Módulo 2 Análisis estadístico básico con R

Descripción estadística de las variables de un conjunto de datos

- Introducción

En este módulo se mostrará cómo utilizar el análisis estadístico para explorar la información contenida en los datos, lo que los científicos de datos denominan "análisis exploratorio de datos" (AED). El AED consta de tres etapas:

1. Generar preguntas sobre los datos
2. Buscar respuestas, resumiendo y modelizando los datos
3. Utilizar lo aprendido anteriormente para replantear nuestras preguntas de investigación o realizar nuevas

Durante el AED debemos sentirnos "libres" para investigar cada idea que se nos ocurra. Algunas de estas ideas se desarrollarán y otras nos llevarán a callejones sin salida.

El AED es la parte más importante del análisis de nuestros datos, incluso si las preguntas ya son conocidas, pues nos permite un primer contacto con la calidad y posibilidades de nuestros datos. No olvidar que la limpieza de datos es otra utilidad del AED.

En esta primera parte utilizaremos funciones del paquete base, en la parte final excepcionalmente utilizaremos algunos paquetes para táreas específicas.

- Las Tablas de Frecuencias

Herramienta para resumir, de forma ordenada, un conjunto de datos estadísticos. Presenta la frecuencia de cada categoría mediante valores absolutos o relativos.

Tabla de frecuencias unidimensional (una sóla variable)

Comenzamos generando nuestros datos manualmente dando valores a un vector. Por ejemplo, el objeto Cafeina que contiene el consumo de cafeina en miligramos/día en una muestra de 20 individuos.

```
Cafeina <-c(959,791,686,112,373,44,751,546,751,575,784,527,384,896,236,960,
22,72,649,934)
```

Utilizamos la función table(..) del paquete base

```
table(Cafeina)
```

```
## Cafeina
##  22  44  72 112 236 373 384 527 546 575 649 686 751 784 791 896 934 959 960
##   1   1   1   1   1   1   1   1   1   1   1   1   2   1   1   1   1   1   1
```

La tabla tal y como se presenta resulta inadecuada para resumir la información. Me conviene organizar las categorías por intervalos, para ello utilizo la función cut(...). Creo el objeto Cafeina_i donde guardaré la variable Cafeina dividida en 4 intervalos de igual amplitud. La misma función me permite especificar la amplitud y el número de intervalos.

```
Cafeina_i <- cut(Cafeina, breaks = 4)
table(Cafeina_i)
```

```
## Cafeina_i
## (21.1,256]  (256,491]  (491,726]  (726,961]
##          5          2          5          8
```

Si queremos mostrar las frecuencias acumuladas:

```
cumsum(table(Cafeina_i))
```

```
## (21.1,256]  (256,491]  (491,726]  (726,961]
##          5          7         12         20
```

Para mostrar la frecuencia en proporciones, utilizo la función prop.table(...).

```
prop.table(table(Cafeina_i))
```

```
## Cafeina_i
## (21.1,256]  (256,491]  (491,726]  (726,961]
##       0.25       0.10       0.25       0.40
```

Gestión de valores missing

En ocasiones nuestro vector puede tener valores desconocidos (missing). Veamos de nuevo el vector Cafeina, ahora con 23 observaciones de las cuales las dis últimas son missing (NA).

```
Cafeina <-c(959,791,686,112,373,44,751,546,751,575,784,527,384,896,236,960,
            22,72,649,934, 1230, NA, NA)
```

Para que la tabla contabilice los valores missing, habrá que indicarlo en la función.

```
table(Cafeina, useNA = "ifany")
```

```
## Cafeina
##   22   44   72  112  236  373  384  527  546  575  649  686  751  784  791  896
##    1    1    1    1    1    1    1    1    1    1    1    1    2    1    1    1
##  934  959  960 1230 <NA>
##    1    1    1    1    2
```

A continuación organizo la variable en intervalos, dejando uno para los valores missing.

```
Cafeina_i <- cut(Cafeina, breaks = c(NA, NA, 21, 100, 400, 600, Inf))

table(Cafeina_i, useNA = "ifany")
```

```
## Cafeina_i
##  (21,100] (100,400] (400,600] (600,Inf]      <NA>
##         3         4         3        11         2
```

Finalmente, indicar que la función table(...) permite excluir de la tabla categorías concretas. Por ejemplo:

```
table(Cafeina_i, exclude=c("(21,100]", "(100,400]"))
```

```
## Cafeina_i
## (400,600] (600,Inf]      <NA>
##         3        11         2
```

Tabla de frecuencias bidimensional (dos variables)

Denominada también como tabla de contingencia o tabla de doble entrada. Está compuesta por filas (horizontales), para la información de una variable y columnas (verticales) para la información de otra variable. Estas filas y columnas delimitan las celdas donde se vuelcan las frecuencias de cada combinación de las variables analizadas.

Incorporo al análisis una nueva variable. Una variable cualitativa que recoge si el individuo tiene Sí/No problemas para dormir.

```
Problema <- c("Sí", "Sí", "No", "Sí", "No", "No", "Sí", "No", "Sí",
              "Sí", "Sí", "No", "Sí", "No", "Sí", "No", "No", "Sí",
              "No", "Sí", "Sí", "No", "No")
```

La primera variable en la función irá a la vertical y la segunda a la horizontal.

```
table(Problema, Cafeina, useNA = "ifany")
```

```
##         Cafeina
## Problema 22 44 72 112 236 373 384 527 546 575 649 686 751 784 791 896 934 959
##       No  1  1  0   0   0   1   0   1   1   0   1   1   0   0   0   1   0   0
##       Sí  0  0  1   1   1   0   1   0   0   1   0   0   2   1   1   0   1   1
##         Cafeina
## Problema 960 1230 <NA>
##       No   1    0    2
##       Sí   0    1    0
```

```
prop.table(table(Problema, Cafeina_i))
```

```
##         Cafeina_i
## Problema   (21,100]  (100,400]  (400,600]  (600,Inf]
##       No 0.09523810 0.04761905 0.09523810 0.19047619
##       Sí 0.04761905 0.14285714 0.04761905 0.33333333
```

Por defecto, la tabla muestra la proporción de observaciones en cada celda respecto al total de observaciones. Si quiero las proporciones por columnas debo incluir margin=2 (=1 si quiero la proporción por filas).

```
prop.table(table(Problema, Cafeina_i), margin=2)
```

```
##         Cafeina_i
## Problema  (21,100] (100,400] (400,600] (600,Inf]
##       No 0.6666667 0.2500000 0.6666667 0.3636364
##       Sí 0.3333333 0.7500000 0.3333333 0.6363636
```

Con addmargins(...) puedo añadir a la tabla la suma de las columnas (margin=1) o de las filas (margin=2)

```
addmargins(table(Problema, Cafeina_i), margin=1)
```

```
##         Cafeina_i
## Problema (21,100] (100,400] (400,600] (600,Inf]
##      No         2         1         2         4
##      Sí         1         3         1         7
##      Sum        3         4         3        11
```

Veamos ahora un ejemplo de utilización de las funciones anteriores en variables que provienen de un data.frame. Para ello deberás cargar el dataset, numberofwords.csv. Consulta el fichero basesdedatos.pdf para conocer su contenido.

Carga la base de datos con import Dataset de R studio y echa un vistazo a la estructura de los datos.

```
DF <- read.csv("numberofwords.csv")
str(DF)  # estructura de los datos
```

```
## 'data.frame':    268 obs. of  8 variables:
##  $ id     : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ age    : int  20 20 20 20 20 20 20 20 18 18 ...
##  $ gender : chr  "male" "male" "female" "female" ...
##  $ region : chr  "urban" "rural" "urban" "urban" ...
##  $ words  : int  34 19 40 540 34 36 58 23 35 133 ...
##  $ nouns  : int  18 2 11 292 14 16 26 3 11 91 ...
##  $ verbs  : int  0 0 2 81 4 1 3 0 3 4 ...
##  $ ageat1w: int  18 18 18 12 18 18 17 18 16 14 ...
```

```
head(DF)
```

```
##   id age gender region words nouns verbs ageat1w
## 1  1  20   male  urban    34    18     0      18
## 2  2  20   male  rural    19     2     0      18
## 3  3  20 female  urban    40    11     2      18
## 4  4  20 female  urban   540   292    81      12
## 5  5  20   male  urban    34    14     4      18
## 6  6  20   male  urban    36    16     1      18
```

Replica el análisis de los apartados anteriores. Utilizando las variable age y gender:

```
table(DF$age)       # tabla de frecuencias de la variable age
```

```
##
## 16 18 20
## 91 79 98
```

```
prop.table(table(DF$age)) # frecuencias en proporciones (tanto por uno)
```

```
##
##        16        18        20
## 0.3395522 0.2947761 0.3656716
```

```
table(DF$gender)       # tabla de frecuencias de la variable age
```

```
##
## female   male
##    143    125
```

```
prop.table(table(DF$gender)) # frecuencias en proporciones (tanto por uno)
```

```
##
##    female      male
## 0.5335821 0.4664179
```

Imagina que quieres otro orden diferente al alfabético

```
DF$gender <- ordered(DF$gender,
                         levels = c("male", "female"),
                         labels = c("male", "female"))

table(DF$gender)
```

```
##
##   male female
##    125    143
```

Como tabla de doble entrada:

```
with(DF, table(gender, ageat1w))
```

```
##         ageat1w
## gender    9 10 11 12 13 14 15 16 17 18 19
##   male    1  0  1  2  3 29 25 31 11 15  3
##   female  0  1  1  2  6 33 28 30 22 12  2
```

```
with(DF, prop.table(table(gender, ageat1w)))
```

```
##         ageat1w
## gender             9          10          11          12          13
##   male   0.003875969 0.000000000 0.003875969 0.007751938 0.011627907
##   female 0.000000000 0.003875969 0.003875969 0.007751938 0.023255814
##         ageat1w
## gender            14          15          16          17          18
##   male   0.112403101 0.096899225 0.120155039 0.042635659 0.058139535
##   female 0.127906977 0.108527132 0.116279070 0.085271318 0.046511628
##         ageat1w
## gender            19
##   male   0.011627907
##   female 0.007751938
```

```
round(with(DF, prop.table(table(gender, ageat1w))), 2) # redondeando a 2 decimales
```

```
##         ageat1w
## gender      9   10   11   12   13   14   15   16   17   18   19
##   male   0.00 0.00 0.00 0.01 0.01 0.11 0.10 0.12 0.04 0.06 0.01
##   female 0.00 0.00 0.00 0.01 0.02 0.13 0.11 0.12 0.09 0.05 0.01
```

Puedo incorporar una tercera variable:

```
with(DF, table(gender, ageat1w, region))
```

```
## , , region = rural
##
##         ageat1w
## gender    9 10 11 12 13 14 15 16 17 18 19
##   male    0  0  0  0  1 15 13 16  7  6  2
##   female  0  1  0  1  5 19 22 22 12  7  2
##
## , , region = urban
##
##         ageat1w
## gender    9 10 11 12 13 14 15 16 17 18 19
##   male    1  0  1  2  2 14 12 15  4  9  1
##   female  0  0  1  1  1 14  6  8 10  5  0
```

```
xtabs(~ gender + ageat1w + region, data=DF) # otra alternativa
```

```
## , , region = rural
##
##         ageat1w
## gender    9 10 11 12 13 14 15 16 17 18 19
##   male    0  0  0  0  1 15 13 16  7  6  2
##   female  0  1  0  1  5 19 22 22 12  7  2
##
## , , region = urban
##
##         ageat1w
## gender    9 10 11 12 13 14 15 16 17 18 19
##   male    1  0  1  2  2 14 12 15  4  9  1
##   female  0  0  1  1  1 14  6  8 10  5  0
```

- Tablas de Frecuencias utilizando el paquete "summarytools"

Para más informaciónsobre este paquete consulta: https://cran.r-project.org/web/packages/summarytools/vignettes/introduction.html

Cargamos la librería una vez descargado el paquete.

```
library(summarytools)

freq(DF)
```

```
## Frequencies
## DF$age
## Type: Integer
##
##               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## ----------- ------ --------- -------------- --------- --------------
##          16     91     33.96          33.96     33.96          33.96
##          18     79     29.48          63.43     29.48          63.43
##          20     98     36.57         100.00     36.57         100.00
##        <NA>      0                               0.00         100.00
##       Total    268    100.00         100.00    100.00         100.00
##
## DF$gender
## Type: Ordered Factor
##
##                Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## ------------ ------ --------- -------------- --------- --------------
##         male    125     46.64          46.64     46.64          46.64
##       female    143     53.36         100.00     53.36         100.00
##         <NA>      0                               0.00         100.00
##        Total    268    100.00         100.00    100.00         100.00
##
## DF$region
## Type: Character
##
##               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## ----------- ------ --------- -------------- --------- --------------
##       rural    157     58.58          58.58     58.58          58.58
##       urban    111     41.42         100.00     41.42         100.00
##        <NA>      0                               0.00         100.00
##       Total    268    100.00         100.00    100.00         100.00
##
## DF$ageat1w
## Type: Integer
##
##               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## ----------- ------ --------- -------------- --------- --------------
##           9      1      0.39           0.39      0.37           0.37
##          10      1      0.39           0.78      0.37           0.75
##          11      2      0.78           1.55      0.75           1.49
##          12      4      1.55           3.10      1.49           2.99
##          13      9      3.49           6.59      3.36           6.34
##          14     62     24.03          30.62     23.13          29.48
```

```
##          15     53     20.54          51.16     19.78          49.25
##          16     61     23.64          74.81     22.76          72.01
##          17     33     12.79          87.60     12.31          84.33
##          18     27     10.47          98.06     10.07          94.40
##          19      5      1.94         100.00      1.87          96.27
##        <NA>     10                                3.73         100.00
##       Total    268    100.00         100.00    100.00         100.00
```

Para evitar saturar de filas la tabla de resultados anterior, se ignoran las variables que tienen más de 25 valores diferentes. Este umbral se puede modificar st_options(); Por ejemplo, para cambiarlo a 5, usaríamos st_options(freq.ignore.threshold = 5)

De nuevo podemos eliminar o mostrar la columna de valores missing. Este paquete nos permite también orderar las categorías por orden de frecuencia.

```
freq(DF$gender, report.nas = F)
```

```
## Frequencies
## DF$gender
## Type: Ordered Factor
##
##                Freq        %   % Cum.
## ------------ ------ -------- --------
##         male    125    46.64    46.64
##       female    143    53.36   100.00
##        Total    268   100.00   100.00
```

```
freq(DF$ageat1w, report.nas = T)
```

```
## Frequencies
## DF$ageat1w
## Type: Integer
##
##               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## ----------- ------ --------- -------------- --------- --------------
##           9      1      0.39           0.39      0.37           0.37
##          10      1      0.39           0.78      0.37           0.75
##          11      2      0.78           1.55      0.75           1.49
##          12      4      1.55           3.10      1.49           2.99
##          13      9      3.49           6.59      3.36           6.34
##          14     62     24.03          30.62     23.13          29.48
##          15     53     20.54          51.16     19.78          49.25
##          16     61     23.64          74.81     22.76          72.01
##          17     33     12.79          87.60     12.31          84.33
##          18     27     10.47          98.06     10.07          94.40
##          19      5      1.94         100.00      1.87          96.27
##        <NA>     10                                3.73         100.00
##       Total    268    100.00         100.00    100.00         100.00
```

```
freq(DF$ageat1w, order="freq", report.nas = T)
```

```
## Frequencies
```

```
## DF$ageat1w
## Type: Integer
##
##               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## ----------- ------ --------- -------------- --------- --------------
##          14     62     24.03          24.03     23.13          23.13
##          16     61     23.64          47.67     22.76          45.90
##          15     53     20.54          68.22     19.78          65.67
##          17     33     12.79          81.01     12.31          77.99
##          18     27     10.47          91.47     10.07          88.06
##          13      9      3.49          94.96      3.36          91.42
##          19      5      1.94          96.90      1.87          93.28
##          12      4      1.55          98.45      1.49          94.78
##          11      2      0.78          99.22      0.75          95.52
##           9      1      0.39          99.61      0.37          95.90
##          10      1      0.39         100.00      0.37          96.27
##        <NA>     10                                3.73         100.00
##       Total    268    100.00         100.00    100.00         100.00
```

Para generar tablas de doble entrada utilizamos ctable(. . .):

```
ctable(x=DF$gender, y=DF$region, prop ="r")
```

```
## Cross-Tabulation, Row Proportions
## gender * region
## Data Frame: DF
##
## -------- -------- ------------- ------------- --------------
##            region         rural         urban          Total
##   gender
##     male             64 (51.2%)    61 (48.8%)   125 (100.0%)
##   female             93 (65.0%)    50 (35.0%)   143 (100.0%)
##    Total            157 (58.6%)   111 (41.4%)   268 (100.0%)
## -------- -------- ------------- ------------- --------------
```

```
# "c" colum, "t" total "n" omit
```

Ejercicio:

1. ¿Cuántos niños de 16, 18 y 20 meses de edad hay en el conjunto de datos?
2. ¿Cuántos niños residentes en zonas rurales tienen 16, 18 y 20 meses de edad?
3. ¿Existe diferencia en la distribución por edades entre los niños residentes en hábitat urbano o rural?

```
# Solución:

addmargins(table(DF$age), margin=1)
```

```
##
##  16  18  20 Sum
##  91  79  98 268
```

```r
addmargins(table(DF$age[DF$region=="rural"]), margin=1)
```

```
##
##  16  18  20 Sum
##  53  49  55 157
```

```r
ctable(x=as.character(DF$age), y=DF$region,
       prop ="c",
       chisq = TRUE)
```

```
## Cross-Tabulation, Column Proportions
## as.character(DF$age) * region
##
##
## ---------------------- -------- -------------- -------------- --------------
##                           region          rural          urban          Total
##   as.character(DF$age)
##                     16             53 ( 33.8%)    38 ( 34.2%)    91 ( 34.0%)
##                     18             49 ( 31.2%)    30 ( 27.0%)    79 ( 29.5%)
##                     20             55 ( 35.0%)    43 ( 38.7%)    98 ( 36.6%)
##                  Total            157 (100.0%)   111 (100.0%)   268 (100.0%)
## ---------------------- -------- -------------- -------------- --------------
##
## ----------------------------
##  Chi.squared   df   p.value
## ------------- ---- ---------
##    0.6347      2    0.7281
## ----------------------------
```

Ejercicio propuesto:

Utiliza funciones de R para responder cada una de las siguientes preguntas. Utiliza la base de datos "Tobacco.csv".

1. Etiqueta la variable "age".
2. Construye una tabla de frecuencia absoluta para la variable "region".
3. Construye una tabla de frecuencia relativa para la variable "ocupation".
4. Construye una tabla de frecuencia relativa para las variables "region" y "ocupation.
5. ¿Qué porcentaje de los menores de 30 años tiene niños menores de 2 años.
6. ¿Qué porcentaje de los que viven en "brussels" gástan dinero en tabaco?

Estadísticos descriptivos

- ¿Qué preguntar a nuestros datos?

Dos preguntas surgen siempre al principio de cualquier AED:

1. Cuál es el valor central de mis variables
2. ¿Qué tipo de variación se produce dentro de mis variables?
3. ¿Qué tipo de correlación hay entre mis variables?

- Medidas de tendencia central

Para este apartado utilizaremos las variables de la base de datos "Medidas.xlsx". Carga los datos y realiza una primera exploración de los mismos.

```
library(readxl)
Medidas <- read_excel("Medidas.xlsx")

head(Medidas)
```

```
## # A tibble: 6 x 6
##    edad  peso altura sexo   muneca biceps
##   <dbl> <dbl>  <dbl> <chr>   <dbl>  <dbl>
## 1    43  87.3   188  Hombre   12.2   35.8
## 2    65  80     174  Hombre   12     35
## 3    45  82.3   176. Hombre   11.2   38.5
## 4    37  73.6   180. Hombre   11.2   32.2
## 5    55  74.1   168. Hombre   11.8   32.9
## 6    33  85.9   188  Hombre   12.4   38.5
```

Utilizo la función range(...) para obtener el valor máximo y mínimo de una variable, Por ejemplo la altura. Otras funciones, max(...) y min(...).

```
range(Medidas$altura)
```

```
## [1] 147.2 190.5
```

Media: función mean(...)

Para calcular la media aritmética de una variable cuantitativa se usa la función mean(...). Los argumentos de la función mean son tres: x (objeto sobre el que calculo la media), trim = , (fracción de observaciones que son excluidas del cálculo) na.rm = FALSE, (función lógica para eliminar "TRUE" las observaciones missing).

Supón que queremos obtener la altura media de los estudiantes. Calcula también la altura media eliminado el 5% de las observaciones más bajas y más altas.

```
mean(Medidas$edad)
```

```
## [1] 31.44444
```

```
mean(Medidas$altura, trim=0.05)
```

```
## [1] 171.7147
```

Suponga que ahora queremos la altura media pero de hombres y mujeres por separado.

```
mean(Medidas$altura[Medidas$sexo=="Hombre"])
```

```
## [1] 179.0778
```

```r
mean(Medidas$altura[Medidas$sexo=="Mujer"])
```

```
## [1] 164.0333
```

Otra alternativa, utilizando la función aggregate(...)

```r
aggregate(Medidas$altura, by=list(Medidas$sexo), mean)
```

```
##   Group.1        x
## 1  Hombre 179.0778
## 2   Mujer 164.0333
```

Mediana: función median(...) y quantile(...)

Calcula la edad mediana de los estudiantes de la base de datos, así como el percentil 32 de la variable peso.

```r
median(Medidas$edad)
```

```
## [1] 28
```

```r
quantile(Medidas$altura, probs = 0.32)
```

```
##   32%
## 167.6
```

Calcula el recorrido intercuartílico de la variable altura.

```r
Q <- quantile(Medidas$altura, probs = c(0.25, 0.75))
Q
```

```
##   25%   75%
## 164.8 179.4
```

```r
R.I <- unname(Q[2]) -unname(Q[1])
R.I
```

```
## [1] 14.6
```

```r
# Hay una función propia en R base

IQR(Medidas$altura)
```

```
## [1] 14.6
```

Ejercicio:

1. ¿Cuál es la estatura mínima y máxima de los sujetos de la muestra? ¿Y la de los varones? ¿Y la de las mujeres?
2. ¿Qué porcentaje de sujetos tiene una estatura menor de 1,65 m.?
3. ¿Cuál es el valor central de la variable estatura? ¿Y de la variable peso?
4. Obtenga el valor del peso que es superado sólo por el 15% de la muestra.
5. ¿Cuántos mujeres hay en la muestra?
6. ¿Qué porcentaje de hombres tiene un tamaño de muñeca superio a 11?
7. ¿Quiénes tienen mayores biceps, los hombres o las mujeres?

Medidas de dispersión: Varianza y Desviación Típica (muestral)

```
var(Medidas$edad)
```

```
## [1] 111.3968
```

```
sd(Medidas$altura)
```

```
## [1] 10.52017
```

Matriz de varianzas y covarianzas. La función var(...) se puede aplicar sobre un marco de datos, por ejemplo variables de un data.frame. Obteniendo en la diagonal principal las varianzas individuales y fuera de ella las covarianzas.

```
var(Medidas)
```

```
##              edad      peso     altura sexo    muneca    biceps
## edad   111.396825  80.88159  36.666032   NA  7.698095 26.720952
## peso    80.881587 221.08713 124.728698   NA 14.844667 70.738381
## altura  36.666032 124.72870 110.673968   NA  8.156476 39.021048
## sexo           NA        NA         NA   NA        NA        NA
## muneca   7.698095  14.84467   8.156476   NA  1.381714  5.400571
## biceps  26.720952  70.73838  39.021048   NA  5.400571 27.398857
```

```
var(Medidas[-4]) # aliminando la variable sexo
```

```
##              edad      peso     altura    muneca    biceps
## edad   111.396825  80.88159  36.666032  7.698095 26.720952
## peso    80.881587 221.08713 124.728698 14.844667 70.738381
## altura  36.666032 124.72870 110.673968  8.156476 39.021048
## muneca   7.698095  14.84467   8.156476  1.381714  5.400571
## biceps  26.720952  70.73838  39.021048  5.400571 27.398857
```

Medidas de correlación

La función cor(...) permite calcular el coeficiente de correlación de Pearson, Kendall o Spearman para dos variables cuantitativas.

```
cor(Medidas$peso, Medidas$altura, method="pearson")
```

```
## [1] 0.7973737
```

```
with(Medidas, cor(peso, altura, method="kendall"))
```

```
## [1] 0.623303
```

Con la función cor.test(...) podemos contrastar estadísticamente si la correlación obtenida es distinta de cero.

```
with(Medidas, cor.test(peso, altura))
```

```
##
##  Pearson's product-moment correlation
##
## data:  peso and altura
## t = 7.7043, df = 34, p-value = 5.853e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6352523 0.8921870
## sample estimates:
##       cor
## 0.7973737
```

Tabla de principales descriptivos

Para finalizar el análisis descriptivo a partir de las funciones de R base. Mencionar la función summary(...) como un buen resumen de estadísticos principales. Esta función se puede aplicar sobre vectores o sobre la base de datos (data.frame) en su conjunto.

```
summary(Medidas$altura)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   147.2   164.8   172.7   171.6   179.4   190.5
```

```
summary(Medidas)
```

```
##       edad            peso           altura          sexo
##  Min.   :19.00   Min.   :42.00   Min.   :147.2   Length:36
##  1st Qu.:24.75   1st Qu.:54.95   1st Qu.:164.8   Class :character
##  Median :28.00   Median :71.50   Median :172.7   Mode  :character
##  Mean   :31.44   Mean   :68.95   Mean   :171.6
##  3rd Qu.:37.00   3rd Qu.:82.40   3rd Qu.:179.4
##  Max.   :65.00   Max.   :98.20   Max.   :190.5
##      muneca          biceps
##  Min.   : 8.300   Min.   :23.50
##  1st Qu.: 9.475   1st Qu.:25.98
##  Median :10.650   Median :32.15
##  Mean   :10.467   Mean   :31.17
##  3rd Qu.:11.500   3rd Qu.:35.05
##  Max.   :12.400   Max.   :40.40
```

Ejercicio:

Leer los datos del fichero "numberofwords.csv".

1. Calcular la media, mediana, de las variables "region" y "ageat1W".
2. Calcula la media recortada de la variable "ageat1w" con una proporción del 0,05,
3. Calcular los percentiles 10% y 90% en la variable "words".
4. Comprueba si existe relación lineal en tre las variables "word" y "ageat1w".
5. Supongamos que se han seguido recogiendo datos de esta variable "words". En concreto se añaden 4 observaciones a la muestra con los siguientes valores: 23 34 78 y 89. Calcula la varianza y desviación típica.

- Análisis descriptivo utilizando el paquete "summarytools"

Al igual que para el análisis de frecuencia, existen muchos paquetes con funciones específicas para realizar el análisis de principales estadísticos. Utilizaremos el paquete "summarytools". A modo de ejemplo trabajaremos de nuevo con el dataset "numberofwords.csv". Utilizamos la función descr(...)

```
# Cargo de nuevo los datos
DF <- read.csv("numberofwords.csv")
descr(DF$ageat1w) # Descriptivos de una variable concreta
```

```
## Descriptive Statistics
## DF$ageat1w
## N: 268
##
##                     ageat1w
## ----------------- ---------
##              Mean     15.45
##           Std.Dev      1.64
##               Min      9.00
##                Q1     14.00
##            Median     15.00
##                Q3     17.00
##               Max     19.00
##               MAD      1.48
##               IQR      2.75
##                CV      0.11
##          Skewness     -0.26
##       SE.Skewness      0.15
##          Kurtosis      0.60
##           N.Valid    258.00
##         Pct.Valid     96.27
```

```
descr(DF) # Descriptivos del total de variables numéricas
```

```
## Descriptive Statistics
## DF
## N: 268
##
##                       age   ageat1w       id    nouns    verbs    words
## ----------------- -------- --------- -------- -------- -------- --------
##              Mean    18.05     15.45   134.50    39.30     7.27    76.28
##           Std.Dev     1.68      1.64    77.51    61.24    15.86   105.36
##               Min    16.00      9.00     1.00     0.00     0.00     0.00
##                Q1    16.00     14.00    67.50     2.50     0.00    17.00
##            Median    18.00     15.00   134.50    12.00     1.00    35.00
##                Q3    20.00     17.00   201.50    49.50     5.00    88.50
##               Max    20.00     19.00   268.00   316.00   101.00   644.00
##               MAD     2.97      1.48    99.33    16.31     1.48    34.10
##               IQR     4.00      2.75   133.50    46.50     5.00    71.25
##                CV     0.09      0.11     0.58     1.56     2.18     1.38
##          Skewness    -0.05     -0.26     0.00     2.22     3.46     2.67
##       SE.Skewness     0.15      0.15     0.15     0.15     0.15     0.15
##          Kurtosis    -1.59      0.60    -1.21     4.75    12.74     7.96
```

```
##           N.Valid   268.00    258.00   268.00   268.00   268.00   268.00
##         Pct.Valid   100.00     96.27   100.00   100.00   100.00   100.00
```

```
descr(DF[,c(2,5:8)],
      stats     = c("mean", "sd"),
      transpose = TRUE,
      headings  = FALSE)
```

```
##
##                  Mean   Std.Dev
## ------------- ------- ---------
##           age   18.05      1.68
##       ageat1w   15.45      1.64
##         nouns   39.30     61.24
##         verbs    7.27     15.86
##         words   76.28    105.36
```

```
#view(dfSummary(DF[,-1]))
```

Pruebas de hipótesis

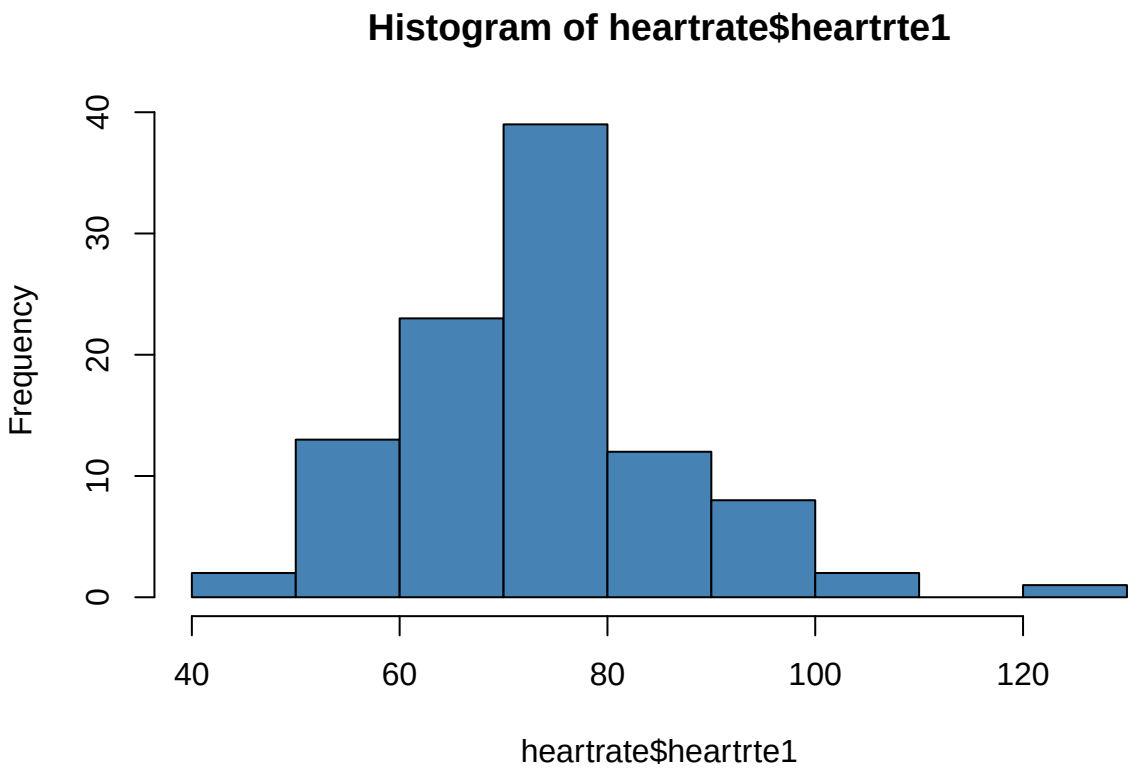
- Pruebas para una muestra

Prueba de hipótesis sobre la media de una población normal

Empezaremos con el contraste de hipótesis sobre una muestra en poblaciones normales. Para realizar este tipo de prueba se puede usar la función t.test(...) de R base. Para mostrar cómo funciona esta función utilizaremos el dataset "heartrate.dta".

Lo primero, contrastar la normalidad de mi variable. Trabajamos primero con la variable heartrate1. Una manera visula de verificar la normalidad es mediante un histograma.

```
# Cargo los datos
library(haven)
heartrate <- read_dta("heartrate.dta")
hist(heartrate$heartrte1, col='steelblue')
```



El histograma nos muestra un valor extremo (outlier) a la derecha. Es probable que dicho valor afecte al test de normalidad. Lo comprobamos.

```
shapiro.test(heartrate$heartrte1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  heartrate$heartrte1
## W = 0.96397, p-value = 0.0078
```

```
shapiro.test(heartrate$heartrte1[heartrate$heartrte1<120])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  heartrate$heartrte1[heartrate$heartrte1 < 120]
## W = 0.98391, p-value = 0.2706
```

La presencia de dicho outlier podría afectar al resultado del contraste.

```
t.test(heartrate$heartrte1, alternative='two.sided',
       conf.level=0.95, mu=80)
```

```
##
##  One Sample t-test
##
## data:  heartrate$heartrte1
## t = -3.852, df = 99, p-value = 0.0002081
## alternative hypothesis: true mean is not equal to 80
## 95 percent confidence interval:
##  72.46987 77.59013
## sample estimates:
## mean of x
##     75.03
```

```
t.test(heartrate$heartrte1[heartrate$heartrte1<120], alternative='two.sided',
       conf.level=0.95, mu=80)
```

```
##
##  One Sample t-test
##
## data:  heartrate$heartrte1[heartrate$heartrte1 < 120]
## t = -4.5643, df = 98, p-value = 1.456e-05
## alternative hypothesis: true mean is not equal to 80
## 95 percent confidence interval:
##  72.14493 76.90557
## sample estimates:
## mean of x
##  74.52525
```

En ambos contrastes no puedo aceptar la hipótesis nula. Realiza ahora el análisis sobre la variable heartrate2 que mide las pulsaciones despues de tratar al paciente.

Aunque la prueba t es relativamente sólida frente a desviaciones de la distribución normal, especialmente en muestras grandes, a veces debemos evitar el supuesto de normalidad. Como alternativa utilizaremos los contrastes no paramétricos. El contraste no-paramétrico equivalente a la prueba t es la prueba de los rangos con signo de Wilcoxon.

```
wilcox.test(heartrate$heartrte1, alternative='two.sided', mu=80)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  heartrate$heartrte1
## V = 965, p-value = 5.73e-05
## alternative hypothesis: true location is not equal to 80
```

Ejercicio:

Un artículo recogía los resultados de un estudio para investigar las concentraciones de mercurio en las lubinas de una granja marina. Se tomaron muestras de 53 jaulas en alta mar y se midió la concentración de mercurio en el tejido muscular (ppm). Los valores observados fueron: 1.230, 1.330, 0.040, 0.044, 1.200, 0.270, 0.490, 0.190, 0.830, 0.810, 0.710, 0.500, 0.490, 1.160, 0.050, 0.150, 0.190, 0.770, 1.080, 0.980, 0.630, 0.560, 0.410, 0.730, 0.590, 0.340, 0.340, 0.840, 0.500, 0.340, 0.280, 0.340, 0.750, 0.870, 0.560, 0.170, 0.180, 0.190, 0.040, 0.490, 1.100, 0.160, 0.210, 0.860, 0.520, 0.650, 0.270, 0.940, 0.400, 0.430, 0.250, 0.270. Se pide:

1) Construir un intervalo de confianza para la media con nivel de confianza 0,90. 2) Contrastar la hipótesis nula de una concentración media menor o igual a 0,7. 3) Contrastar la hipótesis nula de una concentración media igual a 0,7.

- Pruebas para dos muestras

Muestras independientes

Ahora comparamos dos poblaciones. De cada una de ellas disponemos de una muestra. Veamos cuando dichas muestras son independientes. Por ejemplo, los latidos en hombres y mujeres.

```
Male <- heartrate$heartrte1[heartrate$sex==1]
Female <- heartrate$heartrte1[heartrate$sex==2]

t.test(Male, Female, paried=FALSE,
       var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  Male and Female
## t = -0.4141, df = 98, p-value = 0.6797
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -6.371669  4.171585
## sample estimates:
## mean of x mean of y
##  74.35897  75.45902
```

```
t.test(heartrate$heartrte1~heartrate$sex, paried=FALSE,
       var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  heartrate$heartrte1 by heartrate$sex
## t = -0.4141, df = 98, p-value = 0.6797
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
## 95 percent confidence interval:
##  -6.371669  4.171585
## sample estimates:
## mean in group 1 mean in group 2
##        74.35897        75.45902
```

```
wilcox.test(heartrate$heartrte1~heartrate$sex, paried=FALSE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  heartrate$heartrte1 by heartrate$sex
## W = 1058.5, p-value = 0.354
## alternative hypothesis: true location shift is not equal to 0
```

Contraste sobre las varianzas

```
var.test(heartrate$heartrte1~heartrate$sex)
```

```
##
##  F test to compare two variances
##
## data:  heartrate$heartrte1 by heartrate$sex
## F = 1.0916, num df = 38, denom df = 60, p-value = 0.7487
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.621780 1.991464
## sample estimates:
## ratio of variances
##           1.091623
```

Muestras pareadas

Muestras no independientes emparajadas uno a uno. Probemos a contrastar las diferencias en el número de pulsaciones en el examen 1 frente al examen 2.

```
var.test(heartrate$heartrte1, heartrate$heartrte2)
```

```
##
##  F test to compare two variances
##
## data:  heartrate$heartrte1 and heartrate$heartrte2
## F = 0.9957, num df = 99, denom df = 99, p-value = 0.9829
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.6699474 1.4798406
## sample estimates:
## ratio of variances
##          0.9956984
```

```
t.test(heartrate$heartrte1, heartrate$heartrte2, paried=TRUE,
       var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  heartrate$heartrte1 and heartrate$heartrte2
## t = -0.62409, df = 198, p-value = 0.5333
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.742196  2.462196
## sample estimates:
## mean of x mean of y
##     75.03     76.17
```

```
wilcox.test(heartrate$heartrte1, heartrate$heartrte2, paried=TRUE,
       var.equal=TRUE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  heartrate$heartrte1 and heartrate$heartrte2
## W = 4762, p-value = 0.5597
## alternative hypothesis: true location shift is not equal to 0
```

- Pruebas de hipótesis para proporciones

- Pruebas para una muestra

Utiliza el dataset de Stata "healthdisparities.dta". Contrastamos en primer lugar si el porcentaje de personas pobres de mi muestra puede ser igual al 10%. Tenemos dos posibilidades:

```
library(haven)
HD <- read_dta("healthdisparities.dta")
```

En primer lugar calcula cuál es el númerototal de pobres de mi muestra.

```
# Si la muestra es pequeña
binom.test(44, 63, p = .1, alternative = "two.sided")
```

```
##
##  Exact binomial test
##
## data:  44 and 63
## number of successes = 44, number of trials = 63, p-value < 2.2e-16
## alternative hypothesis: true probability of success is not equal to 0.1
## 95 percent confidence interval:
##  0.5697502 0.8076894
## sample estimates:
## probability of success
##              0.6984127
```

```
# Para muestras grandes
prop.test(44, 63, p = .1, alternative = "two.sided")
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  44 out of 63, null probability 0.1
## X-squared = 244.06, df = 1, p-value < 2.2e-16
## alternative hypothesis: true p is not equal to 0.1
## 95 percent confidence interval:
##  0.5681627 0.8043141
## sample estimates:
##         p
## 0.6984127
```

- Pruebas para varias muestras

Por ejemplo, compara la propoción de individuos que ha acudido al médico al menos una vez en los últimos 12 méses, diferenciando entre pacientes pobres y no pobres.

```
chisq.test(HD$doctor, HD$poverty)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  HD$doctor and HD$poverty
## X-squared = 4.3618, df = 1, p-value = 0.03675
```

¿Recuerdas cómo se hacía con el paquete "summarytools"?

Ejercicio:

El fabricante de un contraste para pruebas radiológicas afirma que su producto es visible en el 90 % de todas las radiografías de cadera. Para poner a prueba esta afirmación se realizan 200 radiografías y en 174 se percibe el contraste correctamente. Prueba la afirmación del fabricante a un nivel de significación alfa = 0.01.

- Correlación y regresión

Como ya sabemos, la función cor(...) calcula la correlación de Pearson entre dos variables, X e Y, medidas a nivel cuantitativo. Por ejemplo, la correlación entre altura y peso en el data.frame "Medidas" se obtiene mediante:

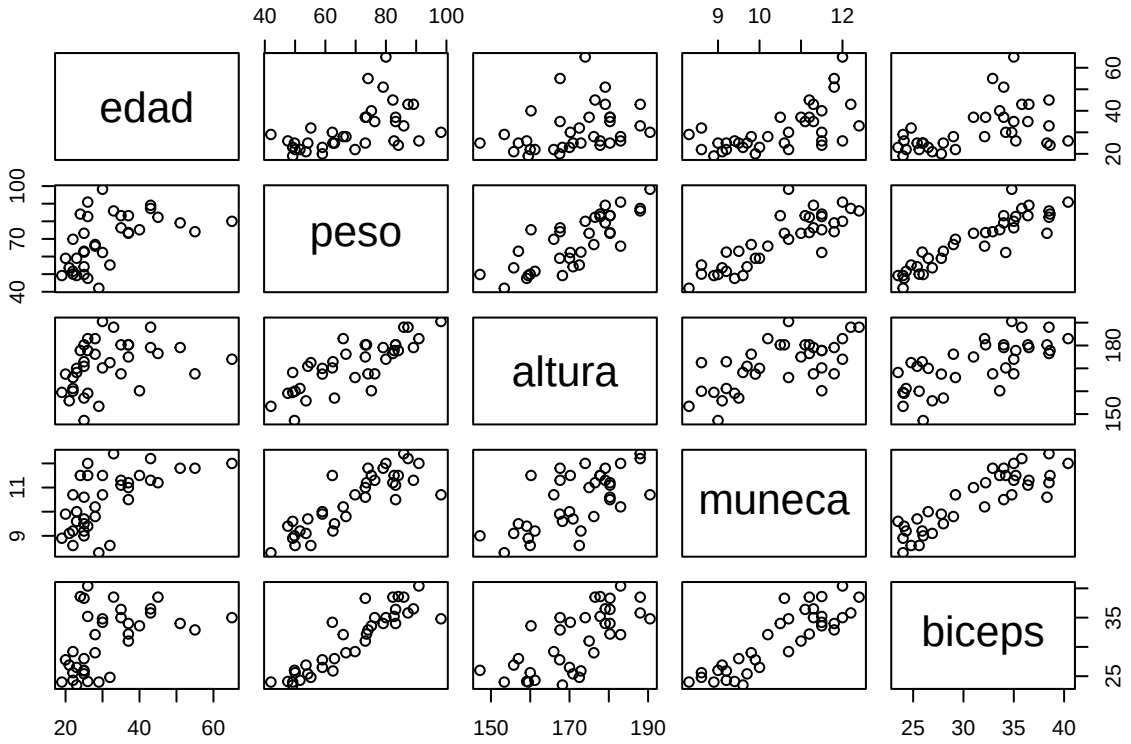
```
library(readxl)
Medidas <- read_excel("Medidas.xlsx")

with(Medidas, cor(altura, peso))
```

```
## [1] 0.7973737
```

Cuando trabajamos con más de dos variables nos puede resulta útil elaborar un diagrama de dispersión que nos proporcione una visual rápida de las posibles asociaciones entre pares de variables. La función pairs(...) es la adecuada.

```
pairs(Medidas[-4]) # dejo fuera la variable sexo
```



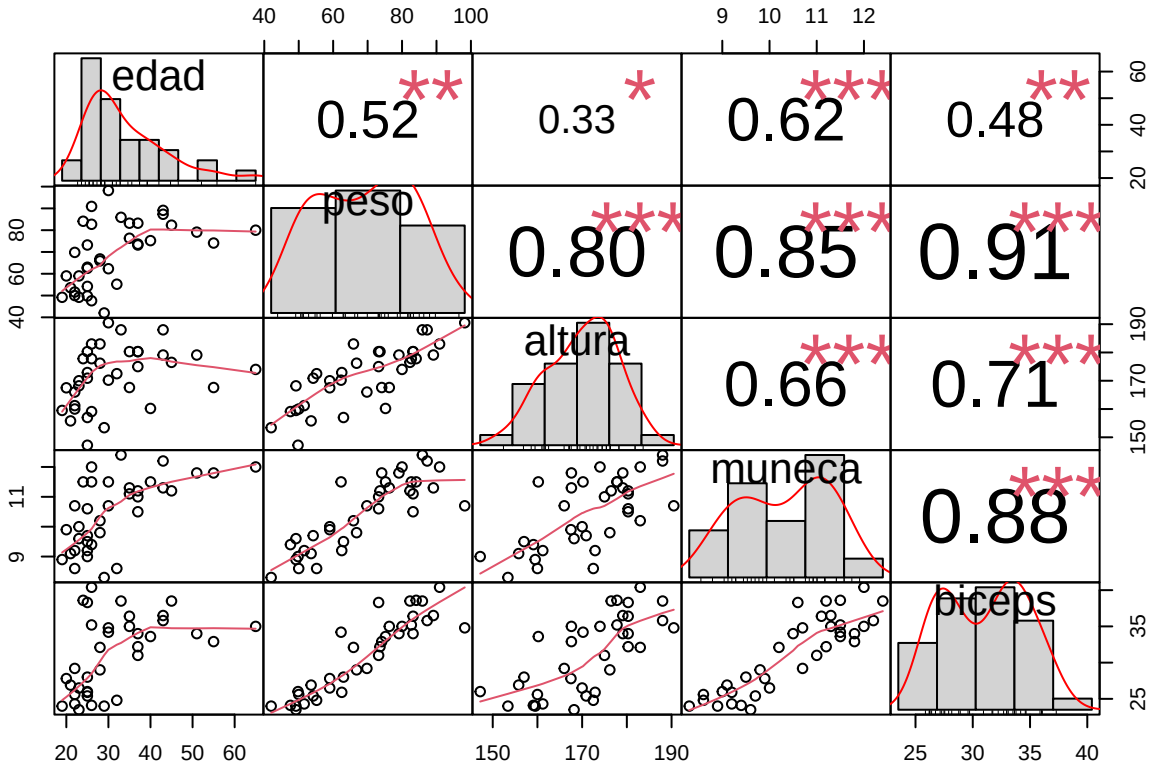
con los siguientes valores en la matriz de correlaciones

```
cor(Medidas[-4])
```

```
##             edad      peso    altura    muneca    biceps
## edad   1.0000000 0.5153847 0.3302211 0.6204942 0.4836702
## peso   0.5153847 1.0000000 0.7973737 0.8493361 0.9088813
## altura 0.3302211 0.7973737 1.0000000 0.6595849 0.7086144
## muneca 0.6204942 0.8493361 0.6595849 1.0000000 0.8777369
## biceps 0.4836702 0.9088813 0.7086144 0.8777369 1.0000000
```

¿y por qué no, todo junto? Utilizando el paquete “PerformanceAnalytics”

```
library(PerformanceAnalytics)
chart.Correlation(Medidas[-4])
```



Modelos de regresión lineal

Para ilustrar las funciones de R comenzaremos con una sencilla aplicación con datos del dataset, "calcium.csv". Carga los datos y explora su contenido.

```
calcium <- read.csv("calcium.csv")
```

La función lm(...) es la función de R base para estimar modelos lineales. Si la aplicamos a nuestro ejemplo:

```
lm(weight ~ dose, data=calcium)
```

```
##
## Call:
## lm(formula = weight ~ dose, data = calcium)
##
## Coefficients:
## (Intercept)         dose
##      64.957        4.243
```

La función sólo muestra el intercepto (término constante) y el coeficiente de la variable explicativa. No muestra medidas de bondad del ajuste, contrastes estadísticos, etc. Para ello debemos guardar la estimación en un objeto. Por ejemplo M1, sobre el cual utilizamos las funciones summary(...) y confint(...):

```r
M1 <- lm(weight ~ dose, data=calcium)
summary(M1)
```

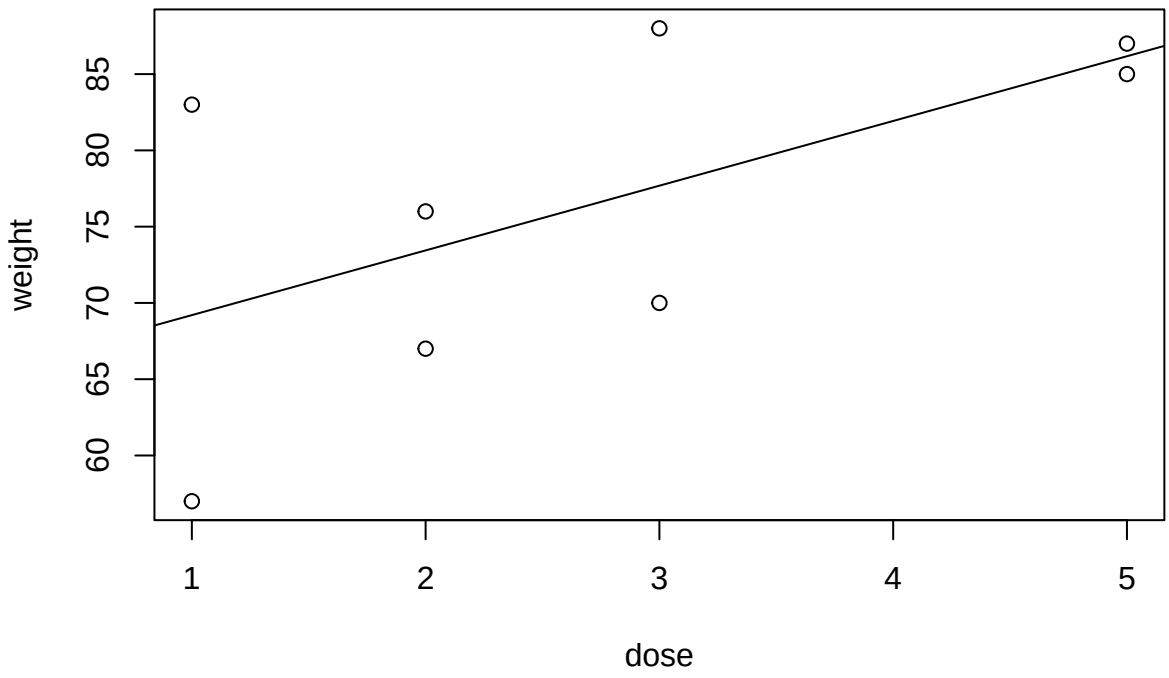
```
##
## Call:
## lm(formula = weight ~ dose, data = calcium)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.2000  -6.7536  -0.1714   4.4964  13.8000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   64.957      7.178   9.050 0.000102 ***
## dose           4.243      2.299   1.846 0.114466
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.616 on 6 degrees of freedom
## Multiple R-squared:  0.3622, Adjusted R-squared:  0.2559
## F-statistic: 3.407 on 1 and 6 DF,  p-value: 0.1145
```

```r
confint(M1, level=0.95)
```

```
##                 2.5 %    97.5 %
## (Intercept) 47.393646 82.520640
## dose        -1.381963  9.867678
```

Aunque en próximas sesiones ampliaremos las posibilidades del analálisis gráfico con R, aquí utilizaremos un sencillo scatter plot para ilustrar nuestro ejemplo.

```r
plot(weight ~ dose, data = calcium)
abline(M1)
```



Para valorar la calidad de mi ajuste y el cumplimiento de las hipótesis básicas detrás de un modelo de regressión lineal, conviene obtener la predicción y el residuo (errores) del modelo. Esto lo haremos con las funciones predict(...) y residuals(...).

```
predict(M1)
```

```
##        1        2        3        4        5        6        7        8
## 69.20000 69.20000 73.44286 73.44286 77.68571 77.68571 86.17143 86.17143
```

```
residuals(M1)
```

```
##           1           2           3           4           5           6
##  13.8000000 -12.2000000   2.5571429  -6.4428571  10.3142857  -7.6857143
##           7           8
##   0.8285714  -1.1714286
```

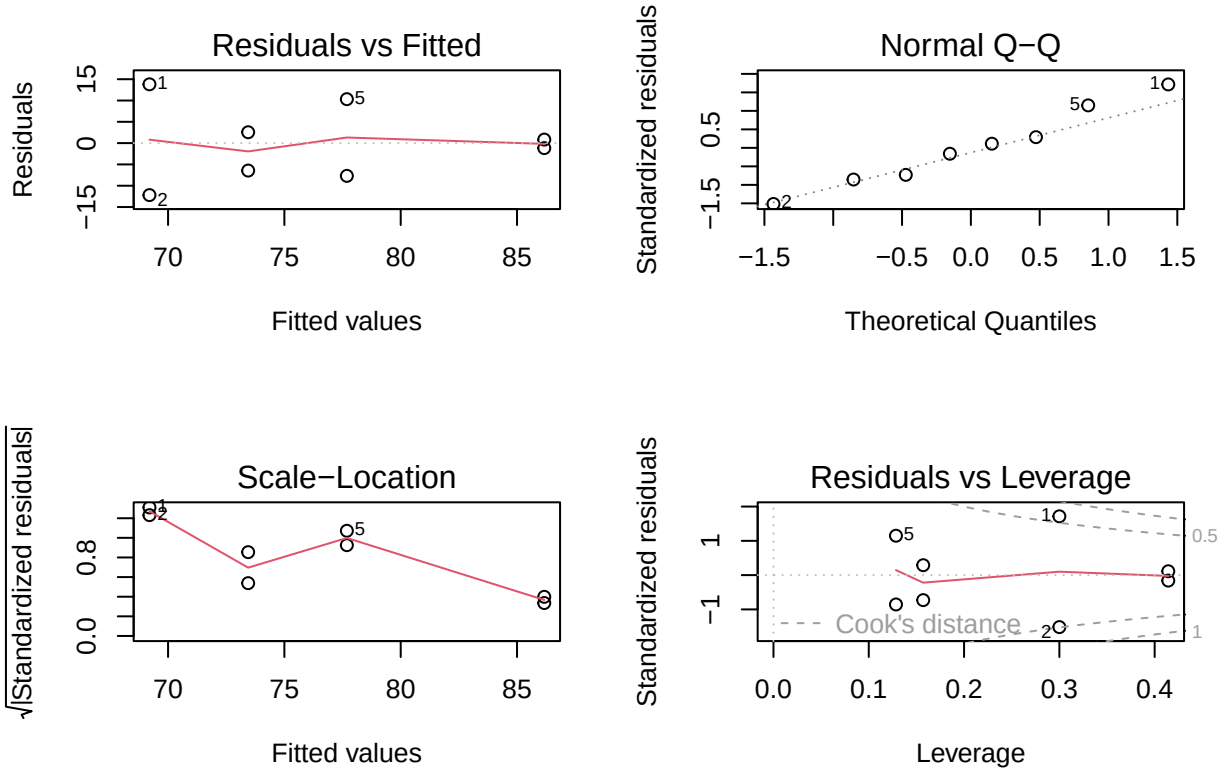
La predicción puede ser sobre un valor concreto, calculando además su intervalo de confianza.

```
predict(M1, newdata = data.frame(dose=4), interval = 'confidence')
```

```
##        fit      lwr      upr
## 1 81.92857 71.03615 92.82099
```

Visualmente la función plot(...), aplicada sobre el objeto que guarda el análisis de regresión, proporciona un interesante diagnóstico del modelo:

```
par(mfrow=c(2,2))
plot(M1)
```




```
par(mfrow=c(1,1))
```

Gráfico 1. El primer gráfico muestra si los residuos tienen patrones no lineales. Podría haber una relación no lineal entre las variables predictoras y la variable de resultado y el patrón debería aparecer en este gráfico. Si el gráfico muestra residuos igualmente distribuidos alrededor de la horizontal, se verfica la relación lineal.

Gráfico 2. Este gráfico muestra si los residuos se distribuyen normalmente. Esto se cumple cuando los residuos están bien alineados con la línea recta discontinua.

Gráfico 3. El objetivo de este gráfico es comprobar la hipótesis de homocedasticidad. El gráfico muestra cómo se distribuyen los residuos a lo largo de los rangos de los predictores. En el supuesto de igualdad de varianza (homocedasticidad) deberá dibujar una línea horizontal con puntos de distribución al azar (sin ningún patrón).

Gráfico 4. El último gráfico nos ayuda a encontrar casos influyentes (outliers). No todos los valores atípicos influyen en el análisis de regresión lineal. Esta vez los patrones no son relevantes. Buscamos la presencia de casos atípicos observando los puntos y las líneas discontinuas basadas en la distancia de Cook. Cuando los casos están fuera, éstos tienen potencial para influir en los resultados de la regresión. En otras palabras, los resultados del modelo se verán alterados si excluimos dichos casos.

Comprueba cómo se comporta el modelo cuando inluimos la variable “sex” en el vector de variables explicativas.

```
M2 <- lm(weight ~ dose + as.factor(sex), data=calcium)
summary(M2)
```

```
##
## Call:
## lm(formula = weight ~ dose + as.factor(sex), data = calcium)
##
## Residuals:
##       1       2       3       4       5       6       7       8
##  6.9250 -5.3250 -4.3179  0.4321  3.4393 -0.8107 -6.0464  5.7036
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       58.082      4.910  11.829  7.6e-05 ***
## dose               4.243      1.421   2.986   0.0306 *
## as.factor(sex)1   13.750      4.204   3.271   0.0222 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.945 on 5 degrees of freedom
## Multiple R-squared:  0.7968, Adjusted R-squared:  0.7156
## F-statistic: 9.806 on 2 and 5 DF,  p-value: 0.0186
```

Compara ambos modelos a partir del estadístico "Raíz del error cuadrático medio" (RMSE en inglés). Lo podemos hacer nosotros mismos:

```
RMSE.M1 <-sqrt(mean(M1$residuals^2))
RMSE.M2 <-sqrt(mean(M2$residuals^2))
```

o mediante la función rmse(...) disponible en el paquete "Metrics"

```
library(Metrics)
rmse(calcium$weight, predict(M1))
```

```
## [1] 8.327986
```

Finalizamos este primer ejemplo, incorporando una interacción entre las variables "sex" y "dose". ¿Consideras justificada esta interacción?

```
M3 <- lm(weight ~ dose*as.factor(sex), data=calcium)
summary(M3)
```

```
##
## Call:
## lm(formula = weight ~ dose * as.factor(sex), data = calcium)
##
## Residuals:
##       1       2       3       4       5       6       7       8
##  2.6000 -1.0000 -6.1714  2.2857  4.0571 -1.4286 -0.4857  0.1429
##
## Coefficients:
```

```
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             51.286      4.410  11.628 0.000313 ***
## dose                     6.714      1.412   4.754 0.008948 **
## as.factor(sex)1         27.343      6.237   4.384 0.011840 *
## dose:as.factor(sex)1    -4.943      1.998  -2.474 0.068618 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.178 on 4 degrees of freedom
## Multiple R-squared:  0.9197, Adjusted R-squared:  0.8595
## F-statistic: 15.28 on 3 and 4 DF,  p-value: 0.01175
```

Con R base podemos comparar los modelos mediante la función anova(...), que muestra un estadístico de contraste F para determinar si la diferencia entre modelos es estadísticamente significativa o no.

```
anova(M1, M2, M3)
```

```
## Analysis of Variance Table
##
## Model 1: weight ~ dose
## Model 2: weight ~ dose + as.factor(sex)
## Model 3: weight ~ dose * as.factor(sex)
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1      6 554.84
## 2      5 176.72  1    378.12 21.660 0.009632 **
## 3      4  69.83  1    106.89  6.123 0.068618 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ejercicio 1:

El dataset hiaa.dta (en formato Stata) incluye mediciones del examen A5-HIA en la orina en 40 pacientes, con el objeto de medir la cantidad de ácido 5-hidroxindolacético (A5-HIA). Este ácido es un producto de degradación de la serotonina.

1. Describa la dependencia de los dos parámetros de laboratorio ("hiaa" y "sero") con la edad y el sexo de los pacientes.
2. ¿Podemos concluir que la excreción urinaria de 5-HIAA varía con la edad?
3. ¿Cuál es la excreción esperada de 5-HIAA para un hombre de 49 años?
4. ¿Cuál es la diferencia esperada de la excreción urinaria de 5-HIAA entre dos mujeres que difieren en 13 años de edad? Calcula también el intervalo de confianza de dicha predicción.

Ejercicio 2: En un estudio sobre la actividad física de los escolares se obtuvieron medidas de 323 niños de entre 6 y 10 años. La muestra se dividió en niños de zonas rurales y urbanas. Fichero "physact.dta".

1. Calcula el valor medio del índice de actividad física en cada tipo de región.
2. ¿Puedes considerar la diferencia de índices estadísticamente significativa?
3. Sin embargo, el estudio se realizó seleccionando algunas escuelas y midiendo una muestra de niños de 1° a 4° grado dentro de cada escuela. El proceso de contacto con los niños tomó más tiempo en las escuelas rurales y, además, es posible que los niños de las áreas urbanas se incorporren al colegio a una edad más temprana. Verifica si podemos asumir que la media de edad de los niños es menor en los niños urbanos que en los rurales.

4. Trata de hacer una comparación más eficaz entre los niños rurales y urbanos con respecto a su actividad física media, teniendo en cuenta el desequilibrio en la distribución por edades.

5. ¿Deberíamos tener en cuenta también el género de los niños en este análisis?

La regresión por pasos

La regresión por pasos es un procedimiento ampliamente utilizado para seleccionar únicamente aquellas variables independientes que resultan "útiles" al modelo. A continuación mostramos una aplicación con la función step(...), a partir de los datos del ejercicio anterior.

```
library(haven)
phy <- read_dta("phyact.dta")

reg <- lm(phyact ~ age + sex + region, data=phy)
step(reg,diretion="both")
```

```
## Start:  AIC=-445.63
## phyact ~ age + sex + region
##
##          Df Sum of Sq     RSS     AIC
## - region  1     0.025  79.323 -447.53
## <none>                 79.298 -445.63
## - sex     1     7.269  86.568 -419.30
## - age     1    37.120 116.419 -323.61
##
## Step:  AIC=-447.53
## phyact ~ age + sex
##
##        Df Sum of Sq     RSS     AIC
## <none>               79.323 -447.53
## - sex   1     7.366  86.689 -420.85
## - age   1    39.676 118.999 -318.53

##
## Call:
## lm(formula = phyact ~ age + sex, data = phy)
##
## Coefficients:
## (Intercept)          age          sex
##      2.3278       0.2223       0.3021
```

El resultado nos señala como modelo preferido, aquel que sólo incorpora las variables age y sex.

Modelos de respuesta discreta

En el análisis de datos es frecuente encontrarse con variables dicotómicas (sí/no, presencia /ausencia),o variables medidas en escala ordinal (intervalos de edad, medidas de una prueba médica, etc.). Una práctica usual, es tratar este tipo de variables como si fueran continuas, asignándoles una puntuación arbitraria basada en la codificación de las distintas categorías de respuesta.

En los modelos de respuesta binaria o dicotómica, la variable de respuesta Y puede tomar dos valores, codificados usualmente como 1 para la categoría de interés y 0 para la otra. Presentaremos cómo estimar en R las especificaciones "logit" y "probit".

Modelo de regresión logística

Utilizaremos la función glm(...). Para ilustrar su manejo utilizamos el dataset "allergy1.dta". Nuestra variable de interés es "allergyc" que toma valor 1 si el niño ha desarrollado una alergia antes de los 6 años.

Carga los datos y explora su contenido.

```
library(haven)
alle <- read_dta("allergy1.dta")
names(alle)
```

```
## [1] "childnr"  "allergyc" "allergym" "smokem"
```

Veamos una tabla cruzada entre la alergia en la madre frente a la alergia en el niño.

```
with(alle, table(allergyc, allergym))
```

```
##         allergym
## allergyc   0   1
##        0 516 264
##        1 203 142
```

Ampliemos la tabla anterior añadiendo el riesgo relativo de que el niño padezca alergia, dependiendo de si la madre tiene o no alergia. Calculamos también los odds y el odds ratio.

```
TAB1 <-with(alle, table(allergym, allergyc))
TAB2 <- cbind(TAB1, Total = rowSums(TAB1))
TAB3 <- cbind(TAB2, Freq = TAB1[,2]/rowSums(TAB1)*100)
TAB4 <- cbind(TAB3, odds = c(TAB3[1,4]/(100-TAB3[1,4]),
                             TAB3[2,4]/(100-TAB3[2,4])))
TAB5 <- cbind(TAB4, OR = c(TAB4[2,5]/TAB4[1,5], NA))

TAB5
```

```
##     0   1 Total     Freq      odds       OR
## 0 516 203   719 28.23366 0.3934109 1.367219
## 1 264 142   406 34.97537 0.5378788       NA
```

Los resultados anteriores deben coincidir con los obtenidos en una regresión logística. Para estimar un modelo logit en R utilizamos la función glm(...), especificando link=logit.

```
Mod1 <- glm(allergyc ~ allergym, family=binomial(link = "logit"), data=alle)
summary(Mod1)
```

```
##
## Call:
## glm(formula = allergyc ~ allergym, family = binomial(link = "logit"),
##     data = alle)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -0.9278  -0.8146  -0.8146   1.4495   1.5904
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.93290    0.08285 -11.260   <2e-16 ***
## allergym     0.31278    0.13302   2.351   0.0187 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1386.9  on 1124  degrees of freedom
## Residual deviance: 1381.4  on 1123  degrees of freedom
## AIC: 1385.4
##
## Number of Fisher Scoring iterations: 4
```

Los coeficientes anteriores muestran el score, para obtener los odds ratio bastará con calcular la exponencial de los mismos.

```
exp(cbind(OR = coef(Mod1), confint(Mod1)))
```

```
##                    OR     2.5 %    97.5 %
## (Intercept) 0.3934109 0.3337724 0.4619307
## allergym    1.3672190 1.0527321 1.7737504
```

A continuación vamos a incorporar al modelo la variable "smokem" y lo voy a hacer como interacción con "allergy",

```
Mod2 <- glm(allergyc ~ allergym*smokem, family=binomial, data=alle)
summary(Mod2)
```

```
##
## Call:
## glm(formula = allergyc ~ allergym * smokem, family = binomial,
##     data = alle)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0816  -0.8993  -0.6694   1.4838   1.7921
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.38174    0.15082  -9.161  < 2e-16 ***
## allergym         0.53802    0.20155   2.669 0.007600 **
## smokem           0.68522    0.18131   3.779 0.000157 ***
## allergym:smokem -0.07107    0.28231  -0.252 0.801229
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 1386.9  on 1124  degrees of freedom
## Residual deviance: 1358.4  on 1121  degrees of freedom
## AIC: 1366.4
##
## Number of Fisher Scoring iterations: 4
```

Sobre el modelo anterior me interesa obtener el OR en las madres no fumadores y el OR en las madres fumadoras. Para las primeras es muy fácil. En las segundas es algo más complejo pero te doy una solución "imaginativa".

```
# OR madre no fumadora
exp(coef(Mod2)[2])
```

```
## allergym
##  1.71261
```

```
exp(confint(Mod2)[2,])
```

```
##    2.5 %   97.5 %
## 1.156360 2.551142
```

```
# OR madre fumadora
exp(sum(coef(Mod2)[c(2,4)])) # pero no sirve para el intervalo de confianza
```

```
## [1] 1.595114
```

```
alle$smokemr = 1-alle$smokem

Mod2b <- glm(allergyc ~ allergym*smokemr, family=binomial, data=alle)
exp(coef(Mod2b)[2])
```

```
## allergym
## 1.595114
```

```
exp(confint(Mod2b)[2,])
```

```
##    2.5 %   97.5 %
## 1.081082 2.348766
```

Otra alternativa es utilizar el paquete "epitools"

```
library(epitools)

oddsratio(as.factor(alle$allergym)[alle$smokem==0],
          as.factor(alle$allergyc)[alle$smokem==0])
```

```
## $data
##          Outcome
## Predictor   0   1 Total
```

```
##     0     219  55   274
##     1     186  80   266
##     Total 405 135   540
##
## $measure
##          odds ratio with 95% C.I.
## Predictor estimate    lower    upper
##         0 1.000000       NA       NA
##         1 1.709629 1.153461 2.548819
##
## $p.value
##          two-sided
## Predictor  midp.exact fisher.exact chi.square
##         0          NA           NA         NA
##         1 0.007475973  0.009610739 0.00728394
##
## $correction
## [1] FALSE
##
## attr(,"method")
## [1] "median-unbiased estimate & mid-p exact CI"
```

```
oddsratio(as.factor(alle$allergym)[alle$smokem==1],
          as.factor(alle$allergyc)[alle$smokem==1])
```

```
## $data
##          Outcome
## Predictor   0   1 Total
##     0     297 148   445
##     1      78  62   140
##     Total 375 210   585
##
## $measure
##          odds ratio with 95% C.I.
## Predictor estimate    lower    upper
##         0 1.000000       NA       NA
##         1 1.594211 1.079652 2.349217
##
## $p.value
##          two-sided
## Predictor midp.exact fisher.exact chi.square
##         0         NA           NA         NA
##         1 0.01914798   0.02018886 0.01767884
##
## $correction
## [1] FALSE
##
## attr(,"method")
## [1] "median-unbiased estimate & mid-p exact CI"
```

La función anova(...) de nuevo me permite valorar si el segundo modelo mejora al primero

```r
anova(Mod1, Mod2, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: allergyc ~ allergym
## Model 2: allergyc ~ allergym * smokem
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      1123     1381.4
## 2      1121     1358.4  2   23.048 9.888e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Al igual que en el modelo lineal existen una serie de funciones para obtener los valores predichos por el modelo. Utilizaremos predict(...)

```r
prob.ajustadas <- predict(Mod1, type = "response", se.fit = TRUE)

prob.ajustada_0 <- predict(Mod1, newdata = data.frame(allergym=0),
                           type = "response", se.fit = TRUE)

prob.ajustada_1 <- predict(Mod1, newdata = data.frame(allergym=1),
                           type = "response", se.fit = TRUE)
```

se.fit=TRUE me proporciona el error esándar de cada valor predicho, el cual utilizo para contruir los intervalos de confianza al 95% (valor crítico 1.96).

```r
intervalo_0 <- c(prob.ajustada_0$fit - 1.96 * prob.ajustada_0$se.fit,
                 prob.ajustada_0$fit + 1.96 * prob.ajustada_0$se.fit)


intervalo_1 <- c(prob.ajustada_1$fit - 1.96 * prob.ajustada_1$se.fit,
                   prob.ajustada_1$fit + 1.96 * prob.ajustada_1$se.fit)
```

Medidas de bondad del ajuste basadas en la clasificación. Curvas ROC

Con estas medidas buscamos conocer que éxito tiene mi modelo para predecir correctamente. Para ello cruzaremos la tabla de frecuencia de mi variable original (variable dependiente) con la asignación que realiza mi modelo respecto a un punto de corte.

```r
Tvi <- table(alle$allergyc)

Mod1.predi <- ifelse(fitted.values(Mod1) >=Tvi[2]/Tvi[1] , 1, 0)
table(Mod1.predi)
```

```
## Mod1.predi
##    0
## 1125
```

```r
table(alle$allergyc, Mod1.predi)
```

```
##    Mod1.predi
##       0
##   0 780
##   1 345
```

```
Tabla.cla <-prop.table(table(alle$allergyc, Mod1.predi))

total.bienp=Tabla.cla[1,1]
total.bienp
```

```
## [1] 0.6933333
```

prueba a obtener la misma tabla con "prob.ajustada_1$fit" como punto de corte.

Dos estadísticos importantes para añadir a los anteriores: la "Sensibilidad" y la "Especificidad". La sensibilidad=Verdaderos.positivos/(Verdaderos.positivos+Falsos.negativos) y la especificidad=Verdadero.negativos/(verdaderos.negat

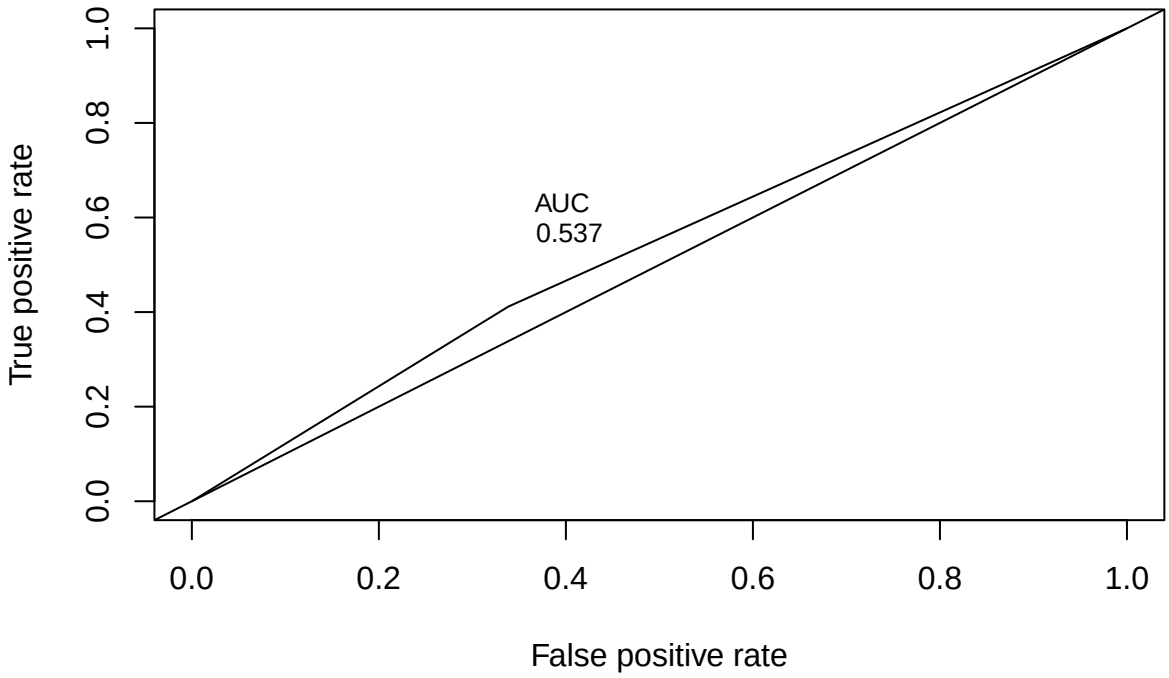
Finalmente, para dibujar la curva de ROC, utilizaremos el paquete "epitools" el cual facilita también el cálculo del área bajo la curva "AUC".

```
library(ROCR)
pred <- prediction(fitted.values(Mod1), as.factor(alle$allergyc))
roc.perf <- performance(pred, measure = "acc")

AUC <- performance(pred, measure = "auc")
AUC <- AUC@y.values[[1]]
AUC
```

```
## [1] 0.5365663
```

```
roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")
plot(roc.perf)
abline(a=0, b= 1)
text(0.4, 0.6, paste("AUC", "\n", round(unlist(AUC),3)), cex = 0.8)
```



Se considera que un modelo es mejor que otro si la curva ROC se acerca al borde superior izquierdo, o lo que es lo mismo, que el área bajo la curva sea mayor.

Ejercicio:

En el estudio mencionado anteriormente sobre el desarrollo de alergias en la primera infancia, también se registró información sobre el estado de tabaquismo y alergia del padre. Esta información se puede encontrar en el dataset "allergy2.dta".

1. Calcule el odds ratio entre el estado de alergia del padre y el estado de alergia del niño y ajuste ésta para el efecto del tabaquismo paterno.

2. ¿Qué sucede si ajusta el estado de alergia materna en lugar del tabaquismo paterno?

3. Si hace una tabulación cruzada entre el estado alérgico de la madre y el tabaquismo paterno (¡hazlo!), encontrará una tendencia a que las madres con alergia tienden a tener una pareja que no fuma. Si ajusta un modelo de regresión logística con alergia materna, tabaquismo materno y tabaquismo paterno (¡hazlo!), puede ver que el tabaquismo paterno influye en el estado de alergia del niño. Esto sugiere que el tabaquismo paterno es un factor de confusión para la asociación entre el estado de alergia de la madre y el estado de alergia del niño. Sin embargo, si eliminamos en el modelo anterior la covariable del tabaquismo paterno (¡hazo!), podemos observar que el odd ratio para la alergia materna no cambia.

4. Ajuste un modelo de regresión logística con las cuatro covariables. ¿Qué podemos concluir de este análisis con respecto al efecto de las covariables individuales? ¿Podemos decir algo sobre la covariable con el efecto más pequeño y más grande? ¿Cuál es la diferencia de probabilidad estimada de desarrollar una alergia entre un hijo de padres fumadores que ambos padecen alergias y un hijo de padres no fumadores y sin alergias?

5. Termina el ejercicio haciendo un análisis completo de la capacidad predictiva del modelo con cuatro covariables.

Ejercicio propuesto:

El conjunto de datos de dolor de espalda "backpain.dta" incluye datos de un estudio epidemiológico sobre la aparición de dolor de espalda. Hay información sobre la edad, el sexo y la clase social de los sujetos, su ocupación física al inicio del estudio y si padecían dolor de espalda al inicio (variable b0) y 5 años después (variable b5).

1. ¿Qué podemos concluir acerca de las diferencias entre los cuatro tipos de ocupaciones físicas al inicio del estudio con respecto al estado del dolor de espalda cinco años después, si ajustamos por edad y sexo?
2. ¿Qué podemos concluir sobre el efecto de la edad y el sexo? Trate de expresar la diferencia de sexo como una razón de probabilidades (OR).
3. Uno de los objetivos del estudio fue establecer que la ocupación física alta es más peligrosa que la ocupación física moderada con respecto al desarrollo del dolor de espalda. ¿Qué podemos concluir acerca de esta diferencia?
4. ¿Qué sucede si ajustamos por clase social?
5. Repita el análisis del apartado 1. considerando ahora el dolor de espalda al inicio del estudio como el resultado de interés.

Modelo de regresión probit

Utilizamos también la función glm(...) y el dataset "allergy1.dta". Carga los datos y explora su contenido.

```
library(haven)
alle <- read_dta("allergy1.dta")
```

Para estimar un modelo probit en R utilizamos la función glm(...), especificando link=probit.

```
Mod1 <- glm(allergyc ~ allergym, family=binomial(link = "probit"), data=alle)
summary(Mod1)
```

```
##
## Call:
## glm(formula = allergyc ~ allergym, family = binomial(link = "probit"),
##     data = alle)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9278  -0.8146  -0.8146   1.4495   1.5904
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.57591    0.04967 -11.595   <2e-16 ***
## allergym     0.18993    0.08095   2.346    0.019 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 1386.9  on 1124  degrees of freedom
## Residual deviance: 1381.4  on 1123  degrees of freedom
## AIC: 1385.4
##
## Number of Fisher Scoring iterations: 4
```

Ya eres un experto/a en análisis de datos con R

¿Te atrevés con datos reales? ¿Qué tal los datos para España de la última encuesta europea de salud?