremos el conjunto de datos *mtcars*, disponible en la instalación base de R. Los datos se refieren a un conjunto de 32 modelos de automóvil de los años 1973 y 1974, y sus características principales, medidas a través de un número de variables. Como siempre, cargaremos los datos y realizaremos una primera exploración. Cabe destacar que los nombres de fila contienen los nombres de los modelos de coche.

```
data(mtcars)
str(mtcars)
```

```
## 'data.frame': 32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0
##                   gear carb
## Mazda RX4            4    4
## Mazda RX4 Wag        4    4
## Datsun 710           4    1
```

```
## Hornet 4 Drive       3    1
## Hornet Sportabout    3    2
## Valiant              3    1
```

Si queremos información más detallada, podemos invocar la ayuda:

```
help(mtcars)
```

El objetivo de esta práctica será formar, a partir de los datos originales, grupos formados por modelos de automóvil con características homogéneas.

Comezamos cargando o conxunto de datos e explorando a estrutura da táboa. Como podemos ver, trátase de 32 observacións dun conxunto de 11 variables numéricas.

6.2. Análisis

6.2.1. Selección y estandarización

Como habremos podido observar en la descripción detallada de los datos y en la descripción de la estructura en la que están almacenados, aunque las 11 variables están codificadas como variables numéricas, una de ellas es en realidad una variable categórica: la variable *am* indica el tipo de transmisión, y está codificada con un 0 para la transmisión automática y un 1 para la transmisión manual. De igual modo, la variable *vs* se refiere al tipo de motor: el valor 0 indica un motor en V y el valor 1 indica un motor en línea.

Podemos explorar la correlación entre las 11 variables utilizando, en este caso, un coeficiente de correlación no lineal como el de Spearman. Como se puede apreciar en la tabla resultante, muchas de las variables están correlacionadas entre sí. Algunas de ellas, como por ejemplo el consumo (variable *mpg*), presentan valores elevados de correlación con prácticamente todas las restantes.

```
cor(mtcars, method = "spearman")
```

```
##          mpg     cyl    disp      hp     drat      wt
## mpg   1.0000 -0.9108 -0.9089 -0.8947  0.65146 -0.8864
## cyl  -0.9108  1.0000  0.9277  0.9018 -0.67888  0.8577
## disp -0.9089  0.9277  1.0000  0.8510 -0.68359  0.8977
## hp   -0.8947  0.9018  0.8510  1.0000 -0.52012  0.7747
## drat  0.6515 -0.6789 -0.6836 -0.5201  1.00000 -0.7504
## wt   -0.8864  0.8577  0.8977  0.7747 -0.75039  1.0000
## qsec  0.4669 -0.5724 -0.4598 -0.6666  0.09187 -0.2254
## vs    0.7066 -0.8138 -0.7237 -0.7516  0.44746 -0.5870
## am    0.5620 -0.5221 -0.6241 -0.3623  0.68657 -0.7377
## gear  0.5428 -0.5643 -0.5945 -0.3314  0.74482 -0.6761
## carb -0.6575  0.5801  0.5398  0.7334 -0.12522  0.4998
##          qsec      vs       am    gear     carb
## mpg   0.46694  0.7066  0.56201  0.5428 -0.65750
## cyl  -0.57235 -0.8138 -0.52207 -0.5643  0.58007
## disp -0.45978 -0.7237 -0.62407 -0.5945  0.53978
## hp   -0.66661 -0.7516 -0.36233 -0.3314  0.73338
```

```
## drat  0.09187  0.4475  0.68657  0.7448 -0.12522
## wt   -0.22540 -0.5870 -0.73771 -0.6761  0.49981
## qsec  1.00000  0.7916 -0.20333 -0.1482 -0.65872
## vs    0.79157  1.0000  0.16835  0.2827 -0.63369
## am   -0.20333  0.1683  1.00000  0.8077 -0.06437
## gear -0.14820  0.2827  0.80769  1.0000  0.11489
## carb -0.65872 -0.6337 -0.06437  0.1149  1.00000
```

Como consecuencia, podríamos optar por reducir el número de variables eligiendo aquellas que consideremos más importantes (en rigor, aquellas más importantes para el tipo de grupos que deseamos formar). En este caso optaremos por utilizar el conjunto original, sin eliminar ninguna.

Por otra parte, el resumen de los valores presentes en la tabla evidencia grandes diferencias en la escala de las variables que contiene. Por ejemplo, la cilindrada (*disp*) presenta valores entre 71 y 472 pulgadas cúbicas, mientras que la relación de transmisión en el diferencial (*drat*) simplemente oscila entre algo menos de 3 y algo menos de 5. Otras, incluso, sólo toman valores de 0 o 1 (*am* o *vs*). Para evitar que estas diferencias en la escala de medida, y su influencia sobre los resultados del análisis de conglomerados, trataremos de estandarizar las diferentes variables.

```
summary(mtcars)
```

```
##       mpg             cyl             disp       
##  Min.   :10.4   Min.   :4.00   Min.   : 71.1  
##  1st Qu.:15.4   1st Qu.:4.00   1st Qu.:120.8  
##  Median :19.2   Median :6.00   Median :196.3  
##  Mean   :20.1   Mean   :6.19   Mean   :230.7  
##  3rd Qu.:22.8   3rd Qu.:8.00   3rd Qu.:326.0  
##  Max.   :33.9   Max.   :8.00   Max.   :472.0  
##        hp             drat            wt      
##  Min.   : 52.0   Min.   :2.76   Min.   :1.51  
##  1st Qu.: 96.5   1st Qu.:3.08   1st Qu.:2.58  
##  Median :123.0   Median :3.69   Median :3.33  
##  Mean   :146.7   Mean   :3.60   Mean   :3.22  
##  3rd Qu.:180.0   3rd Qu.:3.92   3rd Qu.:3.61  
##  Max.   :335.0   Max.   :4.93   Max.   :5.42  
##       qsec             vs              am       
##  Min.   :14.5   Min.   :0.000   Min.   :0.000  
##  1st Qu.:16.9   1st Qu.:0.000   1st Qu.:0.000  
##  Median :17.7   Median :0.000   Median :0.000  
##  Mean   :17.8   Mean   :0.438   Mean   :0.406  
##  3rd Qu.:18.9   3rd Qu.:1.000   3rd Qu.:1.000  
##  Max.   :22.9   Max.   :1.000   Max.   :1.000  
##       gear            carb      
##  Min.   :3.00   Min.   :1.00  
##  1st Qu.:3.00   1st Qu.:2.00  
##  Median :4.00   Median :2.00  
##  Mean   :3.69   Mean   :2.81  
##  3rd Qu.:4.00   3rd Qu.:4.00  
##  Max.   :5.00   Max.   :8.00
```

Para estandarizar los valores de todas las variables emplearemos en este caso el procedimiento de restar a los valores de cada una la media y dividir por la desviación típica correspondiente. Usaremos para ello la orden *scale*, que realiza precisamente esta operación por defecto.[1]

```
mtcars2 <- scale(mtcars)
summary(mtcars2)
```

```
##       mpg               cyl               disp        
##  Min.   :-1.608   Min.   :-1.225   Min.   :-1.288  
##  1st Qu.:-0.774   1st Qu.:-1.225   1st Qu.:-0.887  
##  Median :-0.148   Median :-0.105   Median :-0.278  
##  Mean   : 0.000   Mean   : 0.000   Mean   : 0.000  
##  3rd Qu.: 0.450   3rd Qu.: 1.015   3rd Qu.: 0.769  
##  Max.   : 2.291   Max.   : 1.015   Max.   : 1.947  
##        hp              drat              wt        
##  Min.   :-1.381   Min.   :-1.565   Min.   :-1.742  
##  1st Qu.:-0.732   1st Qu.:-0.966   1st Qu.:-0.650  
##  Median :-0.345   Median : 0.184   Median : 0.110  
##  Mean   : 0.000   Mean   : 0.000   Mean   : 0.000  
##  3rd Qu.: 0.486   3rd Qu.: 0.605   3rd Qu.: 0.401  
##  Max.   : 2.747   Max.   : 2.494   Max.   : 2.255  
##       qsec                vs                am        
##  Min.   :-1.8740   Min.   :-0.868   Min.   :-0.814  
##  1st Qu.:-0.5351   1st Qu.:-0.868   1st Qu.:-0.814  
##  Median :-0.0776   Median :-0.868   Median :-0.814  
##  Mean   : 0.0000   Mean   : 0.000   Mean   : 0.000  
##  3rd Qu.: 0.5883   3rd Qu.: 1.116   3rd Qu.: 1.190  
##  Max.   : 2.8268   Max.   : 1.116   Max.   : 1.190  
##       gear              carb        
##  Min.   :-0.932   Min.   :-1.122  
##  1st Qu.:-0.932   1st Qu.:-0.503  
##  Median : 0.424   Median :-0.503  
##  Mean   : 0.000   Mean   : 0.000  
##  3rd Qu.: 0.424   3rd Qu.: 0.735  
##  Max.   : 1.779   Max.   : 3.212
```

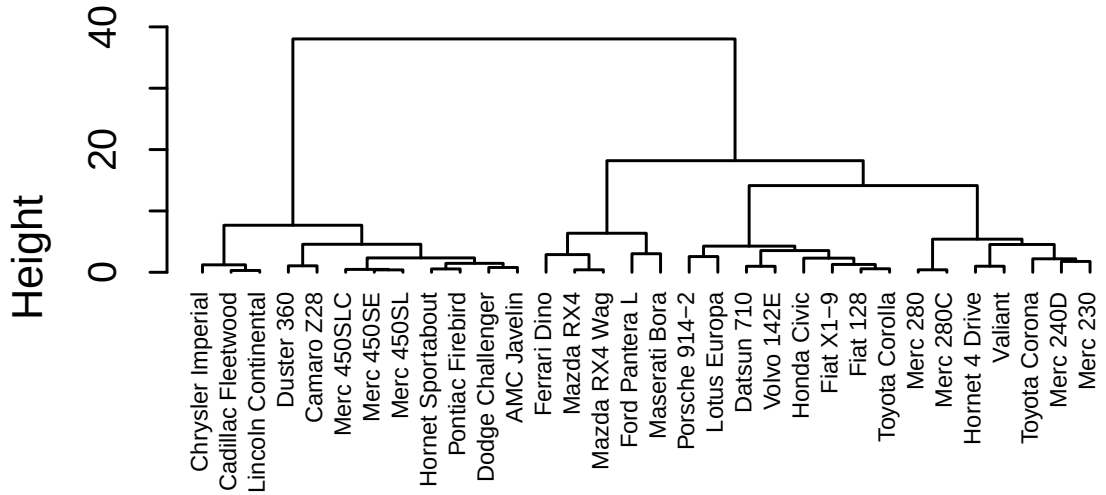
Podemos observar como los valores de las variables estandarizadas oscilan ahora en casi todos los casos entre -2 y 2, de modo que ninguna de ellas debiera tener más influencia de la debida sobre los resultados del análisis.

6.2.2. Métodos de agrupamiento jerárquico

El primer paso para utilizar un método de agrupamiento jerárquico consiste en obtener una matriz de distancias entre observaciones, en el espacio de n dimensiones (11 en este caso) en el que trabajamos. Existen diferentes modos de medir la distancia (euclídea, manhattan, canberra...), pero emplearemos en este caso la distancia euclídea. Sobre ella realizaremos el agrupamiento, para lo que también existen diferentes métodos posibles. Eligiremos en este caso el método de Ward.

[1] La orden *scale* permite especificar otras medidas de posición y dispersión.

```
distancias <- dist(mtcars2)
agrupam <- hclust(distancias, method = "ward")
plot(agrupam, hang = -1, cex = 0.5, main = "", xlab = "", sub = "")
```



En el dendrograma resultante se puede apreciar cómo los diferentes modelos de coche aparecen agrupados en función de su mayor o menor cercanía: cuanto más alto el nodo del que cuelgan dos modelos, menor es la similaridad entre ellos. La decisión de formar un determinado número de grupos nos corresponde a nosotros, de todos modos. En este caso, en el dendrograma podemos apreciar dos grupos claramente diferenciados, y en uno de ellos (situado a la derecha) podríamos formar hasta tres subgrupos más.

Tomemos por lo tanto la decisión de formar cuatro grupos. El resultado de esta última operación es un vector numérico con nombres, en el que el valor numérico corresponde al número de grupo (del 1 al 4 en este caso).

```
grupos1 <- cutree(agrupam, k = 4)
print(grupos1)
```

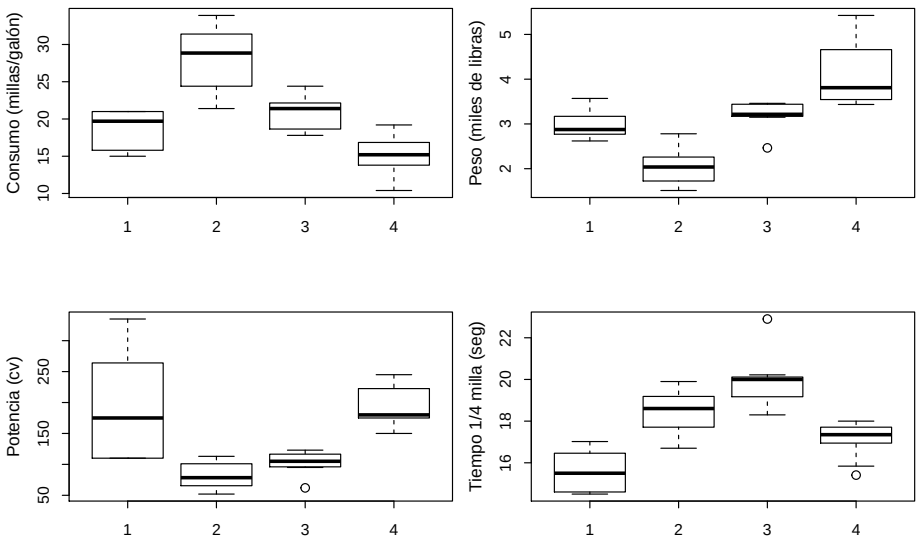
```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710
##                   1                   1                   2
##      Hornet 4 Drive   Hornet Sportabout             Valiant
##                   3                   4                   3
##          Duster 360           Merc 240D            Merc 230
##                   4                   3                   3
##           Merc 280           Merc 280C          Merc 450SE
##                   3                   3                   4
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood
##                   4                   4                   4
## Lincoln Continental   Chrysler Imperial           Fiat 128
```

```
##                   4                   4                   2
##         Honda Civic      Toyota Corolla       Toyota Corona
##                   2                   2                   3
##    Dodge Challenger         AMC Javelin          Camaro Z28
##                   4                   4                   4
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2
##                   4                   2                   2
##        Lotus Europa      Ford Pantera L        Ferrari Dino
##                   2                   1                   1
##       Maserati Bora          Volvo 142E
##                   1                   2
```

Para realizar una interpretación de los grupos formados podemos, por ejemplo, calcular los valores medios de cada grupo para cada variable. Por ejemplo, para calcular los valores medios de consumo y peso en cada grupo podríamos utilizar la orden *tapply*. Pero probablemente es más intuitivo utilizar diagramas de cajas como los de las siguientes figuras:

```
tapply(mtcars$mpg, INDEX = grupos1, mean)

##     1     2     3     4
## 18.50 28.07 20.74 15.05

tapply(mtcars$wt, INDEX = grupos1, mean)

##     1     2     3     4
## 3.001 2.042 3.194 4.104
```

```
boxplot(mpg ~ grupos1, data = mtcars, ylab = "Consumo (millas/galón)")
boxplot(wt ~ grupos1, data = mtcars, ylab = "Peso (miles de libras)")
boxplot(hp ~ grupos1, data = mtcars, ylab = "Potencia (cv)")
boxplot(qsec ~ grupos1, data = mtcars, ylab = "Tiempo 1/4 milla (seg)")
```



6.2.3. K-medias

Podemos realizar otro tipo de agrupamiento, en este caso mediante el uso del algoritmo k-medias, que sólo requiere en este caso que indiquemos el número de grupos que deseamos crear. Igual que en el caso anterior, formaremos grupos partiendo de las variables estandarizadas.

```
grupos2 <- kmeans(mtcars2, centers = 4)
print(grupos2$cluster)
```

```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710
##                   2                   2                   3
##      Hornet 4 Drive   Hornet Sportabout             Valiant
##                   4                   1                   4
##          Duster 360           Merc 240D            Merc 230
##                   1                   4                   4
##            Merc 280           Merc 280C          Merc 450SE
##                   4                   4                   1
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood
##                   1                   1                   1
## Lincoln Continental   Chrysler Imperial            Fiat 128
##                   1                   1                   3
##         Honda Civic      Toyota Corolla       Toyota Corona
##                   3                   3                   4
##    Dodge Challenger         AMC Javelin          Camaro Z28
##                   1                   1                   1
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2
##                   1                   3                   3
##        Lotus Europa      Ford Pantera L        Ferrari Dino
##                   3                   2                   2
##       Maserati Bora          Volvo 142E
##                   2                   3
```

Sobre esta nueva división en grupos podríamos (de hecho, deberíamos) realizar una interpretación gráfica y numérica como la del subapartado anterior. En todo caso, lo que haremos será comparar el grado de coincidencia que existe entre los resultados de los dos métodos. No hay necesariamente nada de particular en que diferentes métodos de agrupamiento (incluídos diferentes algoritmos de agrupamiento jerárquico, utilizados sobre matrices de distancias calculadas de diferentes formas) coincidan o no. De hecho, Es posible que en un caso real tengamos que probar diferentes combinaciones para encontrar aquella que tenga más sentido en el contexto de investigación en el que estemos trabajando.

```
table(grupos1, grupos2$cluster)
```

```
##
## grupos1  1  2  3  4
##       1  0  5  0  0
##       2  0  0  8  0
##       3  0  0  0  7
##       4 12  0  0  0
```

Como se puede observar, los dos métodos utilizados en este ejercicio coinciden sólo en parte. Por ejemplo, de los 11 coches clasificados en el grupo 4 en el primer caso (algoritmo de Ward sobre distancia euclídea), 8 aparecen clasificados en el grupo 2 formado por k-medias mientras que otros 3 aparecen el el grupo 3 junto con otras 4 observaciones.