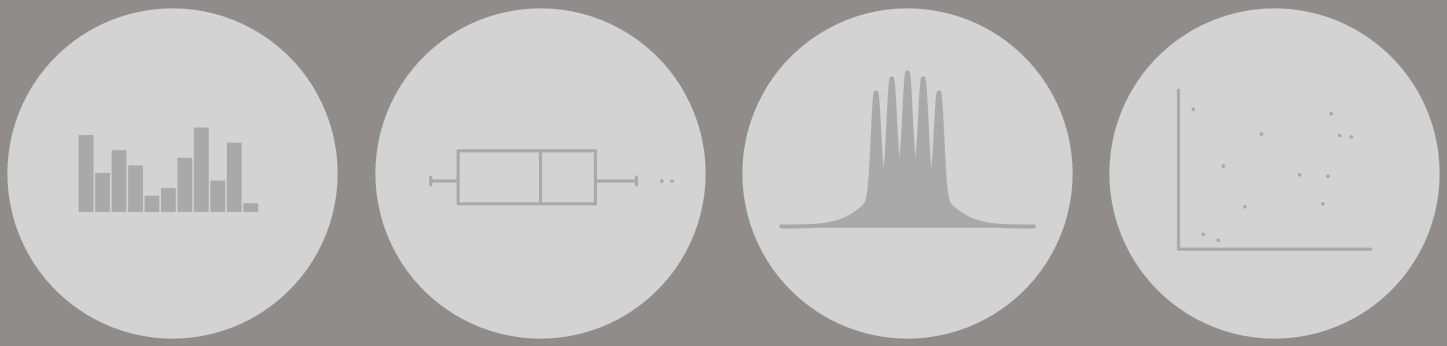
Manual de R para prácticas de Bioestadística



EDICIÓN A CARGO DE
María Alonso Pena
Diego Bolón Rodríguez
Jose Ameijeiras Alonso
Alejandro Saavedra Nieves
Paula Saavedra Nieves

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Manual de  para prácticas de Bioestadística

Manual de R para prácticas de Bioestadística

EDICIÓN A CARGO DE

María Alonso Pena
Diego Bolón Rodríguez
Jose Ameijeiras Alonso
Alejandro Saavedra Nieves
Paula Saavedra Nieves

2023
Universidade de Santiago de Compostela

The R Project for Statistical Computing
https://www.r-project.org/

The R Foundation
https://www.r-project.org/foundation/

Maqueta
María Alonso Pena
Diego Bolón Rodríguez
Jose Ameijeiras Alonso
Alejandro Saavedra Nieves
Paula Saavedra Nieves

Edita
Edicións USC
Campus Vida
15782 Santiago de Compostela
usc.gal/publicacions

DOI: https://dx.doi.org/10.15304/9788419679536

Preliminares

El software R, creado en 1993, se ha convertido en el programa informático más empleado en análisis y visualización de datos. Por ello, es fundamental tener un conocimiento de esta herramienta que, además, es libre y gratuita. La página web del proyecto R[1] contiene toda la información sobre este software y su descarga, varios manuales para su instalación o utilización y el listado completo de librerías o paquetes que permiten aumentar las funcionalidades básicas de R.

El principal objetivo de este manual es introducir R de forma sencilla y práctica en el ámbito de la Bioestadística. De hecho, este documento puede ser útil para cualquier usuario con conocimientos básicos de Bioestadística, que desee iniciarse al manejo de este software. La estructura del documento es la siguiente. En el Capítulo 1, se realiza una breve introducción a R con RStudio, un entorno de desarrollo integrado para R disponible para Windows, Mac y Linux o para navegadores conectados a RStudio Server o RStudio Server Pro. En general, la iniciación a R con RStudio resulta más atractiva e intuitiva para usuarios con conocimientos de programación escasos e incluso nulos. Los siguientes capítulos del manual se centran en abordar con R los contenidos de una materia de Bioestadística básica. Sin centrarse en la descripción de los aspectos estadísticos teóricos, se presentan con detalle las funciones básicas de R que deben emplearse en cada caso. Su uso se ilustra a través de casos prácticos de interés en el ámbito de la Medicina. Además, al final de cada capítulo se proponen varios ejercicios que el usuario puede resolver empleando las técnicas presentadas. El Capítulo 2 se ocupa de introducir con R los apectos fundamentales de la Estadística descriptiva. Las principales distribuciones de probabilidad (discretas y continuas) se presentan en el Capítulo 3. El Capítulo 4 se centra en la estimación (puntual o por intervalos) y de los contrastes de hipótesis para los parámetros poblacionales media y proporción. El contraste de independencia Chi-cuadrado y medidas de riesgo como el riesgo relativo y la odds ratio son presentadas en el Capítulo 5. Finalmente, en el Capítulo 6, se describe el ajuste con R del modelo de regresión lineal simple. A modo de resumen, el Apéndice A contiene la lista de las principales funciones de R utilizadas en este manual con una breve descripción de las mismas.

Las bases de datos empleadas en los casos prácticos que se resuelven en el manual, están disponibles en este repositorio[2].

[1] http://www.r-project.org/

[2] https://github.com/jose-ameijeiras/bioestadistica

Contenidos

1 Introducción a R con RStudio

Este capítulo consiste en una pequeña guía para iniciarse en el uso de R y RStudio. En él, se presentan los conceptos básicos de estos dos programas, así como todas las herramientas que necesitaremos en los siguientes capítulos para poder realizar análisis de datos descriptivo e inferencial con R. En la Sección 1.1, veremos una introducción al lenguaje R y al editor RStudio. Las Secciones 1.2, 1.3 y 1.4 contienen, respectivamente, información sobre la creación de vectores, matrices y bases de datos. En la Sección 1.5, veremos como aumentar las funcionalidades de R mediante la instalación de paquetes. La Sección 1.6 se centra en cómo cargar bases de datos externas. Finalmente, en la Sección 1.7 se propone un ejercicio para reforzar los conceptos introducidos en este capítulo.

1.1 Primeros pasos con R y RStudio

La instalación básica de R es totalmente funcional, pero no tiene la misma interfaz gráfica en todos los sistemas operativos. Es decir, la apariencia del programa y la forma de usarlo es diferente en Windows que en MacOS y en Linux. Por esta razón, este manual utiliza RStudio.

RStudio es un editor de código gratuito especializado en R que tiene la misma apariencia en todos los sistemas operativos. Además, posee varias características que lo hacen muy popular, como el resaltado de sintaxis y el autocompletado de código. Podéis descargar la versión gratuita de RStudio desde su página oficial[3].

La Figura 1 contiene la ventana que aparece al abrir RStudio por primera vez.

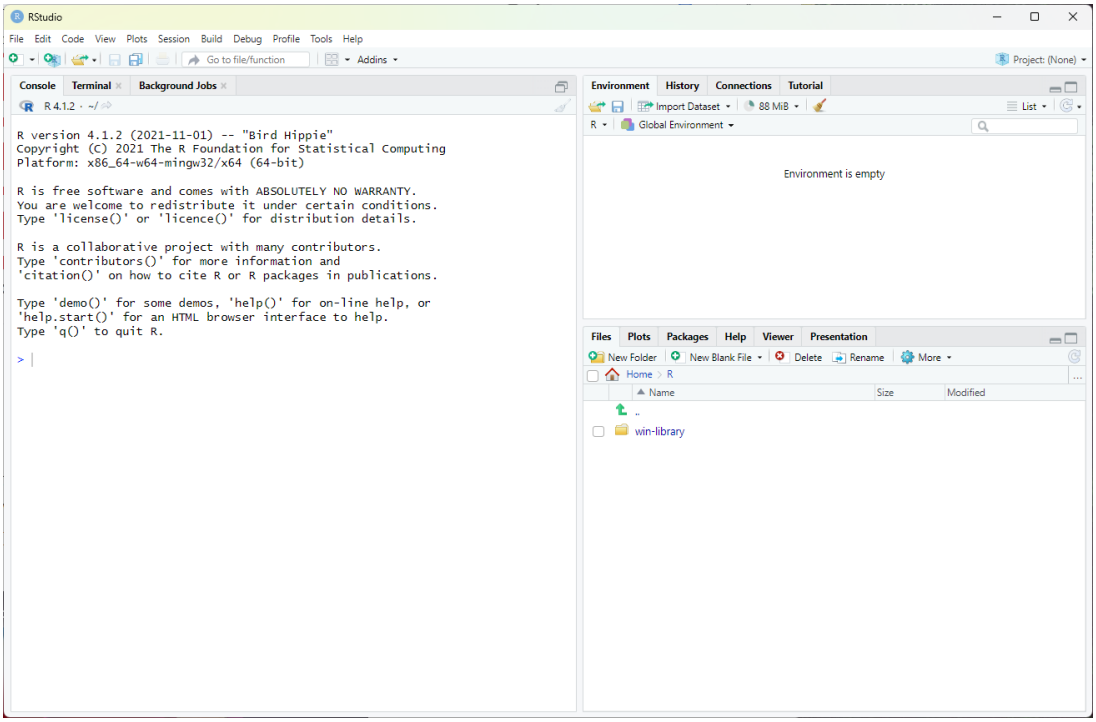


Figura 1: *Captura de pantalla de RStudio donde se aprecian tres paneles diferentes.*

El programa se divide en 3 paneles distintos: uno grande a la izquierda y dos más pequeños

[3] https://posit.co/downloads/

a la derecha. El panel de la izquierda se denomina *consola*. La consola es el espacio en el cual se ejecutan las órdenes que damos a R a través de líneas de código. Si después del símbolo > escribimos cualquier operación y se pulsa la tecla intro, R la ejecutará y devolverá el resultado de la misma. Algunos ejemplos:

```
> 2+2
[1] 4
> 10-2
[1] 8
```

Con R se pueden realizar todas las operaciones de cálculo básicas: la suma (+), la resta (−), el producto (∗), la división (/) y la exponenciación (ˆ). Por ejemplo, para calcular cinco al cuadrado escribimos:

```
> 5^2
[1] 25
```

Y siete entre cinco sería:

```
> 7/5
[1] 1.4
```

El software R **utiliza el punto para separar la parte entera de la parte decimal**. Si intentamos introducir un número con una coma en R obtendremos un mesaje de error:

```
> 1,4
Error:  unexpected ',' in "1,"
```

Si en una misma línea se combinan varias operaciones, R las efectúa en el orden natural: primero potencias, luego productos y divisiones, y por último sumas y restas. Para cambiar el orden de las operaciones es necesario añadir paréntesis.

```
> 2+4*5
[1] 22
> (2+4)*5
[1] 30
```

El programa R no se limita a la realización de operaciones básicas. En general, todas las operaciones que se pueden resolver con una calculadora científica también se pueden realizar con R . Algunos ejemplos son: el logartimo natural (con `log`), la función exponencial (`exp`), y las raíces cuadradas (`sqrt`). Por ejemplo, podemos saber cuánto valen $\ln 2$, e^3 y $\sqrt{5}$ ejecutando en la consola de R:

```
> log(2)
[1] 0.6931472
> exp(3)
[1] 20.08554
> sqrt(5)
[1] 2.236068
```

Si un resultado es muy grande o muy pequeño, R lo expresará en **notación científica**.

```
> 2000^10
[1] 1.024e+33
> exp(-34)
[1] 1.713908e-15
```

En los ejemplos anteriores, `e` es un símbolo que significa "*por diez elevado a*". Por tanto, los resultados de las dos operaciones son $1.024 \cdot 10^{33}$ y $1.713908 \cdot 10^{-15}$ respectivamente. También podemos emplear notación cientifica para introducir números en R:

```
> 2e+4
[1] 20000
> 1.5e-3
[1] 0.0015
```

1.1.1 Asignación de valores

En R, podemos almacenar un valor o varios en memoria para utilizarlos más tarde. Esto se hace empleando el símbolo =. Por ejemplo, si escribimos

```
> a=5
```

le estamos indicando a R "*guarda el valor* `5` *con el nombre* `a`". Así que R ha creado un nuevo objeto con nombre `a` y le ha asignado el valor `5`. Podéis saber el valor que tiene asignado un objeto introduciendo su nombre en la consola. Por ejemplo, tecleando

```
> a
[1] 5
```

R nos devuelve el valor asignado a `a`.

Podemos crear tantos objetos como queramos y operar con ellos sin ningún problema tal y como se muestra a continuación:

```
> b=10; c=3
> a+b; b/a; (a+b)/c
[1] 15
[1] 2
[1] 5
```

En el ejemplo anterior hemos empleado el punto y coma (`;`) para escribir varias órdenes en la misma línea.

En los manuales o tutoriales de R disponibles en internet suele emplearse el símbolo <- para crear objetos. En realidad, los símbolos = y <- son prácticamente equivalentes. En esta guía, emplearemos siempre el símbolo = por comodidad, pero podrían usarse cualquiera de los dos.

1.1.2 Comparaciones lógicas

En R, podemos comparar dos valores empleando, por ejemplo, el símbolo <.

```
> 2<5
[1] TRUE
```

En el ejemplo anterior el resultado es `TRUE` porque 2 es menor que 5. Si la comparación que le indicamos a R es falsa, el resultado será `FALSE`.

```
> 4<1
[1] FALSE
```

En la siguiente tabla, aparecen recogidos todos los símbolos para comparar dos valores.

Símbolo	Significado
<	Menor que
<=	Menor o igual que
>	Mayor que
>=	Mayor o igual que
==	Igual que
!=	Distinto que

Algunos ejemplos de su uso:

```
> 3>2
[1] TRUE
> 3>3
[1] FALSE
> 3>=3
[1] TRUE
> -1<=0
[1] TRUE
> 10==2*5
[1] TRUE
> 1!=(2-1)
[1] FALSE
```

No debemos confundir los símbolos = y ==. El primero sirve para crear un objeto. El segundo es una operación que nos devuelve `TRUE` si los dos valores son iguales y `FALSE` en caso contrario.

1.1.3 Cómo guardar el trabajo realizado: *scripts* y comentarios

Se puede utilizar R ejecutando las órdenes línea a línea en la consola tal y como hemos hecho hasta ahora, pero esto tiene varios problemas. Por ejemplo, introducir las operaciones en una línea puede ser muy engorroso si éstas son complicadas. Además, de esta forma no podemos guardar nuestro trabajo y lo perderemos cuando cerremos la sesión de R en nuestro ordenador.

Por eso, lo más habitual es trabajar mediante *scripts*. Un *script* es simplemente un archivo de texto donde guardamos todas las operaciones que queremos que R ejecute. Las operaciones que hay dentro de un script se denominan habitualmente *código*. Cuando queramos ejecutarlas solo tenemos que copiarlas y pegarlas en la consola.

RStudio nos permite crear y editar scripts de R de forma fácil y cómoda. Si en la barra de menús accedéis a *File* > *New File* > *R Script*, RStudio abrirá un nuevo panel con un script en blanco tal y como aparece en la Figura 2. En él, podéis escribir tantas operaciones de R como queráis y guardarlas para poder ejecutarlas en cualquier momento. A continuación, copiaremos en

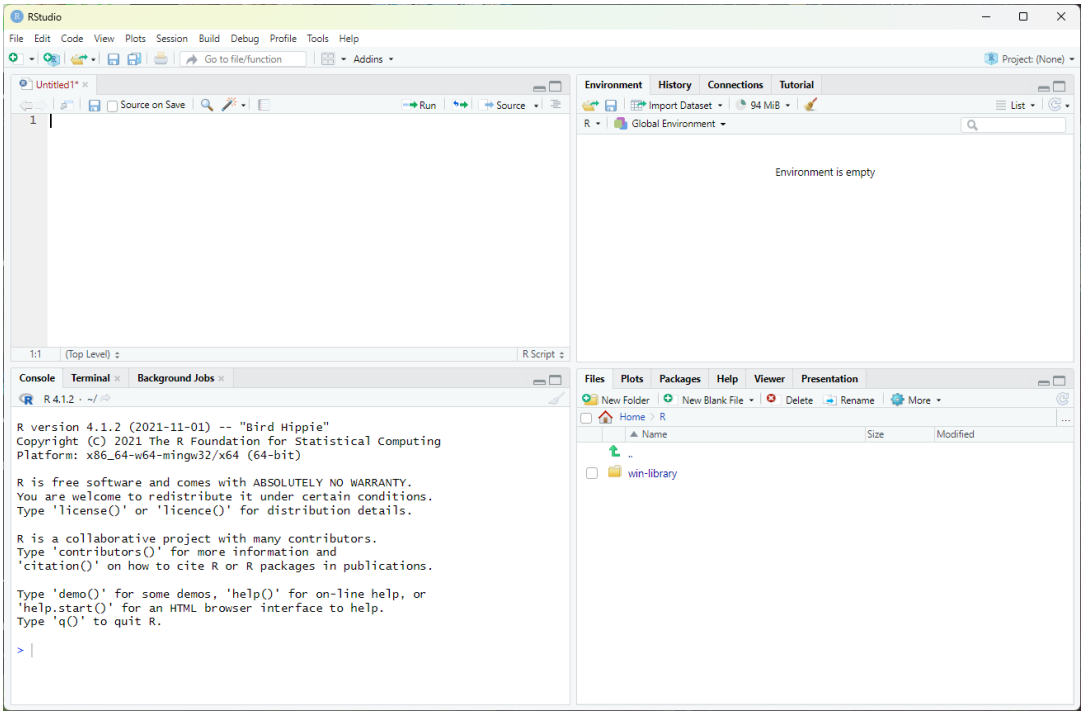


Figura 2: *Captura de pantalla de RStudio con un script en blanco abierto en la esquina superior izquierda. En el nuevo panel podemos escribir todas las operaciones que deseemos para guardarlas y ejecutarlas en cualquier momento.*

el script el código que aparece en el siguiente recuadro.

```
# PRÁCTICA 1: INTRODUCCIÓN A R
# Creamos objetos con el símbolo =
a=5; b=10; c=3
# Podemos operar con ellos en cualquier momento
b-a
a/b
(b+a)/c
```

De este modo, podemos guardar todo lo que hemos aprendido con R hasta el momento. Para ejecutar el código de un script simplemente tenemos que copiarlo, pegarlo en la consola y pulsar la tecla intro.

```
> # PRÁCTICA 1:  INTRODUCCIÓN A R
> # Creamos objetos con el símbolo =
> a=5; b=10; c=3
> # Podemos operar con ellos en cualquier momento
> b-a
[1] 5
> a/b
[1] 0.5
> (b+a)/c
[1] 5
```

En el ejemplo anterior, hay tres líneas de código que empiezan con el símbolo #. Estas líneas

se llaman *comentarios* y sirven para explicar y aclarar lo que estamos haciendo en cada momento. Son muy útiles, sobre todo cuando los scripts son muy grandes o complejos.

> 👁 Copiar y pegar nuestro código en la consola puede ser bastante tedioso. Para evitar esto, RStudio tiene un atajo de teclado que nos permite **ejecutar la línea de código seleccionada automáticamente**. Este atajo es ligeramente distinto en cada sistema operativo:
>
> - **Windows y Linux**: Ctrl + Enter.
> - **MacOS**: Cmd + Enter.

1.2 Vectores

Hasta ahora, solo hemos trabajado con objetos con un único valor asignado. Sin embargo, es posible crear objetos que contengan más de un valor. Los más simples son los *vectores* y se crean con el comando el siguiente comando:

```
c(...)
```

Los elementos del vector deben especificarse dentro de los paréntesis y separados por comas.

Todas las operaciones básicas de R funcionan también con vectores. Al operar un número con un vector R realiza la operación sobre todos los elementos del vector. Esto incluye a la suma $(+)$, la resta $(-)$, el producto $(*)$, la división $(/)$, la exponenciación $(\hat{})$ y todas las comparaciones. Si comparamos un vector con un número, R comparará todos los elementos del vector con el número y nos proporcionará el resultado componente a componente.

Además, muchas funciones básicas de R actúan elemento a elemento. Por ejemplo: `log` calcula el logaritmo natural de todos los elementos del vector y `sqrt` calcula las raíces cuadradas de cada elemento.

Caso práctico: Utiliza R para responder a las siguientes preguntas:

(a) Crea un vector en R que se llame `x` que contenga tres elementos: 3, 5 y 1.

```
x=c(3, 5, 1)
x
```

(b) Súmale 10 a todos los elementos de `x`. Divide todos los elementos de `x` entre 3.

```
x+10; x/3
```

El resultado de `x + 10` es $13, 15$ y 11. El de `x/3` es $1, 1.67$ y 0.33.

(c) Calcula la raíz cuadrada de todos los elementos de `x`.

```
sqrt(x)
```

La raíz cuadrada de los elementos de `x` es $1.732, 2.236$ y 1.

1.2.1 Cómo acceder a los elementos de un vector

Podemos acceder a un elemento de un vector indicando la posición del elemento entre corchetes.

```
vector[...]
```

Si queremos acceder a varios elementos a la vez tenemos que poner dentro de los corchetes un vector con las posiciones de cada elemento.

```
vector[c(...)]
```

Dentro de los corchetes también podemos poner una comparación.

```
vector[comparacion]
```

En este caso, R nos devolverá todos los elementos del vector que cumplan la condición pedida.

Caso práctico (continuación):

(d) Accede al segundo elemento de `x`. Recupera el primer y el último elemento `x`.

```
x[2]
x[c(1, 3)]
```

(e) Accede a los elementos de `x` mayores que 3.

```
x[x>3]
```

R sólo nos devuelve el 5, que es el único elemento de `x` mayor que 3.

1.2.2 *Length*, *sum* y *sort*

R tiene muchas funciones para trabajar con vectores. Algunas de las más comunes son:

- La función `length`, que devuelve el número de elementos de un vector.
- La función `sum` suma todos los elementos de un vector.
- Las funciones `max` y `min`, que devuelven el mínimo y el máximo del vector.
- La función `sort`, que ordena los elementos de mayor a menor. Si queremos ordenar un vector de mayor a menor, podemos hacerlo con `sort` añadiendo la opción `decreasing=TRUE`.

Caso práctico (continuación):

(f) ¿Cuál es la suma de todos los elementos de `x`? ¿Cuál es su máximo y su mínimo? Ordena `x` de mayor a menor.

```
sum(x)
max(x); min(x)
sort(x, decreasing=TRUE)
```

La suma de los elementos de x es 9. Su máximo es 5 y su mínimo es 1. Ordenado de mayor a menor, el vector queda $(5,3,1)$.

Todas las funciones de R tienen una página de ayuda que podemos consultar siempre que tengamos dudas sobre su utilización. Para acceder a ella, empleamos el comando `?nombre_función` o `help(nombre_función)`. Por ejemplo, podemos acceder a la ayuda de `length` con

```
> ?length
```

o con

```
> help(length)
```

1.2.3 Operaciones con dos vectores

Todas las operaciones básicas de R también funcionan entre dos vectores. Si operamos dos vectores entre sí, esa operación se realiza siempre elemento a elemento. Esto incluye también a las comparaciones.

Caso práctico (continuación):

(g) Crea el vector y formado por los elementos 2, 5, y 10. Realiza las operaciones x+y y x/y.

```
y=c(2, 5, 10)
x+y; x/y
```

(h) ¿Cuántos elementos de x son menores o iguales que el elemento correspondiente en y?

```
x<=y
```

El primer elemento de x es mayor que el primero de y, los otros dos son siempre menores o iguales.

Cuando operamos o comparamos dos vectores tenemos que tener cuidado de que **tengan la misma longitud**. Si son de distinto tamaño, R repetirá el vector más pequeño desde el principio hasta que tenga el tamaño del más grande.

```
> a=c(3, 5, 1); b=c(0, 1)
> a+b
   [1] 3 6 1
Warning message:
In a + b :  longer object length is not a multiple of shorter
object length
```

1.2.4 Vectores de caracteres

Hasta ahora solo hemos visto vectores *numéricos*, es decir, vectores formados por números. Pero R nos permite hacer vectores con otro tipo de elementos. Por ejemplo, podemos crear vectores cuyos elementos son palabras siempre que estas aparezcan entrecomilladas.

```
> colores=c("rojo", "verde", "azul")
> colores
[1] "rojo" "verde" "azul"
```

Evidentemente, no podemos tratar estos vectores como si fueran números. Si lo hacemos, R proporciona un mensaje de error.

```
> colores+2
Error in colores + 2 :  non-numeric argument to binary operator
> sum(colores)
Error in sum(colores) :  invalid 'type' (character) of argument
```

Sin embargo, podemos utilizar la función `length()` con ellos como con cualquier otro vector.

```
> length(colores)
[1] 3
```

Caso práctico: En un hospital, registraron la talla y el peso de todos los bebés nacidos durante una mañana, obteniendo:

Talla (en cm)	44	52	51	50	48	55
Peso (en gramos)	3128	4022	3880	3755	3382	4628

Utiliza R para responder a las siguientes preguntas.

(a) Crea un vector en R que se llame `talla` y otro que se llame `peso` que contengan la talla y el peso de los recién nacidos respectivamente.

```
talla=c(44, 52, 51, 50, 48, 55)
peso=c(3128, 4022, 3880, 3755, 3382, 4628)
talla; peso
```

(b) Accede al peso del segundo bebé.

```
peso[2]
```

El peso del segundo bebé es de 4022 gramos.

(c) Recupera la talla de los bebés que pesaron más de 4000 gramos. ¿Cuántos hay?

```
talla[peso>4000]
length(talla[peso>4000])
```

Hay dos: uno mide 52 cm y otro 55 cm.

(d) Crea un nuevo vector que se llame `peso.kg` con el peso expresado en kilogramos.

```
peso.kg=peso/1000; peso.kg
```

El peso de los bebés en kg es: 3.128, 4.022, 3.880, 3.755, 3.382 y 4.628

1.3 Matrices

En R podemos emplear el comando `matrix` para crear una matriz. Para ello tenemos que indicar los elementos de la matriz y el número de filas y columnas.

```
matrix(data, nrow, ncol)
```

donde

- `data` es un vector con los elementos de la matriz.
- `nrow` el número de filas.
- `ncol` el número de columnas.

Por ejemplo, vamos a crear una matriz A con 3 filas y 2 columnas formada por los números del 1 al 6.

```
> A=matrix(c(1, 2, 3, 4, 5, 6), nrow=3, ncol=2)
> A
[1,] 1 4
[2,] 2 5
[3,] 3 6
```

Los elementos de una matriz se introducen por columna. Teclearemos en la consola `B=matrix(c(1, 3, 5, 2, 4, 6), nrow=3, ncol=2)` y observaremos las diferencias entre la matriz A y la matriz B.

1.3.1 Operaciones con matrices

Las operaciones básicas entre matrices también están implementadas en R. Podemos sumar y restar matrices sin problema con los símbolos habituales $+$ y $-$. Sin embargo, R tiene dos símbolos distintos para la multiplicación de matrices. Si operamos dos matrices con el símbolo $*$ **la multiplicación se hará elemento a elemento**. Si lo que queremos es calcular el producto matricial, tenemos que utilizar el comando `%*%`.

1.3.2 Acceder a los elementos de una matriz

Al igual que en los vectores, podemos acceder a los elementos de una matriz empleando los corchetes. Para eso tenemos que indicar el número de fila y el número de columna (en ese orden y separados por una coma) del elemento al que queremos acceder.

```
A[numero_fila, numero_columna]
```

Si queremos acceder a una fila concreta de la matriz, podemos hacerlo con los corchetes indicando solo el índice de fila y dejando vacío el de columna.

```
A[numero_fila, ]
```

Para acceder a las columnas hacemos lo mismo: dejamos vacío el índice de la fila y ponemos el número de la columna que queremos.

```
A[, numero_columna]
```

Caso práctico: Realiza las siguientes operaciones con R.

(a) Crea las siguientes tres matrices en R:

$$A = \begin{pmatrix} 2 & -1 & 0 \\ 1 & 3 & 1 \\ 0 & 2 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 2 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 2 & 0 \\ -1 & 0 & 3 \end{pmatrix}.$$

```
A=matrix(c(2, 1, 0, -1, 3, 2, 0, 1, 2), nrow=3, ncol=3)
B=matrix(c(1, 0, -1, 1, 1, 0, 1, 0, 2), nrow=3, ncol=3)
C=matrix(c(1, -1, 2, 0, 0, 3), nrow=2, ncol=3)
A; B; C
```

Recuerda que los elementos de cada matriz se introducen **columna por columna.**

(b) Accede a la segunda columna de la matriz A, la tercera fila de B, y al elemento de C que está en la primera fila y segunda columna.

```
A[, 2]
B[3, ]
C[1, 2]
```

(c) Realiza las siguientes operaciones: $A+B$, $B-A$, CB.

```
A+B
B-A
C%*%B
```

Recuerda que para la multiplicación matricial se emplea el símbolo especial `%*%`.

1.4 Bases de datos (`data.frame`)

Las matrices en R tienen una limitación importante: no podemos combinar distintos tipos de datos (números y caracteres) en una misma matriz. Esto es un problema, ya que muchas veces tendremos que trabajar con bases de datos que combinan variables cualitativas, como pueden ser el sexo o el grupo sanguíneo, con cuantitativas, como la edad o el nivel de glucosa en sangre.

Para combinar variables de distinto tipo dentro de un mismo objeto podemos usar lo que se conoce como un `data.frame`. Un `data.frame` es una colección de vectores de la misma longitud. Podemos crear un `data.frame` con el comando del mismo nombre:

```
data.frame(...)
```

Dentro de los paréntesis introducimos los distintos vectores separados por comas.

Caso práctico: Registramos la altura, el peso y el sexo de varios pacientes, obteniendo lo siguiente:

Altura (en cm)	175	166	162	155	191
Peso (en kg)	80	62	75	59	107
Sexo	H	M	H	M	H

(a) Crea un `data.frame` con el nombre `pacientes` que contenga toda la información de los pacientes.

```
pacientes=data.frame(altura=c(175, 166, 162, 155, 191),
peso=c(80, 62, 75, 59, 107), sexo=c("H", "M", "H", "M", "H"))
pacientes
```

1.4.1 Acceder las filas y columnas de un `data.frame`

Podemos acceder a las filas, a las columnas y a cada elemento de un `data.frame` empleando los corchetes como si fuera una matriz. También podemos acceder a cada columna del `data.frame` mediante el símbolo `$` seguido por el nombre de la columna.

```
data$nombre_columna
```

Cada columna de un `data.frame` es un vector. Por tanto, podemos realizar con ellas todas las operaciones habituales de un vector.

1.4.2 Funciones especiales para objetos `data.frame`

Las funciones más comunes a la hora de trabajar con un `data.frame` son:

- Las funciones `nrow()` y `ncol()`, que devuelven el número de filas y el número de columnas del `data.frame`.
- La función `names()` devuelve el nombre de todas las columnas.
- La función `head()` muestra por pantalla las 6 primeras filas del `data.frame`. Es muy útil cuando estamos trabajando con bases de datos muy grandes y queremos hacernos una idea de lo que contiene.

Caso práctico (continuación):

(b) ¿Cuántos pacientes se han estudiado? ¿Qué variables se han registrado a cada paciente?

```
nrow(pacientes)
names(pacientes)
```

Hay cinco pacientes en total. Las variables registradas son: altura, peso y sexo.

(c) Obtén la altura del tercer paciente y el peso del segundo.

```
pacientes$altura[3]
pacientes$peso[2]
```

Este ejercicio también se puede resolver usando solo los corchetes. Así, con el siguiente código también obtendríamos la altura del tercer paciente y el peso del segundo, que son 162 cm y 62 kg respectivamente.

```
pacientes[3, 1]
pacientes[2, 2]
```

(d) Ordena los pesos de los pacientes de menor a mayor.

```
sort(pacientes$peso)
```

1.5 Paquetes

Como ya hemos visto, R posee varias funciones que nos permiten realizar una gran cantidad de operaciones. Sin embargo, habrá problemas en los que la versión básica de R no tiene una función específica para resolverlos. Esto se puede solucionar instalando librerías o paquetes (*packages* en inglés). Los paquetes son archivos que contienen funciones extra, aumentando así la funcionalidad de R.

Para instalar un nuevo paquete utilizamos la función

```
install.packages("nombre_paquete")
```

donde `nombre_paquete` es el nombre del paquete que queremos instalar.

Para utilizar un paquete no tenemos solo que instalarlo, sino que también tendremos que cargarlo cada vez que lo queramos usar. Esto se hace con la función

```
library(nombre_paquete)
```

Para obtener información sobre un paquete podemos hacerlo desde la consola empleando el comando

```
help("nombre_paquete")
```

o bien descargar la ayuda en pdf del paquete disponible online[4].

Caso práctico: Realiza en R lo siguiente:

(a) Instala y carga el paquete `readxl`.

```
install.packages("readxl")
library(readxl)
```

(b) Accede a la ayuda del paquete `readxl`.

```
help("readxl")
```

Ejecutando lo anterior se abre una página de ayuda titulada *readxl: Read Excel Files*.

[4] https://cran.r-project.org/web/packages/available_packages_by_name.html

1.6 Cargar datos con R

Hasta ahora, hemos introducido todos los datos en R directamente empleando los comandos `c()` y `data.frame()`. Esto era factible porque todos los ejemplos considerados tenían pocos datos. Pero, en investigación biomédica resulta habitual trabajar con datos de cientos e incluso miles de pacientes. Por ejemplo, en el estudio realizado para comprobar la eficacia de la vacuna de AstraZeneca[5] contra el SARS-CoV-2 se analizaron datos de 11.636 personas.

En casos como ese no tiene ningún sentido teclear todos los datos de forma manual. Dado que R es compatible con distintos formatos de bases de datos (por ejemplo, excel, csv, json o sav) una vez almacenados esos datos en un archivo, es necesario cargarlos en R cada vez que queramos trabajar con ellos. El archivo que contiene los datos puede ser de muchos tipos: un .txt, una tabla de Excel .csv, un archivo .json,... R tiene una función distinta para cargar cada tipo de archivo. En este capítulo, vamos a ver dos: `read.table` y `read.csv`. Pero antes, tenemos que ver cómo modificar el directorio de trabajo.

1.6.1 Cambiando el directorio de trabajo

Cada vez que iniciamos R, este escoge una carpeta para usar como *directorio de trabajo*. Es en esta carpeta donde R va a buscar todos los archivos de datos que queramos cargar. Por tanto, cada vez queramos cargar una base de datos tenemos que asegurarnos de que esté en el directorio de trabajo.

Podemos cambiar el directorio de trabajo mediante la barra de menús de RStudio, siguiendo la ruta *Session* > *Set Working Directory* > *Choose Directory...* como se muestra en la Figura 3. Así, se abrirá una nueva ventana donde podremos escoger la carpeta que queráis usar como directorio de trabajo. Una vez realizado el cambio el contenido del nuevo directorio de trabajo aparecerá en el panel *Files* de RStudio tal como aparece en la Figura 4.

> También es posible cambiar el directorio de trabajo empleando la consola. Para ello, se emplea la función `setwd(...)` indicando dentro de los paréntesis la ruta de la carpeta que queramos usar como directorio de trabajo. De hecho, cuando cambiamos el directorio de trabajo a través de los menús de RStudio se ejecuta esta función automáticamente, como se puede ver en la Figura 4.

1.6.2 La función `read.table`

La función `read.table` es la que se usa normalmente para archivos de datos tipo texto (con extensión .txt). Esta función tiene muchos argumentos de entrada que describiremos en detalle a continuación:

```
read.table(file, header, sep, dec)
```

donde

[5] El artículo original puede consultarse en el siguiente enlace: https://doi.org/10.1016/S0140-6736(20)32661-1.

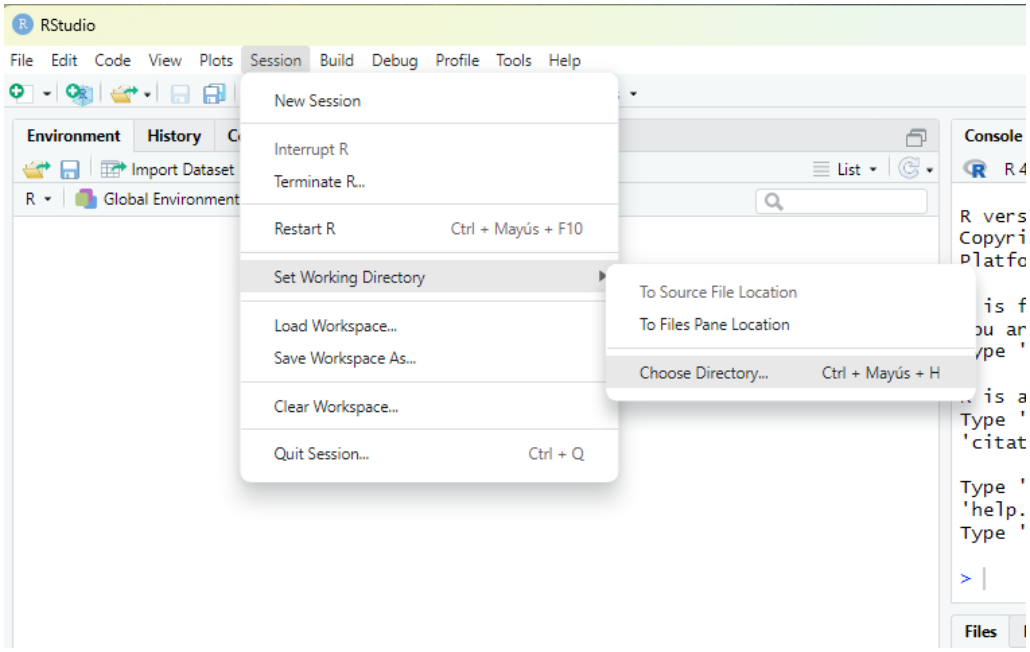


Figura 3: *Captura de pantalla con la ruta a seguir para cambiar el directorio de trabajo de* R.

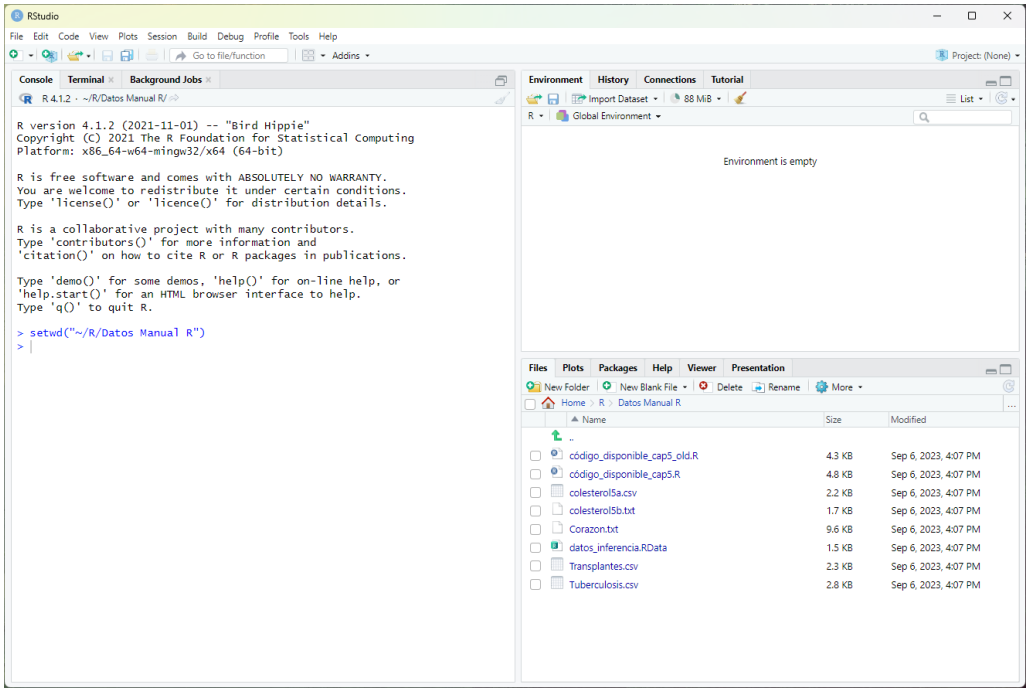


Figura 4: *Captura de pantalla de RStudio con el nuevo directorio de trabajo. En el panel Files (esquina inferior derecha) podemos ver el contenido de la carpeta que hemos escogido como directorio de trabajo.*

`file`	es el nombre del archivo que queremos abrir.
`header`	con esta opción indicamos si la primera fila del archivo se corresponde (`TRUE`) o no (`FALSE`) con una cabecera con los nombres de las variables. Por defecto vale `FALSE`.
`sep`	el símbolo que separa cada columna del archivo. Los mas habituales son el espacio, la coma y el punto y coma. Su valor por defecto es un espacio.
`dec`	el símbolo de separación decimal. Por defecto es un punto.

Caso práctico: El archivo *corazon.txt* contiene información de varios pacientes afectados con enfermedades cardiovasculares. A continuación, podemos ver una previsualización de las primeras diez filas del archivo:

corazon.txt

```
"Edad" "Sexo" "Dolor" "Presion" "Colesterol" "Ritmo_cardiaco_max"
63 TRUE 1 145 233 150
67 TRUE 4 160 286 108
67 TRUE 4 120 229 129
37 TRUE 3 130 250 187
41 FALSE 2 130 204 172
56 TRUE 2 120 236 178
62 FALSE 4 140 268 160
57 FALSE 4 120 354 163
63 TRUE 4 130 254 147
(...)
```

De cada paciente, se registraron:

`Edad`	la edad del paciente en años.
`Sexo`	el sexo del paciente: `TRUE` indica que el paciente es un hombre, `FALSE` indica que es mujer.
`Dolor`	el tipo de dolor que presenta el paciente. Puede ser: angina común (1), angina atípica (2), dolor no anginal (3) o asintomático (4).
`Colesterol`	el colesterol en sangre en mg/dl.
`Ritmo_cardiaco_max`	el máximo de pulsaciones registradas por minuto.

(a) Carga la base de datos *corazon.txt* en R y muestra las primeras filas por pantalla.

```
corazon=read.table(file="corazon.txt", header=TRUE)
head(corazon)
```

La primera fila es una cabecera, por lo que tenemos que indicárselo a R con la opción `header=TRUE`. La separación de las columnas es un espacio, por lo que no tenemos que indicar nada más.

(b) ¿Cuántos pacientes hay registrados en total? ¿Cuál es la edad máxima y la mínima de los pacientes?

```
nrow(corazon)
max(corazon$Edad); min(corazon$Edad)
```

Hay 297 pacientes en total. La edad máxima es 77 y la mínima 29 años.

(c) Obtén los datos del paciente número 73.

```
corazon[73, ]
```

El paciente número 73 es un hombre asintomático de 62 años, una presión arterial de 120 mmHg, 267 mg/dl de colesterol en sangre y 99 pulsaciones por minuto.

Antes de cargar la base de datos *corazon.txt* debes cambiar el directorio de trabajo de R a la carpeta que contenga el archivo en tu ordenador como se explica en la Sección 1.6.1. Si no lo haces R devolverá un error como el siguiente:

```
> corazon=read.table(file="corazon.txt", header=TRUE)
Error in file(file, "rt") :  cannot open the connection
In addition:  Warning message:
In file(file, "rt") :
   cannot open file 'Corazon.txt':  No such file or directory
```

1.6.3 La función `read.csv`

La función `read.csv` se usa para importar a R archivos de Excel con extensión .csv. Sus opciones son las mismas que `read.table`, lo único que cambian son los valores por defecto de `header` y `sep`. Ahora, por defecto, `header` vale `TRUE` y la separación de columnas es una coma.

Excel en español emplea la coma como separador decimal. Esto puede dar problemas a la hora de cargar archivos de Excel con R, ya que por defecto R asume que la coma es la separación entre columnas. Es decir, R lee `1,55` como dos números (el 1 y el 55) no como un número decimal. Para evitar problemas, lo mejor es **revisar siempre el archivo antes de cargarlo** para saber como están separados los decimales y cambiar las opciones `sep` y `dec` en consecuencia.

Caso práctico: En el archivo de excel *trasplantes.csv* contiene información de varios pacientes con cánceres hematológicos que han recibido un trasplante de médula ósea. A continuación, podemos ver una previsualización de las primeras diez filas del archivo:

trasplantes.csv

```
"Sexo";"Diagnosis";"Tipo_diagnosis";"Tiempo_hasta_trasplante"
1;"acute myeloid leukemia";1;5,16
1;"non-Hodgkin lymphoma";0;79,05
0;"non-Hodgkin lymphoma";0;35,58
0;"Hodgkin lymphoma";0;33,02
0;"acute lymphoblastic leukemia";0;11,4
1;"myelofibrosis";1;2,43
1;"acute myeloid leukemia";1;9,59
0;"multiple myelomas";0;43,43
1;"chronic lymphocytic leukemia";0;92,71
(...)
```

A cada paciente se le registraron varias variables de interés.

`Sexo`	el sexo del paciente: 0 indica que el paciente es un hombre, 1 indica que es mujer.
`Diagnosis`	el cáncer del paciente.
`Tipo_diagnosis`	variable que indica si es un cáncer mieloide (0) o linfoide (1).
`Tiempo_hasta_trasplante`	tiempo transcurrido entre el diagnóstico y el trasplante medido en meses.

(a) Carga la base de datos *trasplantes.csv* en R y muestra las primeras filas por pantalla.

```
trasplantes=read.csv(file="trasplantes.csv", sep=";", dec=",")
head(trasplantes)
```

La separación de las columnas es el punto y coma, y el símbolo decimal es una coma. Como no son los valores por defecto, tenemos que indicárselo a R con las opciones `sep=";"` y `dec=","`.

(b) ¿Cuántos pacientes hay registrados en total? Ordena el tiempo hasta el trasplante de mayor a menor.

```
nrow(trasplantes)
sort(trasplantes$Tiempo_hasta_trasplante, decreasing=TRUE)
```

Hay 57 pacientes en total.

(c) Obtén los datos de los dos últimos pacientes.

```
trasplantes[c(56, 57), ]
```

El penúltimo paciente es una mujer con síndrome mieloproliferativo que ha esperado 173.83 meses hasta su trasplante. El último paciente es un hombre con leucemia mieloide aguda que ha esperado 24.44 meses hasta el trasplante.

1.7 Ejercicio propuesto

En el archivo de excel *tuberculosis.csv* contiene los resultados de un estudio realizado en 1948 para comprobar si la estreptomicina era efectiva a la hora de combatir la tuberculosis. A conti-nuación puedes ver una previsualización de las primeras doce filas del archivo:

tuberculosis.csv

```
"sexo","grupo","mejora","temperatura"
"M","Control",TRUE,"0"
"F","Control",TRUE,"2"
"F","Control",TRUE,"0"
"M","Control",TRUE,"0"
"F","Control",TRUE,"1"
"M","Control",TRUE,"2"
"F","Control",TRUE,"1"
"M","Control",TRUE,"1"
"F","Control",TRUE,"1"
"M","Control",TRUE,"3"
"F","Control",TRUE,"2"
(...)
```

A cada paciente se le registraron varias variables de interés.

`sexo`	el sexo del paciente: `M` indica que el paciente es un hombre, `F` indica que es mujer.
`grupo`	indica si el paciente recibió el tratamiento (`Streptomycin`) o si por el contrario recibió un placebo (`Control`).
`mejora`	variable que indica si el paciente ha experimentado una mejora (`TRUE`) o no (`FALSE`) al final del estudio.
`temperatura`	temperatura corporal del paciente al inicio del estudio. Está registrada en cuatro categorías: sin fiebre (0), temperatura entre 37.2ºC y 37.75ºC (1), entre 37.75ºC y 38.3ºC (2), y más de 38.3ºC (3).

(a) Carga la base de datos *tuberculosis.csv* en y muestra las primeras filas por pantalla.

(b) ¿Cuántos pacientes hay registrados en total? ¿Cuántos han recibido el tratamiento de estreptomicina y cuántos el placebo?

(c) Obtén los datos de los pacientes numero 50 y 60.

2 Estadística descriptiva

La estadística descriptiva se ocupa de recoger, clasificar y resumir la información contenida en un conjunto de datos sobre una muestra o población, con el objetivo de describir las variables estadísticas registradas. Para llevar a cabo un análisis correcto de los datos, resulta de suma importancia la adecuada clasificación de las variables de interés, que pueden ser cualitativas o cuantitativas. En las Secciones 2.1 y 2.2, presentaremos las funciones de R básicas para realizar un análisis descriptivo de variables cualitativas y cuantitativas, respectivamente. Además, resolveremos distintos casos prácticos de interés biomédico para ilustrar el uso de dichas funciones.

2.1 Variables cualitativas

Las variables cualitativas expresan características, categorías o atributos. A su vez, estas variables se clasifican en nominales y ordinales: las variables cualitativas ordinales presentan una relación de orden entre sus posibles valores (por ejemplo, nivel bajo, medio y alto), mientras que en las variables cualitativas nominales no existe esta relación de orden (por ejemplo, verde, azul, marrón u otros).

Para realizar un análisis descriptivo básico de una variable muestral cualitativa, tanto nominal como ordinal, debemos obtener su tabla de distribución de frecuencias. Para llevarlo a cabo con R, el primer paso es calcular el tamaño de la muestra como

```
n=length(x)
```

A continuación, obtenemos las frecuencias absolutas con la función `table`

```
ni=table(x)
```

Las frecuencias relativas se pueden calcular dividiendo las frecuencias absolutas entre el tamaño muestral:

```
fi=ni/n
```

En el caso de las variables cualitativas nominales, la tabla de distribución de frecuencias se compone solo de las frecuencias absolutas y relativas, por lo que podemos obtener la tabla completa uniendo estas dos con la función `cbind`

```
tabla=cbind(ni,fi)
```

En el caso de las variables cualitativas ordinales, debemos calcular las frecuencias absolutas y relativas acumuladas, del siguiente modo:

```
Ni=cumsum(ni)
Fi=cumsum(fi)
```

y, por lo tanto, obtenemos la tabla de frecuencias con

```
tabla=cbind(ni, fi, Ni, Fi)
```

Una forma de obtener información visual de las observaciones es construir representaciones gráficas. En el caso de las variables cualitativas, resulta adecuado representar tanto las frecuencias absolutas como las frecuencias relativas con un diagrama de barras. Esto lo podemos hacer en R mediante la función

```
barplot(freq, main, col, names)
```

donde

`freq`	es el vector con las frecuencias a representar.
`main`	indica el título del gráfico.
`col`	denota el color (o vector con colores) de las barras.
`names`	es un vector con el nombre de las catergorías de la variable de interés.

Otra manera alternativa de representar los datos de una variable cualitativa es mediante un diagrama de sectores, que obtenemos en R mediante la función

```
pie(freq, main, col, labels)
```

donde

`freq`	denota el vector con las frecuencias (absolutas o relativas).
`main`	es el título del gráfico.
`col`	indica el color (o vector con colores) de los sectores.
`labels`	es un vector con el nombre de las categorías de la variable de interés.

Caso práctico: La base de datos *trasplantes.csv* contiene información sobre 58 pacientes que se sometieron a un trasplante de médula ósea.

(a) Lee la base de datos con R.

```
datos1a=read.csv("trasplantes.csv", sep=";", dec = ",")
```

Una vez cambiado el directorio de trabajo, podemos leer la base de datos con la función `read.csv`, teniendo en cuenta la separación entre columnas de este archivo y la coma decimal.

(b) Construye una tabla de frecuencias para la variable que contiene el tipo de diagnóstico (0: cáncer mieloide; 1: cáncer linfoide).

```
n=length(datos1a$Tipo_diagnosis)
ni=table(datos1a$Tipo_diagnosis)
fi=ni/n
tabla=cbind(ni, fi)
tabla
```

El tipo de diagnóstico es una variable cualitativa nominal, ya que no existe un orden entre las distintas categorías (cáncer mieloide y cáncer linfoide). Por lo tanto, la tabla de frecuencias se compone de las frecuencias absolutas y las frecuencias relativas. Una vez calculada la tabla, podemos verificar que el número de pacientes con cáncer mieloide es 28, mientras que 30 pacientes fueron diagnosticados con cáncer linfoide. Como se aprecia en las frecuencias relativas, esto supone un porcentaje de pacientes cercano al 50% para los dos tipos de diagnóstico.

(c) Representa un diagrama de barras y un diagrama de sectores para el tipo de diagnóstico.

```
barplot(ni, main="Tipo de diagnóstico", col=c("blue", "red"),
      names=c("Mieloide","Linfoide"))
pie(ni, main="Tipo de diagnóstico", col=c("blue", "red"),
      labels=c("Mieloide","Linfoide"))
```

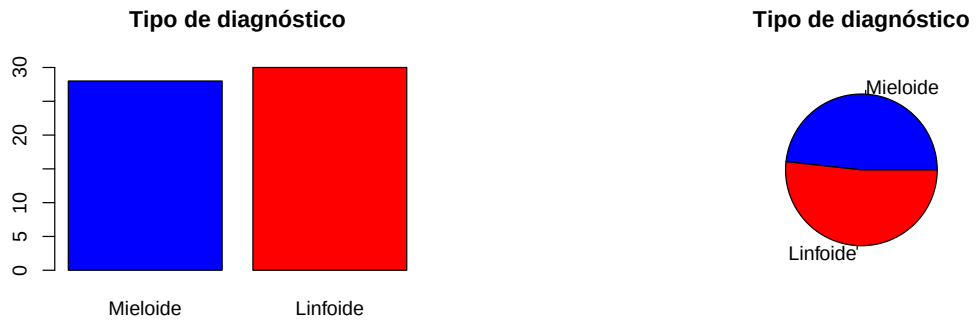


Figura 5: *Diagrama de barras (izquierda) y diagrama de sectores (derecha) para la variable tipo de diagnóstico.*

La Figura 5 muestra las dos representaciones gráficas obtenidas.

2.2 Variables cuantitativas

Las variables cuantitativas, al contrario que las cualitativas, toman valores numéricos. Son, frecuentemente, el resultado de una medición, como el contenido de alcohol en sangre (g/l). A su vez, este tipo de variables se clasifican en cuantitativas discretas y cuantitativas continuas.

2.2.1 Variables cuantitativas discretas

Las variables cuantitativas discretas pueden tomar un número discreto de valores (por ejemplo, cero, uno, dos, tres,...). Para poder realizar un análisis descriptivo básico para este tipo de variables, debemos obtener la tabla de distribución de frecuencias que, como en el caso de variables cualitativas ordinales, consta de las frecuencias absolutas, relativas, absolutas acumuladas y relativas acumuladas. Además, podemos obtener una representación gráfica adecuada de este tipo de variables mediante un diagrama de barras, tanto de las frecuencias absolutas como relativas, como los introducidos en la Sección 2.1.

Caso práctico: En la base de datos `birthwt` de la librería `MASS` se encuentran datos sobre 189 partos ocurridos en Springfield en 1986.

(a) Carga la base de datos en R.

```
library(MASS)
datos1b = birthwt
```

(b) Construye una tabla de frecuencias para la variable `ftv` que contiene el número de consultas médicas en el primer trimestre.

```
n=length(datos1b$ftv)
ni=table(datos1b$ftv)
fi=ni/n
Ni=cumsum(ni)
Fi=cumsum(fi)
tabla=cbind(ni, fi, Ni, Fi)
tabla
```

Dado que la variable número de consultas médicas es cuantitativa discreta, la tabla de frecuencias se compone de las frecuencias absolutas, relativas, absolutas acumuladas y relativas acumuladas. Una vez calculadas, observamos que el valor más repetido (moda) es el de cero consultas en el primer trimestre, con cien pacientes. Las frecuencias relativas nos aportan información acerca de la distribución del número de visitas: más de la mitad de las pacientes no tuvieron ninguna consulta médica en el primer trimestre, mientras que casi el 25% de las pacientes tuvieron únicamente una consulta. Además, observamos con las frecuencias acumuladas que 177 pacientes, lo que supone un más del 93% del total, tuvieron dos consultas o menos en el primer trimestre.

(c) Representa el número de consultas durante el primer trimestre mediante un diagrama de barras.

```
barplot(ni, main="Número de visitas (trimestre 1)", col="pink")
```

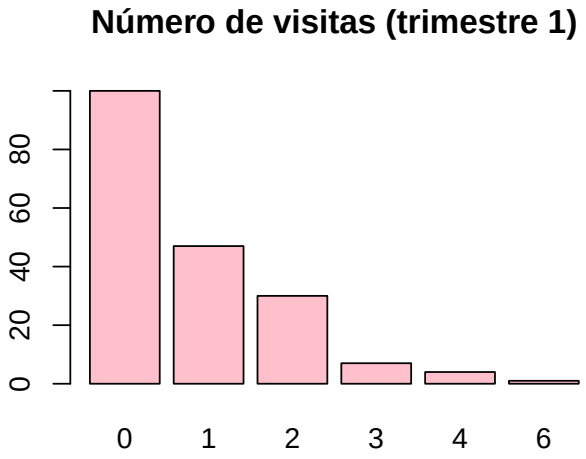


Figura 6: *Diagrama de barras para la variable número de consultas durante el primer trimestre.*

La Figura 6 muestra la representación gráfica obtenida. La barra con mayor altura es la asociada al valor cero visitas correspondiente a la moda.

2.2.2 Variables cuantitativas continuas

Las variables cuantitativas continuas toman valores dentro de un intervalo en la recta real. La descripción de este tipo de variables se puede realizar mediante el cálculo de medidas características, como son las medidas de posición, dispersión y forma, cuya obtención con R se describe a continuación.

En lo referente a las medidas de posición, destaca el uso de la media aritmética, que obtenemos en R con la función

```
mean(x)
```

Otra medida de posición central, la mediana, se puede calcular en R con la función

```
median(x)
```

Por otra parte, los cuantiles proporcionan medidas de posición no centrales. Podemos obtenerlos en R con la función

```
quantile(x, probs, type)
```

donde

`x`	denota el vector que contiene la muestra.
`probs`	es un número o vector que proporciona las probabilidades asociadas a los cuantiles.
`type`	se corresponde con un número entre 1 y 9 indicando el método para el cálculo del cuantil.

> La función `quantile(x)` dispone de nueve métodos para el cálculo del cuantil, que se pueden especificar con el argumento `type`. Por defecto, se emplea el método número 7. Podemos encontrar información sobre los nueve métodos en la documentación de la función, mediante `help(quantile)`.

Además de las medidas de posición, resulta de utilidad el cálculo de medidas de dispersión, que describen la variabilidad o esparcimiento de la muestra con respecto a la posición central. Una de estas medidas es la cuasivarianza, que podemos calcular en R con la función

```
var(x)
```

Relacionada con la cuasivarianza está la cuasi desviación típica, que se puede obtener como la raíz cuadrada de la varianza

```
sqrt(var(x))
```

o también directamente con la función

```
sd(x)
```

El rango de las observaciones, que se define como la diferencia entre el máximo y el mínimo de los datos, se puede obtener de dos formas distintas: o bien manualmente con

```
max(x)-min(x)
```

o también con la siguiente función:

```
diff(range(x))
```

Las funciones `var(x)` y `sd(x)` calculan, respectivamente, la cuasivarianza y la cuasidesviación típica de la muestra. Es decir, para el cálculo de estas medidas, se divide entre $n-1$ en lugar de entre n, donde n es el tamaño muestral. La varianza de la muestra definida como $\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2$ puede calcularse mediante

```
n=length(x)
(n-1)*var(x)/n
```

Para valores de n suficientemente altos, varianza y cuasivarianza tomarán valores similares.

De forma similar, el recorrido intercuartílico o rango intercuartílico es la diferencia entre el tercer cuartil y el primer cuartil, y lo podemos calcular en R mediante

```
quantile(x,0.75)-quantile(x,0.25)
```

Finalmente, podemos obtener el coeficiente de variación, medida que no depende de la escala de nuestra variable, y nos permite la comparación la dispersión entre distintas muestras. Dado que el coeficiente de variación se define como la desviación típica entre la media, podemos calcularlo como

```
sd(x)/mean(x)
```

El coeficiente de variación así definido solo tiene sentido para variables que tomen valores positivos. A veces, este coeficiente se define como el valor absoluto del cociente entre la desviación típica y la media, calculándose entonces como

```
abs(sd(x)/mean(x))
```

Además de las medidas de posición y dispersión, existen también medidas de forma, para poder estudiar la distribución de las observaciones. Es el caso del coeficiente de apuntamiento o curtosis y el coeficiente de asimetría de Pearson. En R, el cálculo de estos coeficientes se realiza a través de funciones que están disponibles en la librería `moments`, que es necesario instalar y cargar mediante

```
install.packages("moments")
library(moments)
```

Una vez cargado el paquete, calculamos el coeficiente de apuntamiento y asimetría, respectivamente, con las funciones

```
kurtosis(x)
skewness(x)
```

Por último, también resulta de utlidad la representación gráfica de variables continuas. El gráfico más utilizado para este tipo de variables es el histograma, que representamos en R mediante la función

```
hist(x, main, xlab)
```

donde

`x`	es el vector con los datos de la variable continua.
`main`	indica el título del gráfico.
`xlab`	es el nombre de la variable de interés.

Otra manera de representar los datos de una variable continua es mediante un diagrama de caja, también conocido por su nombre en inglés, *boxplot*. R permite la representación del diagrama de caja con la función

```
boxplot(x, main, xlab)
```

donde los argumentos denotan lo mismo que en el caso del histograma.

Caso práctico: La base de datos *corazon.txt* recopila información acerca de pacientes con enfermedades relacionadas con el corazón.

(a) Lee la base de datos con R.

```
datos1c=read.table("corazon.txt", header=TRUE)
```

Una vez cambiado el directorio de trabajo, podemos leer la base de datos con la función `read.table`, teniendo en cuenta la cabecera. No necesitamos el argumento `sep` dado que la separación del archivo es mediante espacios, que es el valor por defecto.

(b) Se tiene interés en conocer el nivel de colesterol de los pacientes, medido en mg/dl. Calcula las siguientes medidas de posición para el colesterol: media, mediana, cuartiles 1 y 3 y cuantil 0.6.

```
mean(datos1c$Colesterol)
median(datos1c$Colesterol)
quantile(datos1c$Colesterol, 0.25)
quantile(datos1c$Colesterol, 0.75)
quantile(datos1c$Colesterol, 0.6)
```

Obtenemos que la media de la muestra es igual a 247.35 mg/dl. La mediana, que es algo inferior, es 243 mg/dl, es decir, la mitad de la muestra tienen un nivel de colesterol inferior a este valor. Además, el 25% de la muestra tiene un nivel de colesterol menor que 211 mg/dl y el 25% de la muestra con un nivel de colesterol más alto tiene niveles de colesterol superiores a 276 mg/dl. Por último, el 60% de la muestra tiene un nivel de colesterol inferior a 254.6 mg/dl.

(c) Calcula las siguientes medidas de dispersión para el colesterol: varianza, desviación típica, coeficiente de variación, rango y rango intercuartílico.

```
var(datos1c$Colesterol)
sd(datos1c$Colesterol)
sd(datos1c$Colesterol)/mean(datos1c$Colesterol)
diff(range(datos1c$Colesterol))
quantile(datos1c$Colesterol,0.75)-quantile(datos1c$Colesterol,0.25)
```

Obtenemos una varianza de 2703.749 para el colesterol de los pacientes de la muestra. La desviación típica es de 51.998 mg/dl, y el coeficiente de variación es 0.21. El rango de las

observaciones es de 438 mg/dl, que es la diferencia entre el mayor y el menor valor de la muestra. Por último, el rango intercuartílico es de 65 mg/dl.

(d) Calcula los coeficientes de apuntamiento y asimetría de Pearson para el colesterol.

```
library(moments)
kurtosis(datos1c$Colesterol)
skewness(datos1c$Colesterol)
```

Obtenemos un coeficiente de apuntamiento (o curtosis) para el colesterol de 7.349, que es mayor que tres, lo que implicaría una distribución leptocúrtica, es decir, en comparación con la distribución normal la distribución del colesterol es apuntada. En cuanto al coeficiente de asimetría de Pearson, obtenemos un valor de 1.112, mayor que cero, que implica que la distribución del colesterol presenta asimetría positiva. Esta asimetría positiva se puede observar a través de los gráficos representados en la Figura 7 del apartado (e).

(e) Representa la variable colesterol mediante un histograma y un diagrama de caja.

```
hist(datos1c$Colesterol, main="Histograma", xlab="Colesterol")
boxplot(datos1c$Colesterol, main="Diagrama de caja",
xlab="Colesterol")
```

La Figura 7 muestra el histograma y el diagrama de caja de la variable colesterol. Los datos atípicos identificados con puntos en el diagrama de caja muestran la existencia de pacientes que presentan niveles de colesterol extremadamente elevados. Además, ambos gráficos muestran la asimetría positiva que calculamos con el coeficiente de asimetría de Pearson, aunque puede verse que esta asimetría se debe, principalmente, a los datos atípicos. Por último, en el histograma también observamos que la distribución de los datos es bastante apuntada (en comparación con la normal), en concordancia con el coeficiente de apuntamiento.

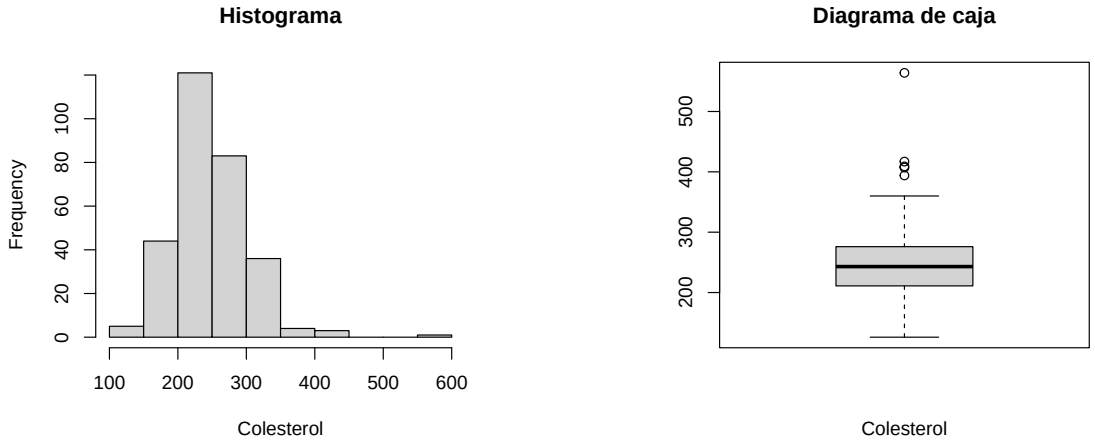


Figura 7: *Histograma (izquierda) y diagrama de caja (derecha) de la variable colesterol.*

2.3 Ejercicios propuestos

1. La base de datos `birthwt` de la librería `MASS` recoge adicionalmente información acerca de si la madre era fumadora o no (variable `smoke`: 0 no fumadora; 1 fumadora). Construye una

tabla de frecuencias para esta variable y representa un diagrama de barras y un diagrama de sectores.

2. Construye una tabla de frecuencias para la variable `Dolor` que recoge el tipo de dolor en la base de datos *corazon.txt* (1: angina común; 2: angina atípica; 3: dolor no anginal; 4: asintomático). Además, representa esta variable mediante diagramas de barras y de sectores.

3. Calcula todas las medidas de posición, dispersión y forma que conozcas para la variable `Tiempo_hasta_trasplante` contenida en la base de datos *trasplantes.csv* que recoge el tiempo hasta el trasplante. Representa esta variable mediante un histograma y un diagrama de caja.

3 Distribuciones de probabilidad

Las distribuciones de probabilidad describen el comportamiento de las variables aleatorias. En particular, permiten determinar la probabilidad de resultados concretos del experimento aleatorio modelizado por la variable.

En Teoría de Probabilidad existen muchos modelos teóricos empleados para modelizar un gran número de situaciones reales. Según el tipo de variable aleatoria, se distinguen distribuciones de probabilidad discretas y distribuciones de probabilidad continuas. En las Secciones 3.1 y 3.2, presentaremos las funciones de R básicas asociadas a los principales modelos de distribuciones discretas y continuas, respectivamente. Para ilustrar su uso, resolveremos varios casos prácticos interesantes en biomedicina.

3.1 Distribuciones de probabilidad discretas

Existen muchas distribuciones de probabilidad discretas pero, dado el carácter introductorio de este manual, nos limitaremos a estudiar en detalle la distribución binomial y la distribución de Poisson.

3.1.1 Distribución binomial

En R, los valores de la función masa de probabilidad de una variable con distribución binomial con parámetros n (número de veces que se repite el experimento aleatorio, $n = 0, 1, 2, \cdots$) y p (probabilidad de éxito, $0 < p < 1$), se obtienen a través de la función

```
dbinom(x, size, prob)
```

donde

`x`	es el valor de la variable para el cual queremos calcular la función masa de probabilidad. El argumento `x` podría ser también un vector de valores.
`size`	denota al parámetro n.
`prob`	se corresponde con la probabilidad de éxito p.

En R, el cálculo de valores de la función de distribución de una distribución binomial con parámetros n y p puede realizarse con la función

```
pbinom(q, size, prob)
```

donde `q` es el valor sobre el que queremos evaluar la función de distribución. El argumento `q` podría ser también un vector de valores. La definición de los argumentos `size` y `prob` es la misma que para la función `dbinom`.

El cálculo de cuantiles de una distribución binomial en R, requiere el uso de la función

```
qbinom(p, size, prob)
```

donde `p` es el orden del cuantil que queremos calcular (en tanto por uno). El argumento `p` podría ser también un vector de valores. Como antes, las definiciones de los argumentos `size` y `prob` no cambian.

Si el argumento `size` toma el valor 1, las tres funciones de R descritas permiten calcular la función de masa, la función de distribución y los cuantiles de la distribución Bernoulli.

Caso práctico: Estudios recientes demuestran que la probabilidad de falso negativo de las pruebas PCR (*polymerase chain reaction*) y de los test de antígenos a los ocho días de haber sido infectado por coronavirus son 0.2 y 0.6, respectivamente. Si se realizan las dos pruebas diagnósticas a 10 pacientes a los ocho días de la infección:

(a) ¿Cuál es la probabilidad de que no se produzca ningún falso negativo al realizar las 10 pruebas?

```
dbinom(x=0, size=10, prob=0.2)
dbinom(x=0, size=10, prob=0.6)
```

Para las pruebas PCR, probabilidad de que no se produzca ningún falso negativo es 0.107. Para los tests de antígenos, $1.048 \cdot 10^{-4}$.

(b) ¿Cuál es la probabilidad de que se produzcan 8 o 9 falsos negativos?

```
sum(dbinom(x=c(8, 9), size=10, prob=0.2))
sum(dbinom(x=c(8, 9), size=10, prob=0.6))
```

En el caso de las pruebas PCR, la probabilidad es $7.782 \cdot 10^{-5}$. Para los test de antígenos, 0.161.

(c) ¿Cuál es la probabilidad de que se produzcan menos de 4 falsos negativos al realizar las 10 pruebas?

```
pbinom(q=3, size=10, prob=0.2)
pbinom(q=3, size=10, prob=0.6)
```

La probabilidad es 0.879 para las PCR y 0.055 para los tests de antígenos.

(d) ¿Cuál es la probabilidad de que se produzcan al menos 2 falsos negativos?

```
1-pbinom(q=1, size=10, prob=0.2)
1-pbinom(q=1, size=10, prob=0.6)
```

La probabilidad de que se produzcan al menos 2 falsos negativos en las 10 pruebas PCR es 0.624. En los tests de antígenos, 0.998.

(e) ¿Cuál es el valor mediano de falsos negativos en cada una las pruebas? ¿Y el tercer cuartil?

```
qbinom(p=c(0.5,.75), size=10, prob=0.2)
qbinom(p=c(0.5,0.75), size=10, prob=0.6)
```

El valor mediano y el tercer cuartil son 2 y 3 para las pruebas PCR. Para los test de antígenos, 6 y 7, respectivamente.

3.1.2 Distribución de Poisson

Los valores de la función masa de probabilidad de una variable con distribución Poisson con media y varianza igual al parámetro $\lambda > 0$ se obtienen a través de la función

```
dpois(x, lambda)
```

donde

`x` es el valor de la variable para el cual queremos calcular la función masa de probabilidad. El argumento `x` podría ser también un vector de valores.

`lambda` es el parámetro λ.

En R, el cálculo de valores de la función de distribución de una distribución Poisson con parámetro λ puede realizarse con la función

```
ppois(q, lambda)
```

donde `q` es el valor sobre el que queremos evaluar la función de distribución. El argumento `q` podría ser también un vector de valores. La definición del argumento `lambda` es la misma que para la función `dpois`.

El cálculo de cuantiles de una distribución Poisson en R, requiere el uso de la función

```
qpois(p, lambda)
```

donde `p` es el orden del cuantil que queremos calcular (en tanto por uno). El argumento `p` podría ser también un vector de valores. Como antes, la definición del argumento `lambda` no cambia.

Caso práctico: Los leucocitos o glóbulos blancos son células sanguíneas móviles asociadas al sistema inmune. Tienen mayor tamaño que los hematíes y están presentes en la circulación en un número mucho menor. El ser humano adulto tiene, en media, 7 glóbulos blancos por dm^3 de sangre. Asumamos que el número de leucocitos sigue una distribución de Poisson.

(a) Determina la probabilidad de que un individuo tenga un leucocito por dm^3. Calcula también la probabilidad de que tenga uno o dos leucocitos por dm^3.

```
dpois(1, lambda=7)
sum(dpois(c(1, 2), lambda=7))
```

La probabilidad de un leucocito por dm^3 es 0.006. La probabilidad de uno o dos, 0.029.

(b) Si un individuo tiene un número de leucocitos inferior a 4 por dm^3 se dice que tiene una leucopenia. Calcula la probabilidad de leucopenia.

```
ppois(3, lambda=7)
```

La probabilidad de leucopenia es 0.082.

(c) Si un individuo tiene un número de leucocitos mayor a 10 por dm^3 indica que hay una leucocitosis. Calcula la probabilidad de leucocitosis.

```
1-ppois(q=10, lambda=7)
```

La probabilidad de leucocitosis es 0.098.

(d) Calcula el primer, segundo y tercer cuantil de la distribución del número de leucocitos por dm^3.

```
qpois(p=c(0.25, 0.5, 0.75), lambda=7)
```

Los cuartiles son 5, 7 y 9, respectivamente.

3.1.3 Otras distribuciones discretas

Las funciones de masa, distribución y cuantil de las restantes distribuciones discretas también están disponibles en R. Por ejemplo, para la distribución geométrica, están implementadas en R como `dgeom`, `pgeom` y `qgeom`, respectivamente. Para la distribución binomial negativa, están disponibles las funciones `dnbinom`, `pnbinom` y `qnbinom`. Para la hipergeométrica, `dhyper`, `phyper` y `qhyper`. Para la multinomial, `dmultinom`, `pmultinom` y `qmultinom`.

La función `help` de ayuda en R puede ser útil para obtener una descripción detallada de los argumentos de cada una de ellas. En particular, `help(dgeo)` describe en detalle las tres funciones y argumentos asociados a la distribución geométrica.

3.2 Distribuciones de probabilidad continuas

Al igual que en el caso discreto, existen múltiples distribuciones de probabilidad continuas pero, en este manual, nos centraremos en el estudio de la distribución normal y las distribuciones exponencial y Weibull por sus interesantes aplicaciones en análisis de supervivencia.

3.2.1 Distribución normal

En R, el valor de la función de densidad normal de media 0 y desviación típica 1 en un punto se obtiene a través de la función

```
dnorm(x, mean, sd)
```

donde

`x`	es el valor sobre el que deseamos evaluar la función de densidad. El argumento `x` podría ser también un vector de valores.
`mean`	representa la media de la variable. Por defecto, igual a 0.
`sd`	es la desviación típica de la variable. Por defecto, igual a 1.

Aunque, la distribución normal estándar es la considerada por defecto, la modificación de los valores de los argumentos `mean` y `sd` permitiría considerar cualquier valor para la media y para la desviación típica.

Las funciones `pnorm` y `qnorm` se comportan de forma similar a sus equivalentes para las variables discretas y devuelven valores de la función de distribución y cuantiles para una distribución normal, respectivamente. Sus argumentos son los siguientes:

```
pnorm(q, mean, sd)
qnorm(p, mean, sd)
```

donde

`q` es el valor sobre el que deseamos evaluar la función de distribución. El argumento `q` podría ser también un vector de valores.
`p` representa el orden del cuantil que queremos obtener.
`mean` es la media. Por defecto, vale 0.
`sd` representa la desviación típica. Por defecto, es 1.

Caso práctico: La tensión ocular en individuos sanos se distribuye como una normal de media 13mmHg y desviación típica 2.7mmHg. Sin embargo, la tensión ocular en pacientes con glaucoma se distribuye como una normal de media 24mmHg y desviación típica 5mmHg. Para estudiar la capacidad diagnóstica de la tonometría ocular en el diagnóstico del glaucoma, se establece como criterio diagnóstico una cifra de tensión ocular de 16mmHg.

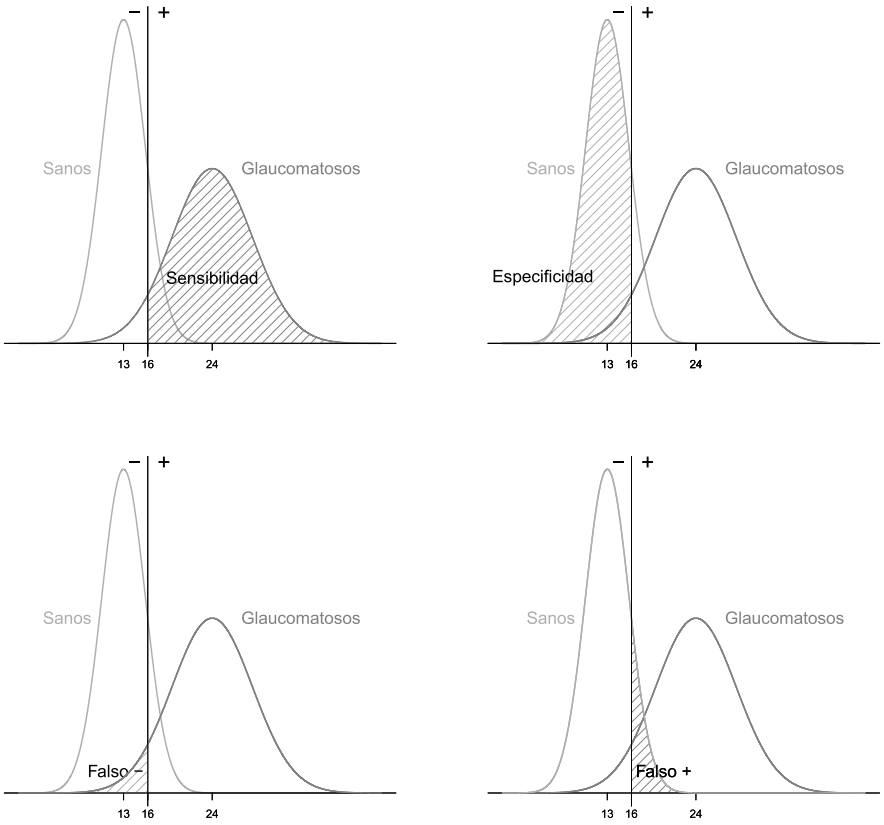


Figura 8: *Representación gráfica de la sensibilidad, de la especificidad, de la probabilidad de falsos positivos y de falsos negativos.*

(a) Representa gráficamente las funciones de densidad correspondientes a la variable tensión ocular en individuos sanos y enfermos.

```
x=seq(from=0, to=45, length=1000)
plot(x, dnorm(x, mean=13, sd=2.7), type="l")
points(x, dnorm(x, mean=24, sd=5), type="l", col="darkred")
```

La Figura 8 contiene la representación gráfica de las dos funciones de densidad.

> 👁 La función `seq(from, to, length)` genera un vector de elementos equiespaciados desde el valor indicado en `from` hasta `to` con longitud `length`. La función `points(x, y)` agrega la secuencia de puntos con abcisas en el vector `x` y con ordenadas en el vector `y` sobre un gráfico ya existente.

(b) Calcula la sensibilidad y especificidad de esta prueba diagnóstica representadas gráficamente en la Figura 8.

```
1-pnorm(q=16, mean=24, sd=5)
pnorm(q=16, mean=13, sd=2.7)
```

La sensibilidad es igual a 0.945 y la especificidad, 0.867.

(c) Determina las probabilidades de falsos negativos y de falsos positivos representadas gráficamente en la Figura 8.

```
pnorm(q=16,  mean=24, sd=5)
1-pnorm(q=16, mean=13, sd=2.7)
```

Son 0.055 y 0.133, respectivamente.

(d) Determina el tercer cuartil de la variable tensión ocular en individuos sanos y en glaucomatosos. ¿Qué observas?

```
qnorm(p=0.75, mean=13, sd=2.7)
qnorm(p=0.75, mean=24, sd=5)
```

En los individuos sanos, el tercer cuartil es igual a 14.821mmHg. En los glaucomatosos, 27.372mmHg. El 75% de los individuos sanos tienen valores de tensión ocular por debajo de 14.821mmHg; sin embargo, en el 75% de los pacientes con glaucoma ocular, están por debajo de 27.372mmHg. Los valores de tensión ocular son claramente mayores en los individuos glaucomatosos.

(e) Calcula el valor máximo de tensión ocular por debajo del cual se encuentran el 80% de los pacientes con glaucoma.

```
qnorm(p=0.8, mean=24, sd=5)
```

El 80% de los pacientes con glaucoma tienen valores de tensión ocular inferiores a 28.208 mmHg.

(f) ¿Qué porcentaje de individuos con glaucoma tienen tensión ocular por encima del tercer cuartil correspondiente a individuos sanos? ¿Y entre ese valor y 24mmHg?

```
1-pnorm(q=14.821, mean=24, sd=5)
pnorm(q=24, mean=24, sd=5)-pnorm(q=14.821, mean=24, sd=5)
```

El 96.681% de los individuos con glaucoma tienen valores de tensión ocular por encima del tercer cuartil correspondiente a individuos sanos. El 46.681% tienen valores de tensión ocular entre 14.821mmHg y 24mmHg.

3.2.2 Distribución exponencial

El valor de la función de densidad exponencial con media $a > 0$ en un punto se obtiene a través de la función

```
dexp(x, rate)
```

donde

`x` es el valor sobre el que deseamos evaluar la función de densidad. El argumento `x` podría ser también un vector de valores.

`rate` representa el valor $1/a$. Por defecto, su valor es 1.

Las funciones `pexp` y `qexp` se comportan de forma similar a sus equivalentes para la distribución normal y devuelven valores de la función de distribución y cuantiles para una distribución exponencial, respectivamente. Sus argumentos principales son los siguientes:

```
pexp(q, rate)
qexp(p, rate)
```

donde

`q` es el valor sobre el que deseamos evaluar la función de distribución. El argumento `q` podría ser también un vector de valores.

`p` representa el orden del cuantil que queremos obtener.

`rate` representa el valor $1/a$. Por defecto, su valor es 1.

Caso práctico: Se ha comprobado que el tiempo de vida (en años) de cierto tipo de marcapasos es una variable continua con distribución exponencial de media 16.

(a) ¿Cuál es la probabilidad de que a una persona a la que se le ha implantado este marcapasos se le deba reimplantar otro antes de 20 años?

```
pexp(20, rate=1/16)
```

La probabilidad de reimplantación antes de los 20 años es 0.713.

(b) ¿Cuál es la probabilidad de que el marcapasos implantado dure más de 16 años? ¿Y más de 20 años?[6]

```
1 - pexp(16, rate=1/16)
1 - pexp(20, rate=1/16)
```

La probabilidad de que dure más de 16 años es 0.368 y de que dure más de 20 años, 0.287.

(c) Representa gráficamente la función de supervivencia de la variable tiempo de vida (en años) del marcapasos.

```
x=seq(0, 50, length=1000)
plot(x, 1-pexp(x, rate=1/16), type="l", ylim=c(0,1))
```

La Figura 9 (izquierda) contiene la representación gráfica (negro) de la función de supervivencia.

[6] La función de supervivencia S se define como $S(x) = 1 - F(x)$, $x \geqslant 0$ donde F denota la función de distribución.

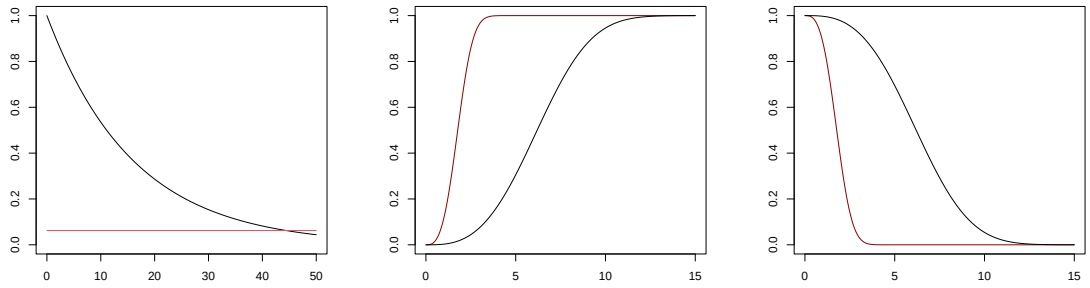


Figura 9: *Función de supervivencia (negro) y riesgo (rojo) para la variable tiempo de vida (en años) de un marcapasos con distribución exponencial de media 16 (izquierda). Funciones de distribución (centro) y supervivencia (derecha) para la variable tiempo (en años) hasta la recurrencia después de recibir un tratamiento de quimioterapia con diseminación (rojo) y sin diseminación (negro).*

(d) Representa gráficamente la función de riesgo de la variable tiempo de vida (en años) del marcapasos.[7] ¿Qué observas?

```
plot(x, dexp(x, rate=1/16)/(1-pexp(x, rate=1/16)), ylim=c(0,1),
type="l")
```

La Figura 9 (izquierda) contiene la representación gráfica (rojo) de la función de riesgo que es constante e igual a $1/16$.

3.2.3 Distribución Weibull

El caso práctico anterior, permite comprobar que la función de riesgo para la distribución exponencial con media $a > 0$ es constante e igual a $1/a$. Coloquialmente, se dice que la distribución exponencial sufre pérdida de memoria porque el riesgo de que ocurra el evento de interés en un intervalo de tiempo concreto no depende del paso del tiempo. Por eso, su uso en análisis de supervivencia es bastante escaso. Asumir que el riesgo es constante puede ser poco realista si interesa estudiar el tiempo hasta la muerte de un paciente desde el diagnóstico de una enfermedad, el tiempo de recurrencia desde la finalización de un tratamiento o el tiempo de eficacia de una intervención desde que fue realizada. Habitualmente, la distribución que se emplea para modelar datos de supervivencia es la Weibull.

En R, el valor de la función de densidad Weibull con parámetro de forma b y de escala c ($b, c > 0$) en un punto se obtiene a través de la función

```
dweibull(x, shape, scale)
```

donde

[7] La función de riesgo se escribe como $h(x) = f(x)/S(x)$, $x \geqslant 0$, donde f y S denotan las funciones de densidad y superviencia, respectivamente.

`x`	es el valor sobre el que deseamos evaluar la función de densidad. El argumento `x` podría ser también un vector de valores.
`shape`	representa el parámetro de forma b.
`c`	es el parámetro de escala c. Por defecto, su valor es 1.

Las funciones `pweibull` y `qweibull` se comportan de forma similar a sus equivalentes para la distribución normal y devuelven valores de la función de distribución y cuantiles para una distribución exponencial, respectivamente. Sus argumentos son los siguientes:

```
pweibull(q, shape, scale)
qweibull(p, shape, scale)
```

donde

`q`	es el valor sobre el que deseamos evaluar la función de distribución. El argumento `q` podría ser también un vector de valores.
`p`	representa el orden del cuantil que queremos obtener.
`shape`	es el parámetro de forma b.
`scale`	representa el parámetro de escala c. Como antes, por defecto, es 1.

Si el argumento `shape` toma el valor 1 entonces la distribución Weibull coincide con una distribución exponencial de media igual al valor seleccionado para `scale`. Por ejemplo, los comandos

```
dweibull(1.5, shape=1, scale=3)
dexp(1.5, rate=1/3)
```

permiten evaluar la densidad exponencial de media 3 en el punto 1.5. Ambas, proporcionan el mismo resultado igual a 0.20.

Caso práctico: En un grupo de pacientes oncológicos con diseminación, se sabe que la variable que mide el tiempo (en años) hasta la recurrencia después de recibir un tratamiento de quimioterapia sigue una distribución Weibull de parámetros $b = 3$ y $c = 2$. Sin embargo, si no se ha observado diseminación, esta variable sigue una distribución Weibull de parámetros $b = 3$ y $c = 7$.

(a) Representa gráficamente las funciones de distribución de esta variable en los dos grupos descritos.

```
x=seq(0, 15, length=1000)
plot(x, pweibull(x, shape=3, scale=2), type="l", col="darkred")
points(x, pweibull(x, shape=3, scale=7), type="l")
```

La Figura 9 (centro) contiene la representación gráfica de las funciones de distribución en rojo para el grupo donde se ha observado diseminación y en negro, para el otro grupo.

(b) Representa gráficamente las funciones de supervivencia de esta variable en los dos grupos de pacientes oncológicos. ¿Qué observas?

```
plot(x, 1-pweibull(x, shape=3, scale=2), type="l",  col="darkred")
points(x, 1-pweibull(x, shape=3, scale=7), type="l")
```

La Figura 9 (derecha) contiene la representación gráfica de las funciones de supervivencia en rojo para el grupo donde se ha observado diseminación y en negro, para el otro grupo.

(c) Para los dos grupos calcula la probabilidad de que el tiempo de recaída supere los 5 años de haber recibido el tratamiento. ¿Qué grupo tiene un prognóstico más favorable?

```
1-pweibull(5, shape=3, scale=2)
1-pweibull(5, shape=3, scale=7)
```

En el grupo en el que se ha observado diseminación, la probabilidad es $1.637 \cdot 10^{-7}$. En el otro grupo, 0.694. Por tanto, el grupo en el que no se ha observado diseminación tiene claramente un prognóstico más favorable.

Caso práctico: En 10 pacientes con esclerosis múltiple remitente-recurrente sin tratamiento se registró el tiempo (en años) entre el primer brote producido por la enfermedad y el segundo. Los valores registrados fueron:

$$5.6, 19.5, 12.9, 1.4, 2.4, 0.2, 1, 10.7, 0.2, 4.9.$$

Para un grupo de 12 pacientes con esclerosis múltiple remitente-recurrente que seguió un tratamiento específico desde el primer brote también se registró el tiempo (en años) entre ambos brotes. Los resultados obtenidos en este caso fueron:

$$7.8, 21.5, 13.1, 3, 2.9, 1, 2.7, 13.2, 3.2, 5.9, 13.3, 15.1.$$

Se sabe que, en cada grupo, la variable tiempo transcurrido desde el primer al segundo brote sigue una distribución Weibull con parámetros desconocidos.

(a) El paquete `MASS` de R permite estimar por máxima versimilitud los parámetros de muchas distribuciones univariantes[8] entre las que se encuentra la distribución Weibull. Calcula los estimadores de los parámetros de forma y escala para el grupo con y sin tratamiento:

```
install.packages("MASS")
library(MASS)
grupo1=c(5.06, 3.96, 2.46, 0.57, 5.92, 0.62, 0.37, 1.92, 2.11, 9.39)
grupo2=c(7.8, 21.5, 13.1, 3, 2.9, 1, 2.7, 13.2, 3.2, 5.9, 13.3, 15.1)
p1=fitdistr(grupo1, densfun="weibull")$estimate; p1
p2=fitdistr(grupo2, densfun="weibull")$estimate; p2
```

En el grupo sin tratamiento, el estimador del parámetro de forma es 1.154 y el estimador del parámetro de escala es 3.409. En el grupo con tratamiento, el estimador del parámetro de forma es 1.357 y el estimador del parámetro de escala es 9.353.

(b) A partir de los resultados obtenidos en el apartado (a), representa gráficamente las funciones de supervivencia estimadas para los dos grupos.

```
x=seq(0, 40, length=1000)
plot(x, 1-pweibull(x, shape=p1[1], scale=p1[2]), type="l", col=2)
points(x, 1-pweibull(x, shape=p2[1], scale=p2[2]), type="l")
```

La Figura 10 (izquierda) contiene la representación gráfica de las funciones de supervivencia estimadas para los dos grupos. En negro, se ha representado la curva del grupo con tratamiento y, en rojo, la curva del grupo sin tratamiento.

[8] Entre otras, puede usarse con la Poisson, la exponencial, la χ^2 de Pearson, la normal o la t de Student

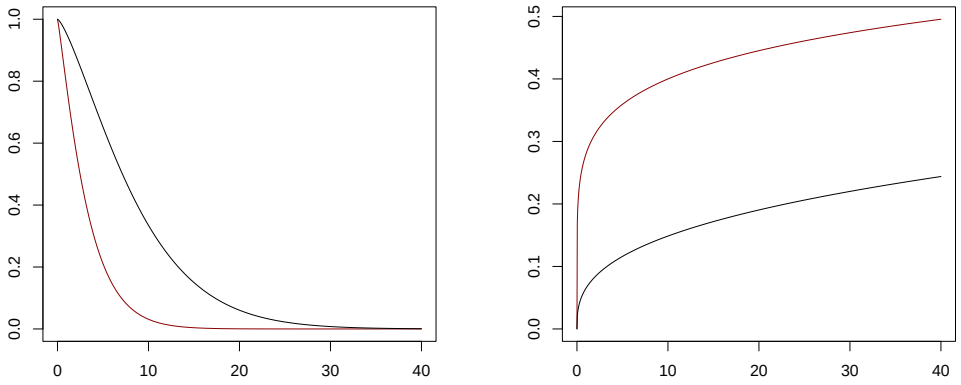


Figura 10: *Funciones de supervivencia (izquierda) y riesgo (derecha) para la variable tiempo (en años) entre el primer brote producido por la esclerosis múltiple remitente-recurrente y el segundo para los grupos sin tratamiento (rojo) y con tratamiento (negro).*

(c) A partir de los resultados obtenidos en el apartado (a), representa las funciones de riesgo estimadas para los dos grupos.

```
plot(x, dweibull(x, shape=p1[1], scale=p1[2])/(1-pweibull(x,
shape=p1[1], scale=p1[2])), type="l", col=2)
points(x, dweibull(x,shape=p2[1], scale=p2[2])/(1-pweibull(x,
shape=p2[1], scale=p2[2])), type="l")
```

La Figura 10 (derecha) contiene la representación gráfica de las funciones de riesgo estimadas para los dos grupos. En negro, se ha representado la curva del grupo con tratamiento y, en rojo, la curva del grupo sin tratamiento.

(d) De acuerdo con los resultados obtenidos, ¿crees que el tratamiento es eficaz?

Las funciones de supervivencia y riesgo representadas en la Figura 10 demuestran la eficacia del tratamiento. En particular, la Figura 10 (izquierda) permite deducir que la probabilidad de que el tiempo entre brotes sea superior a 10 años es 0.0312 en el grupo sin tratamiento y 0.335 en el grupo con tratamiento. De acuerdo con la Figura 10 (derecha), la función de riesgo en el grupo sin tratamiento toma valores claramente superiores que en el grupo con tratamiento.

3.2.4 Otras distribuciones continuas

Las funciones de densidad, distribución y cuantil de las restantes distribuciones continuas también están disponibles en R. Por ejemplo, para la distribución t de Student están implementadas en R como `dt`, `pt` y `qt`, respectivamente. Para la distribución χ^2 de Pearson, están disponibles las funciones `dchisq`, `pchisq` y `qchisq`. Para la distribución F, también conocida como distribución de Fisher-Snedecor, `df`, `pf` y `qf`. Como en el caso discreto, la función `help` de ayuda en R permite obtener una descripción detallada de los argumentos de cada una de ellas. En particular, `help(dt)` describe en detalle las tres funciones y argumentos asociados a la distribución t de Student.

3.3 Ejercicios propuestos

1. Una de las complicaciones más habituales de una operación de cadera es la infección que ocurre con una probabilidad de 0.01. Si, en un hospital, operan cada mes a 18 individuos:
   (a) Calcula la probabilidad de que ninguno de ellos sufra una infección.
   (b) Calcula la probabilidad de que exactamente un paciente sufra una infección.
   (c) Calcula la probabilidad de que más de 4 sufran una infección.
   (d) Calcula el valor mediano para el número de infecciones.
2. El número de enfermos recibidos cada hora en un centro sanitario sigue una distribución de Poisson. En media, se reciben 5 pacientes por hora.
   (a) Calcula la probabilidad de que entre las 10 y las 10:30 horas no se reciba ningún paciente.
   (b) Calcula la probabilidad de que entre las 10 y las 11 horas se reciban exactamente dos enfermos.
   (c) Calcula la probabilidad de que entre las 10 y las 11:15 horas se reciban más de 3 pacientes.
   (d) Calcula la probabilidad de que entre las 10 y las 11:15 horas se reciban entre 2 y 5 pacientes.
3. La concentración (en partes por millón) de plomo en sangre tiene distribución normal de media 0.25 y desviación típica igual a 0.11.
   (a) Una concentración superior o igual a 0.6 partes por millón se considera extremadamente alta. ¿Cuál es la probabilidad de que un individuo seleccionado al azar esté incluido en esa categoría?
   (b) Obtener la probabilidad de que la concentración (en partes por millón) de plomo en sangre de un individuo sea inferior a 0.16.
   (c) Determinar la probabilidad de que la concentración (en partes por millón) en plomo de un individuo esté comprendida entre 0.25 y 0.75.
   (d) Determinar la concentración mínima del 40% de los individuos con más concentración.
   (e) Determinar la mediana de esta distribución.
4. En un seguimiento realizado a 15 pacientes infectados por COVID-19, se ha medido el tiempo (en semanas) desde la aparición de síntomas hasta obtener un resultado negativo con test de antígenos. Los resultados obtenidos fueron:

   8.95, 4.69, 3.45, 7.49, 8.08, 9.49, 8.02, 5.71, 9.76, 10.35, 6.88, 4.72, 6.96, 6.94, 7.51

   (a) Asumiendo que los datos siguen una distribución Weibull, calcula los estimadores por máxima verosilitud de los parámetros de la distribución.
   (b) Evalúa la función de supervivencia estimada en 1, 3 y 10. Interpreta los resultados obtenidos.

4 Inferencia estadística

En numerosas ocasiones, queremos analizar una característica numérica (parámetro) de una población de interés. Por ejemplo, podemos estar interesados en la proporción de pacientes que pueden padecer un efecto adverso tras tomar un medicamento o cuál es el nivel medio de colesterol en personas "sanas". Como, en general, obtener esta información para toda la población es difícil y/o muy costoso, suele considerarse una muestra representativa de esa población y a partir de ella se hace inferencia sobre toda la población. Nótese que, a diferencia del Capítulo 2, en este tema no estamos solo interesados en saber qué ocurre con una muestra si no que pretendemos extraer conclusiones sobre toda la población.

A lo largo de este capítulo nos vamos a centrar en dos conceptos. El primero será la estimación puntual y los intervalos de confianza, esto es, como estimar (o aproximar) el valor de un parámetro y determinar un intervalo que lo contiene con cierta probabilidad. En segundo lugar, trataremos de comprobar (contrastar) si se puede concluir que el parámetro toma ciertos valores. Todo esto lo haremos para realizar inferencia sobre la proporción o la diferencia de proporciones en la Sección 4.1, mientras que las Secciones 4.2 y 4.3, se centrarán, respectivamente, en la media y en la diferencia de medias.

4.1 Inferencia sobre una y dos proporciones

Empezaremos este capítulo estudiando cómo hacer inferencia sobre la proporción o la diferencia de proporciones. Como, en general, es inviable preguntar a toda la población si cumple (o no) una condición para calcular la proporción poblacional, se tomará una muestra y se aporixamará (estimará) su valor con su estimador puntual, que es la proporción muestral.

Un problema que tiene la estimación puntual es que no sabemos cuánto se va a "aproximar" la proporción muestral (el estimador) a la verdadera proporción poblacional (el parámetro). En este contexto, puede ser de especial interés construir un intervalo de confianza que, haciendo uso de la muestra, nos porporcionará un intervalo que contiene proporción poblacional con cierta probabilidad (esta viene dada por el nivel de confianza). Finalmente, los contrastes de hipótesis nos permitirán dar respuesta a la hipótesis que el investigador puede plantearse sobre los valores que puede tomar la proporción.

Tanto los intervalos de confianza como los contrastes de hipótesis para la proporción y la diferencia de proporciones se van a realizar en R con la función

```
prop.test(x, n, p, alternative, conf.level)
```

donde

`x`	valor (o vector) con número de éxitos.
`n`	valor (o vector) con tamaño muestral.
`p`	valor que se quiere contrastar (p_0). Por defecto, $p_0 = 0.5$ para una proporción. Por defecto, $p_0 = 0$ para la diferencia de proporciones.
`alternative`	término indicando como es la hipótesis alternativa H_1, 'greater', 'less' o 'two-sided' según el contraste sea unilateral por la izquierda, por la derecha o bilateral. Por defecto, es bilateral ($H_1 : p \neq p_0$ o $H_1 : p_1 - p_2 \neq p_0$).
`conf.level`	nivel de confianza, entre 0 y 1 en el intervalo de confianza. Por defecto, se realiza el intervalo al nivel 0.95.

4.1.1 Intervalos de confianza para la proporción

Como hemos mencionado, siempre que queramos hacer intervalos de confianza para la proporción lo podemos hacer directamente con la función `prop.test`.

Caso práctico: Supongamos que queremos saber la proporción de fumadores en una población. Para aproximar su valor, tomamos, de forma completamente aleatoria (muestra aleatoria simple), a un total de $n = 100$ personas. De estas 100 personas, 30 nos han dicho que fuman.

(a) Obtén el estimador puntual para la proporción poblacional.

```
n=100
pgorro=30/n
pgorro
```

El estimador puntual es la proporción poblacional, que en este caso es igual a $\hat{p} = 0.3$

(b) Construye el intervalo de confianza para la proporción de fumadores al 90%.

```
prop.test(30, 100, conf.level=0.9)
```

El primer argumento de esta función es el número de elementos de la muestra que cumplen la condición que estamos estudiando. En este caso, el número de fumadores (30). Como segundo argumento, el tamaño muestral ($n = 100$). El tercer argumento (`conf.level`) es el nivel de confianza $(1-\alpha) = 0.9$. En la salida de esta función, podemos encontrar el intervalo de confianza debajo del encabezado `90 percent confidence interval`. En nuestro caso, vemos que el resultado es que el intervalo de confianza para la proporción de fumadores al 90% es $(0.226, 0.385)$. Entonces, la proporción poblacional $p \in (0.226, 0.385)$ con probabilidad 0.9.

> El intervalo de confianza que proporciona la función `prop.test` no coindice con el intervalo para la proporción obtenido como:
>
> $$\left(\hat{p} - z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}\right),$$

Caso práctico (continuación):

(b) Construye el intervalo de confianza para la proporción de fumadores al 90%.

Vamos a ver con el siguiente código que cuando repetimos este apartado con la fórmula anterior se obtiene otro resultado.

```
nivel=0.9
alfa=1-nivel
cuantil=qnorm(1-alfa/2, mean=0, sd=1)
ET=sqrt(pgorro*(1-pgorro)/n)
LI=pgorro-cuantil*ET
LS=pgorro+cuantil*ET
c(LI, LS)
```

El resultado de aplicar el código anterior es que el intervalo al 90% para la proporción es igual a $(0.225, 0.375)$.

Esta pequeña diferencia es debida a que, a la hora de proporcionar la anterior fórmula del intervalo para la proporción, se han realizado ciertas aproximaciones, como usar la distribución normal o intercambiar p por $\hat{p}$ en el Error Típico, que son correctas cuando n es "muy grande". Cuando n es "pequeño" existen otras formas más apropiadas de calcular el intervalo y que nos permiten obtener intervalos de confianza más precisos, pero que son tediosas de calcular a mano. Por defecto, internamente, la función `prop.test` realiza uno de estos métodos más precisos. Es por ello que en todos los casos donde queramos hacer inferencia para la proporción o para la diferencia de proporciones, podríamos ver que los resultados que nos devuelve R son ligeramente distintos a los que obtendríamos si calculamos el intervalo manualmente. Como los resultados de `prop.test` son más precisos que los que calculamos a mano, se recomienda emplear esta función.

4.1.2 Contrastes para la proporción

Una vez hemos visto cómo estimar la proporción poblacional desconocida (a través de una estimación puntual) y cómo construir un intervalo (de confianza) que contiene a la verdadera proporción con cierta probabilidad, vamos a ver cómo podemos responder a preguntas acerca de la proporción a través del contraste de hipótesis.

Para ello, supongamos que se suele asumir que la proporción p toma un determinado valor p_0 y que queremos plantearnos si, en nuestra población, esta suposición no es cierta. Esto es, queremos demostrar que la proporción p es distinta de p_0, por lo que esta será nuestra hipótesis alternativa (H_1). En contraposición está la hipótesis nula (H_0), lo que asumimos por defecto (la presunción de inocencia), que es que p toma el valor de p_0. Por tanto, nuestro contraste de hipótesis se escribe como:

$$\begin{cases} H_0: \ p = p_0 \\ H_1: \ p \neq p_0. \end{cases}$$

Una vez que sabemos el contraste que queremos realizar, deberemos imponer un nivel de significación α, que se corresponde con la cota del error de tipo I del contraste. Es decir, la probabilidad de rechazar H_0 cuando es cierta nunca superará el valor α.

Por fines ilustrativos, vamos a replicar con R como sería el proceso de un contraste, calculando el estadístico observado y viendo si está en la región de aceptación o de rechazo. Más adelante veremos como se puede simplificar este proceso con la función `prop.test`.

El estadístico y su distribución, bajo hipótesis nula cierta ($H_0 : p = p_0$), serían los siguientes:

$$\frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \sim N(0,1).$$

Una vez calculado el estadístico observado, rechazaremos la hipótesis nula cuando este estadístico sea "muy" distinto de 0. Para respetar el nivel de significación α prefijado, rechazamos la hipótesis nula $H_0 : p = p_0$ frente a $H_1 : p \neq p_0$ si

$$\frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \leqslant -z_{\alpha/2} \quad \text{ó} \quad \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \geqslant z_{\alpha/2}.$$

Caso práctico (continuación): Supongamos que queremos saber la proporción de fumadores en una población. Para aproximar su valor, tomamos, de forma completamente aleatoria (muestra aleatoria simple), a un total de $n = 100$ personas. De estas 100 personas, 30 nos han dicho que fuman.

(c) ¿Se puede afirmar que la proporción de fumadores es distinta del 20%?.

Como queremos demostrar que la proporción de fumadores (p) es distinta de 0.2, vamos a plantear el siguiente contraste de hipótesis:

$$\begin{cases} H_0: \ p = 0.2 \\ H_1: \ p \neq 0.2. \end{cases}$$

Para nuestro problema, vamos a imponer el nivel de significación más estándar, $\alpha = 0.05$. Pero recordemos, que en ocasiones nos interesará fijar niveles de significación más bajos para tener más "confianza" de que se puede rechazar la hipótesis nula (de que podemos concluir que $p \neq 0.2$ en nuestro caso).

Usando ahora la muestra, vamos a calcular el valor del estadístico (observado) con la ayuda de R.

```
p0=0.2
n=100
pgorro=30/n
ET=sqrt(p0*(1-p0)/n)
estadistico=(pgorro-p0)/ET
estadistico
```

El estadístico observado toma el valor 2.5. Si $\alpha = 0.05$, la región de aceptación será $(-z_{0.025}, z_{0.025})$ que, como podemos ver usando el código de abajo, es igual a $(-1.96, 1.96)$. La región de rechazo será el resto de la recta real, esto es, la unión de los intervalos $(-\infty, -1.96]$ y $[1.96, \infty)$.

```
alfa=0.05
cuantil=qnorm(1-alfa/2, mean=0, sd=1)
# Región de aceptación
c(-cuantil, cuantil)
# Regiones de rechazo
c(-Inf, -cuantil)
c(cuantil, Inf)
```

En nuestro caso, como el estadístico observado cae en la región de rechazo, $2.5 \in [1.96, \infty)$, nuestra conclusión es que podemos rechazar la hipótesis nula para un nivel de significación $\alpha = 0.05$. Por tanto, podemos concluir que, en esta población, la proporción de fumadores es distinta de 0.2.

Como mencionamos antes, con la función `prop.test` podemos evitar escribir todas esas líneas de código. En la salida de esta función podemos observar lo que se suele llamar p-valor (`p-value`), que es la probabilidad de obtener resultados del contraste al menos tan extremos como el resultado realmente observado, suponiendo que la hipótesis nula es correcta. Esto se traduce en que, independientemente del contraste que se esté realizando, lo único que debemos hacer es comparar

el p-valor con el nivel de significación α. Si el p-valor es menor o igual que α siempre rechazaremos la hipótesis nula. Por el contrario si el p-valor es mayor que α no tendremos evidencias significativas para rechazar la hipótesis nula.

Caso práctico (continuación):

(c) ¿Se puede afirmar que la proporción de fumadores es distinta del 20%?.

Planteando el mismo contraste y nivel de significación, podemos resolver esta pregunta directamente indicándole a `prop.test` que, bajo la hipótesis nula, $p_0 = 0.2$, a través del argumento `p`.

```
prop.test(30, 100, p=0.2)
```

En la salida de esta función, podemos observar que `p-value = 0.018`. Como el p-valor$= 0.018 < 0.05 = \alpha$, vemos que llegamos a la misma conclusión que cuando lo resolvimos paso a paso, y es que podemos rechazar la hipótesis nula de que $p = 0.2$ para un nivel de significación del 5%.

> Si cambiamos el nivel de significación, no necesitamos cambiar nada en nuestro código de R, ya que si, por ejemplo $\alpha = 0.01$, lo único que deberemos hacer es comparar ese valor de α con el p-valor$= 0.018$. Ahora, como $0.018 > 0.01$, no tenemos evidencias significativas para rechazar la hipótesis nula de que $p = 0.2$ para un nivel de significación $\alpha = 0.01$. Recordemos que las conclusiones de un contraste pueden depender del nivel de significación que hayamos impuesto.

Tanto la función `prop.test` como las que veremos más adelante permiten realizar contrastes bilaterales ($H_a : p \neq p_0$) y unilaterales ($H_a : p > p_0$ o $H_a : p < p_0$). Pero, por defecto, siempre van a realizar contrastes bilaterales. En el caso de que queramos hacer un contraste unilateral deberemos introducir el argumento `alternative`. Así, si queremos determinar si la proporción p es significativamente mayor que un valor dado p_0, podemos plantear el contraste unilateral por la derecha:

$$\begin{cases} H_0 : \ p \leqslant p_0 \\ H_1 : \ p > p_0. \end{cases}$$

Esto se podrá realizar indicando que en la hipótesis alternativa H_1 (`alternative`) hay un signo de mayor que (`greater`). Mientras que para el contraste unilateral por la izquierda,

$$\begin{cases} H_0 : \ p \geqslant p_0 \\ H_1 : \ p < p_0, \end{cases}$$

tendríamos que especificar en el argumento `alternative` un menor que (`less`).

Caso práctico (continuación):

(d) ¿Se puede afirmar que la proporción de fumadores está por debajo del 40% a un nivel de significación $\alpha = 0.05$?.

En este caso, el contraste puede plantearse como:

$$\begin{cases} H_0 : \ p \geqslant 0.4 \\ H_1 : \ p < 0.4. \end{cases}$$

Podemos obtener el p-valor del siguiente modo:

```
prop.test(30, 100, p=0.4, alternative="less")
```

Como el p-valor es 0.026, a un nivel de significación del 5% rechazaremos la hipótesis nula, concluyendo que la proporción de fumadores está por debajo del 40%.

4.1.3 Inferencia para dos proporciones

Supongamos que podemos dividir a los individuos en dos poblaciones. En ese caso, podemos estar interesados en saber si la proporción en la primera población(p_1) difiere de la de la segunda (p_2). Para analizarlo, podemos tomar dos muestras independientes y obtener tanto los intervalos de confianza como los contrastes de hipótesis con la función `prop.test`.

Caso práctico: Supongamos que para la anterior muestra de 100 personas, teníamos a 70 individuos de zonas urbanas, de los cuales 23 eran fumadores. El resto de los individuos son de zonas rurales, es decir, hay $30-23=7$ fumadores en el total de las $100-70=30$ personas que son de zonas rurales.

(a) Obtén el intervalo de confianza al 95% para la diferencia de proporciones de fumadores entre zonas urbanas y rurales.

A la hora de usar la función `prop.test`, deberemos pasarle (en un vector) cuantos sí que son fumadores en cada una de las dos poblaciones, `c(23,7)`, y cual es el tamaño muestral de cada una de las dos muestras, `c(70,30)`.

```
prop.test(c(23, 7), c(70, 30))
```

Por defecto, todas las funciones de R que vamos a ver calculan los intervalo de confianza al nivel de confianza $(1-\alpha)=0.95$. Así, en la salida de la función anterior, podemos ver que el intervalo de confianza al 95% para la diferencia de proporciones de fumadores es $(-0.116, 0.306)$.

(b) ¿Se puede afirmar que la proporción de fumadores difiere entre zonas urbanas y rurales a un nivel de significación $\alpha=0.05$?.

Por defecto, la función anterior realiza el contraste:

$$\begin{cases} H_0: \ p_1-p_2=0 \\ H_1: \ p_1-p_2\neq 0, \end{cases} \quad \text{o equivalentemente} \quad \begin{cases} H_0: \ p_1=p_2 \\ H_1: \ p_1\neq p_2. \end{cases}$$

Como podemos ver, se ha obtenido un p-valor igual a 0.475. Por tanto, para un nivel de significación $\alpha=0.05$, concluimos que no hay evidencias significativas para rechazar que ambas proporciones de fumadores (en urbano y rural) sean iguales.

4.2 Inferencia para la media

Supongamos que queremos estudiar la media de una variable en una población, por ejemplo, en los individuos "sanos" de un determinado país queremos saber el nivel medio de colesterol. En ese caso, si quisiésemos obtener la información precisa, deberíamos medir el nivel de colesterol de todos los individuos sanos del país y luego obtener la media. Esto, está claro que es completamente inviable, por lo que debemos tomar una muestra (aleatoria simple) de individuos de esa población,

esto es, vamos a escoger a un grupo "pequeño" de personas de esa población completamente al azar y hacer inferencia para saber cuánto vale la media.

Para aproximar esta media poblacional, un buen estimador puntual es la media muestral, que ya hemos visto en el Capítulo 2 cómo calcularla en R. Por otro lado, tenemos los intervalos de confianza y el contraste de hipótesis, cuyos resultados dependen de si conocemos o no la varianza poblacional. Vamos a ver, a continuación, los dos comandos que necesitamos para hacer inferencia sobre la media.

Cuando la varianza poblacional σ^2 es conocida, en R, podemos hacer uso de la función `z.test` de la librería `BSDA` para calcular tanto los intervalos de confianza como los contrastes. Debemos recordar, que si no tenemos esta librería instalada, debemos instalar este paquete antes de usar esta función.

```
install.packages("BSDA")
library(BSDA)
```

La inferencia para la media con varianza conocida se puede obtener con la siguiente función:

```
z.test(x, alternative, mu, sigma.x, conf.level)
```

donde

`x`	vector que contiene la muestra.
`alternative`	'greater', 'less' o 'two-sided' según sea la hipótesis alternativa. Por defecto, se realiza un contraste bilateral ('two-sided', $H_1 : \mu \neq \mu_0$).
`mu`	valor que se quiere contrastar (μ_0). Por defecto, $\mu_0 = 0$.
`sigma.x`	valor de la desviación típica σ conocida.
`conf.level`	nivel de confianza en el intervalo de confianza. Por defecto, se realiza el intervalo al nivel 0.95.

Por otro lado, cuando la varianza de la población es desconocida, para hacer inferencia no tenemos ni que cargar ninguna librería, ni que especificar el valor de σ (vamos a ver que no aparece el argumento `sigma.x`) y podemos emplear directamente esta función:

```
t.test(x, alternative, mu, conf.level)
```

donde

`x`	vector que contiene la muestra.
`alternative`	'greater', 'less' o 'two-sided' según sea la hipótesis alternativa. Por defecto, se realiza un contraste bilateral ('two-sided', $H_1 : \mu \neq \mu_0$).
`mu`	valor que se quiere contrastar (μ_0). Por defecto, $\mu_0 = 0$.
`conf.level`	nivel de confianza en el intervalo de confianza. Por defecto, se realiza el intervalo al nivel 0.95.

Caso práctico: Los datos tomados de una muestra que contiene el nivel de colesterol de pacientes de una población "sana" están en el fichero *colesterol5a.csv*.

(a) Carga estos datos en R y visualiza los niveles de colesterol de los primeros seis individuos.

Una vez cambiado el directoria de trabajo, seleccionando la carpeta donde están los datos (véase Sección 1.6), vamos a abrir los datos del fichero *colesterol5a.csv*.

```
datos5a=read.csv("colesterol5a.csv")
colesterol=datos5a$col
head(colesterol)
```

(b) Calcula un estimador puntual de la media poblacional.

```
estimador=mean(colesterol)
estimador
```

La media muestral, que en este caso es igual a 199.628 es un buen estimador de la media poblacional.

A la hora de saber cuánto vale la media poblacional, si queremos tener en cuenta la incertidumbre que se genera de tener solo una muestra, podemos calcular el intervalo que nos va a decir con cierta probabilidad dónde está la media poblacional. Como para la proporción podría calcularse paso a paso. Sin embargo, la función `z.test` de R nos permite obtenerlo directamente. En ambos casos, llegaríamos al mismo resultado. Además, la función `z.test` también nos permite resolver contrastes de hipótesis sobre la media.

Caso práctico (continuación): Los datos tomados de una muestra que contiene el nivel de colesterol de pacientes de una población "sana" están en el fichero *colesterol5a.csv*.

(c) Suponiendo que el colesterol se distribuye como una normal de varianza $\sigma^2 = 100 = 10^2$, construye los intervalos de confianza al 90% y al 99%

```
library(BSDA)
z.test(colesterol, sigma.x=10, conf.level=0.9)
z.test(colesterol, sigma.x=10, conf.level=0.99)
```

En los comandos anteriores, hemos indicado el valor de la desviación típica conocida $\sigma = 10$ en el argumento `sigma.x`. En su salida, podemos ver que el intervalo de confianza para el nivel medio de colesterol μ al 90% es $(197.983, 201.272)$, mientras que al 99% es $(197.052, 202.204)$. Esto es, podemos ver que a medida que aumenta el nivel de confianza, el intervalo será más grande.

(d) Se suele asumir que el nivel medio de colesterol es igual a 200, pero nosotros dudamos de esa veracidad en nuestra población. Si sabemos que $\sigma = 10$, comprueba, a un nivel de significación $\alpha = 0.05$, si se puede afirmar que el nivel medio de colesterol es distinto de 200.

En este caso, debemos plantear el contraste de hipótesis:

$$\begin{cases} H_0 : \mu = 200 \\ H_1 : \mu \neq 200. \end{cases}$$

Como bajo la hipótesis nula, $\mu_0 = 200$, vamos a pasarle esta información a la función `z.test` a través del argumento `mu`:

```
z.test(colesterol, mu=200, sigma.x=10)
```

En la salida de esta función podemos observar que el p-valor$= 0.71 > 0.05 = \alpha$. Por tanto, no tenemos evidencias significativas para rechazar la hipótesis nula de que $\mu = 200$ a un nivel de significación $\alpha = 0.05$.

Como hemos comentado, hasta ahora, hemos asumido un escenario ideal, poco realista, en el que hay un conocimiento de cuánto vale la varianza de la población, pero en general este dato es desconocido. En ese caso, como no sabemos que ocurre a nivel poblacional, tendremos que estimar esta varianza a partir de la muestra, con la varianza muestral. Como consecuencia, la distribución de estadístico de contraste no es gaussiana y pasa a ser una t de Student. En el caso de R, tanto los intervalos de confianza como los contrastes se pueden realizar con la función `t.test`.

Caso práctico (continuación): Los datos tomados de una muestra que contiene el nivel de colesterol de pacientes de una población "sana" están en el fichero *colesterol5a.csv*.

(e) Obtén el intervalo de confianza para la media de colesterol al nivel 95%, cuando la varianza es desconocida.

```
t.test(colesterol)
```

El intervalo de confianza para μ al 95% es $(197.775, 201.482)$.

(f) Supongamos que el equipo médico sospecha que en esa población el nivel medio de colesterol está por debajo de 200. Si no se conoce la varianza poblacional, ¿qué conclusiones debería sacar este equipo en base a la muestra?

En este caso, se quiere plantear el contraste:

$$\begin{cases} H_0 : \mu \geqslant 200 \\ H_1 : \mu < 200. \end{cases}$$

Si se fija el nivel de significación $\alpha = 0.05$, obtendría los resultados que se muestran a continuación:

```
t.test(colesterol, mu=200, alternative="less")
```

En la salida de esta función vemos que el p-valor (`p-value`) es igual que 0.346, como es mayor que $\alpha = 0.05$, este equipo no tendrá evidencias significativas para rechazar la hipótesis nula, por lo que no pueden demostrar su sospecha de que el nivel medio de colesterol esté por debajo de 200.

4.3 Inferencia para la diferencia de medias

Supongamos ahora que tenemos dos poblaciones normales y queremos comparar sus medias, μ_1 y μ_2. En ese caso, podemos tomar dos muestras independientes de cada una de las dos poblaciones. Como las medias poblacionales son desconocidas, podemos estar interesados en hacer inferencia sobre su diferencia $\mu_1 - \mu_2$ para saber si hay variación entre una población y la otra. Si solo queremos tener una intuición de en torno a qué valor debe estar esta diferencia, podemos construir un estimador puntual para la diferencia de medias, a través del estimador $\bar{X}_1 - \bar{X}_2$. Si queremos construir intervalos de confianza o hacer un contraste, distinguiremos los casos en los que las varianzas poblacionales son conocidas de los que no.

En función de si conocemos o no las varianzas poblaciones, se nos abren distintos escenarios. Si las varianzas poblacionales son conocidas, podemos usar de nuevo la función `z.test` con los siguientes argumentos:

```
z.test(x, y, alternative, mu, sigma.x, sigma.y, conf.level)
```

donde

`x`	vector que contiene la muestra de la población 1.
`y`	vector que contiene la muestra de la población 2.
`alternative`	'greater', 'less' o 'two-sided' según sea la hipótesis alternativa. Por defecto, se realiza un contraste bilateral ('two-sided', $H_1 : \mu_1 - \mu_2 \neq \mu_0$).
`mu`	valor que se quiere contrastar en la diferencia de medias (μ_0). Por defecto, $\mu_0 = 0$.
`sigma.x`	valor de la desviación típica de la población 1, σ_1, conocida.
`sigma.y`	valor de la desviación típica de la población 2, σ_2, conocida.
`conf.level`	nivel de confianza en el intervalo de confianza. Por defecto, se realiza el intervalo al nivel 0.95.

Cuando las varianzas son desconocidas, tenemos que los resultados dependen de si sabemos que se puede asumir iguales o si, por el contrario, no disponemos de esta información y por tanto hemos de asumir que son distintas.

En todos los escenarios analizados hasta ahora, hemos asumido que las muestras de las dos poblaciones son independientes, pero la función `t.test` nos permite también hacer inferencia con muestras apareadas. A continuación, mostramos los argumentos con los que podemos tratar el problema de diferencia de medias con varianzas poblacionales desconocidas:

```
t.test(x, y, alternative, mu, paired, var.equal, conf.level)
```

donde

`x`	vector que contiene la muestra de la población 1.
`y`	vector que contiene la muestra de la población 2.
`alternative`	'greater', 'less' o 'two-sided' según sea la hipótesis alternativa. Por defecto, se realiza un contraste bilateral ('two-sided', $H_1 : \mu_1 - \mu_2 \neq \mu_0$).
`mu`	valor que se quiere contrastar en la diferencia de medias (μ_0). Por defecto, $\mu_0 = 0$.
`paired`	muestras apareadas (`TRUE`) o independientes (`FALSE`). Por defecto, se asumen independientes.
`var.equal`	varianzas iguales (`TRUE`) o distintas (`FALSE`). Por defecto, se asumen distintas.
`conf.level`	nivel de confianza en el intervalo de confianza. Por defecto, se realiza el intervalo al nivel 0.95.

Caso práctico: Se quiere medir la diferencia en el nivel medio de colesterol entre pacientes jóvenes, que acuden a pediatría (μ_1), y adultos (μ_2). Para ello, se ha tomado una muestra que se puede encontrar en el fichero *colesterol5b.txt*. Los datos que se recogieron son los correspondientes a la variable `colesterolsinmed`. De estos, los 30 primeros corresponden a adultos y los 20 últimos a pediatría.

(a) Carga los datos y obtén un estimador puntual para la diferencia de medias entre jóvenes y adultos.

```
datos5b=read.table("colesterol5b.txt",header=TRUE)
colesteroladu=datos5b$colesterolsinmed[1:30]
colesterolped=datos5b$colesterolsinmed[31:50]
mean(colesterolped)-mean(colesteroladu)
```

El estimador puntual para la diferencia de medias es igual a -21.502.

(b) Obtén un intervalo de confianza al 99% para la diferencia de medias, suponiendo que la desviación típica de los pacientes de pediatría es igual a 30 y que la de los adultos es igual a 10.

En este caso, debemos pasarle a z.test la información de que la desviación típica de la primera población es `sigma.x=30` y la de la segunda población es `sigma.y=10`.

```
z.test(colesterolped, colesteroladu, sigma.x=30, sigma.y=10,
       conf.level=0.99)
```

El intervalo de confianza para la diferencia de medias al nivel $(1-\alpha) = 0.99$ será igual a $(-39.410, -3.594)$.

(c) Bajo las condiciones del ejercicio (b), determina si existe diferencia en el nivel medio de colesterol entre jóvenes y adultos.

La salida del código anterior también nos da un p-valor (`p-value`) asociado al contraste (`alternative hypothesis: true difference in means is not equal to 0`):

$$\begin{cases} H_0: \mu_1 - \mu_2 = 0 \\ H_1: \mu_1 - \mu_2 \neq 0. \end{cases}$$

Como el p-valor es 0.00198, a un nivel de significación del 5% podemos rechazar la hipótesis nula de que la diferencia de de estas dos medias sea igual a 0. Esto se traduce en que, a ese nivel de significación, efectivamente podemos afirmar que la media de colesterol en adultos es distinta a la de los pacientes que acuden a pediatría.

(d) Bajo las condiciones del ejercicio (b), ¿existen evidencias para afirmar que el nivel medio de colesterol es menor en jóvenes que en adultos?

En este caso, queremos plantear el contraste unilateral por la izquierda:

$$\begin{cases} H_0: \mu_1 \geqslant \mu_2 \\ H_1: \mu_1 < \mu_2, \end{cases} \text{ o equivalentemente } \begin{cases} H_0: \mu_1 - \mu_2 \geqslant 0 \\ H_1: \mu_1 - \mu_2 < 0. \end{cases}$$

Esto lo podemos hacer, al igual que antes, indicándole a R que en la alternativa hay un símbolo menor que:

```
z.test(colesterolped, colesteroladu, sigma.x=30, sigma.y=10,
       alternative="less")
```

Como el p-valor es igual a 0.000991, para un nivel de significación del 5% llegaríamos a la conclusión de que, efectivamente, el nivel medio de colesterol en pacientes que acuden a pediatría está por debajo de los adultos.

(e) Supongamos que las varianzas poblacionales son desconocidas, pero tenemos información de que se puede asumir que son iguales. Calcula el intervalo de confianza al 99% para la diferencia de medias

En ese caso, el intervalo se obtiene con la función `t.test` y el argumento `var.equal = TRUE`.

```
t.test(colesterolped, colesteroladu, var.equal=TRUE, conf.level=0.99)
```

El intervalo de confianza para la diferencia de medias cuando las varianzas son desconocidas e iguales al nivel de confianza $(1-\alpha) = 0.99$ es igual a $(-35.494, -7.51)$.

(f) Se suele asumir que la media de colesterol en pacientes que acuden a pediatría es 20 unidades menor que las que acuden a medicina para adultos. Bajo las condiciones del ejercicio (e), estudia si en nuestro caso no ocurre esto.

En este caso, vamos a plantear el contraste:

$$\begin{cases} H_0: \mu_1 - \mu_2 = -20 \\ H_1: \mu_1 - \mu_2 \neq -20. \end{cases}$$

Si estamos bajo el mismo escenario que planteamos antes (varianzas son desconocidas e iguales), el p-valor asociado a este contraste se puede obtener del siguiente modo:

```
t.test(colesterolped, colesteroladu, var.equal=TRUE, mu=-20)
```

En ese caso, vemos que `p-value = 0.775`. Por tanto, a un nivel de significación $\alpha = 0.05$, no tenemos evidencias significativas para rechazar la hipótesis nula. Esto se traduce en que no podemos afirmar que la verdadera diferencia de medias entre jóvenes y adultos sea distinta de -20 unidades.

(g) Obtén el intervalo de confianza al 95% para la diferencia de medias cuando las varianzas son desconocidas y diferentes. En este escenario, estudia si la media de las dos poblaciones difiere.

El escenario que asume por defecto la función `t.test`, cuando no le pasamos ningún argumento a mayores, es aquel en el que no tenemos ninguna información acerca de las varianzas de las dos poblaciones, son desconocidas, y no se sabe si son iguales, por lo que se asume que son distintas. Este caso, que es difícil de resolver a mano, es, en general, el más realista y por eso R realiza esta opción por defecto. En el siguiente ejemplo, vamos a pasarle a la función `t.test` las dos muestras, de forma que va a calcular el intervalo de confianza al 95% para la diferencia de medias cuando las varianzas son desconocidas y diferentes.

```
t.test(colesterolped, colesteroladu)
```

Vemos que en este caso el intervalo de confianza para la diferencia de medias de colesterol al 95% es $(-34.105, -8.899)$ y que el p-valor es igual a 0.002. Por tanto, al nivel de significación del 5%, rechazaremos la hipótesis nula de que ambas medias sean iguales.

Caso práctico: Supongamos que queremos estudiar la diferencia de medias de colesterol entre adultos antes (cuya muestra está en `colesterolsinmed`) y después (la muestra es `colesterolmed`) de suministrarles una cierta medicación. Nótese que los adultos, a los que se le mide el colesterol, corresponden a los 30 primeros datos y son los mismos en ambas muestras.

(a) Obtén el intervalo de confianza al 95% para la diferencia de medias y estudia si la media de las dos poblaciones difiere.

En ese caso, deberemos indicarle a `t.test` que las muestras están apareadas con el argumento `paired = TRUE`.

```
colesterolmed=datos5b$colesterolmed[1:30]
t.test(colesteroladu, colesterolmed, paired=TRUE)
```

En este caso, el intervalo de confianza para la diferencia de medias de colesterol antes y después de tomar el medicamento al 95% es $(-19.448, 0.825)$ y al nivel de significación del 5%, podemos ver que no podemos rechazar la hipótesis nula de que el medicamento mantiene nivel medio de colesterol (`p-value = 0.070`). Nótese que si hubiésemos impuesto un valor de $\alpha = 0.1$, si que tendríamos que rechazar la hipótesis nula de igualdad, llegando a la conclusión de que el medicamento altera el nivel medio de colesterol.

4.4 Ejercicios propuestos

1. Para estudiar el efecto del ejercicio físico sobre el nivel de triglicérido, se ha realizado el siguiente experimento con 11 individuos: previo al ejercicio se tomaron muestras de sangre para determinar el nivel de triglicéridos por 100 mililitros de sangre de cada sujeto. Después los individuos fueron sometidos a un programa de ejercicios que se centraba diariamente en carreras y marchas. Al final del periodo de ejercicios, se tomaron nuevamente muestras de sangre y se obtuvo una segunda lectura del nivel de triglicérido. De este modo, se dispone de dos conjuntos de observaciones del nivel de triglicérido por 100 mililitros de sangre de los sujetos:

Individuo	1	2	3	4	5	6	7	8	9	10	11
Previo	68	77	94	73	37	131	77	24	99	629	116
Posterior	95	90	86	58	47	121	136	65	131	630	104

Suponiendo normalidad de la variable que mide el nivel de triglicérido.

   (a) Construye un intervalo de confianza de nivel 99% para la diferencia entre el nivel medio de triglicérido antes y después del programa de ejercicios.
   (b) Al nivel de significación del 5%, ¿hay pruebas suficientes para afirmar que el ejercicio físico aumenta el nivel medio de triglicérido?

2. Un médico está interesado en determinar si la concentración en sangre de una proteína en pacientes con cáncer de hígado es, en media, inferior a 15 mg/dl. Para evaluar esta posibilidad, obtiene una muestra aleatoria de veinte pacientes con esta enfermedad con las siguientes concentraciones (mg/dl):

13.7	15.8	16.1	14.1	10.5	15.2	19.0	12.8	15.0	19.2
13.6	16.5	13.5	14.4	16.7	10.9	15.1	13.3	12.3	14.3

Asumiendo normalidad, ¿constituyen estos resultados una prueba significativa de que la concentración media de los pacientes con cáncer de hígado es menor de 15 mg/dl?

3. Sabiendo que la variable que mide la concentración en sangre (g/dl) de una proteína en sangre sigue la distribución normal, se han tomado los siguientes datos de diez individuos sanos y nueve individuos que padecen una enfermedad neurológica grave:

Sanos	56.7	46.2	45.5	40.9	41.9	42.1	40.8	41.3	33.9	33.0
Enfermos	34.8	41.0	34.3	36.6	56.9	53.9	42.7	46.2	46.1	

(a) Asumiendo que las varianzas de ambas poblaciones son iguales, construye un intervalo de confianza de nivel 90% para la diferencia de concentraciones medias.

(b) Si no se pudiese asumir que las varianzas son iguales, al nivel de significación del 1%, ¿existirían evidencias de que la concentración media en individuos sanos difiere de la concentración media en enfermos?

4. Una compañía farmacéutica ha estado trabajando en la elaboración de un nuevo medicamento para bajar el nivel de colesterol. Para probar que es eficaz, se administra este producto a 200 pacientes con colesterol alto, y en 120 de ellos el colesterol disminuye. Se considera que este producto es efectivo si, después de su administración, el porcentaje de pacientes que han conseguido bajar el colesterol es superior al 50%. ¿Constituyen estos resultados una prueba significativa de la eficacia del medicamento, al nivel de significación del 5%?

5. La exostosis auditiva externa (EAE) es una anomalía ósea del canal auditivo externo. Esta lesión está asociada a una prolongada inmersión en agua fría y aparece con frecuencia en individuos que participan en actividades acuáticas como el surf. Se cree además que la temperatura del agua es un factor que influye en la prevalencia de EAE. Supongamos que en un estudio se examinan a 307 surfistas que surfean fundamentalmente en aguas frías (por debajo de 12ºC). De los 307 surfistas examinados, 230 fueron diagnosticados de EAE. En otro estudio realizado a 75 surfistas de aguas templadas, 30 fueron diagnosticados de EAE.

(a) Construye un intervalo de confianza al 99% para la diferencia de prevalencia de EAE entre surfistas de aguas frías y templadas.

(b) Para una significación del 5%, ¿se puede concluir que la prevalencia de EAE es significativamente mayor en los surfistas de aguas frías?

5 Contraste de independencia y medidas de riesgo

En el ámbito biomédico es habitual tratar con variables cualitativas categóricas, como el sexo, la raza, el tipo de tratamiento aplicado, o padecer (o no) una enfermedad, entre otros. Los contrastes de hipótesis para variables categóricas permiten responder preguntas sobre la posible asociación entre dos variables de este tipo. Por ejemplo, podría ser interesante estudiar si existe asociación entre el bajo peso al nacer y si la madre es fumadora.

En este capítulo presentaremos las funciones de R básicas para abordar esta clase de cuestiones. En la Sección 5.1, presentaremos la función para aplicar el test χ^2 de independencia en tablas 2×2. En la Sección 5.2, describiremos las correspondientes funciones para el cálculo de medidas de asociación entre variables categóricas, como el riesgo relativo y la odds-ratio.

5.1 Test Chi-cuadrado de independencia en tablas 2×2

El test χ^2 de independencia permite determinar si dos variables cualitativas X e Y están o no asociadas, esto es, se plantea el contraste:

$$\begin{cases} H_0: X \text{ e } Y \text{ son independientes} \\ H_1: X \text{ e } Y \text{ no son independientes.} \end{cases}$$

Este test comprueba la hipótesis de independencia comparando las frecuencias observadas con las frecuencias esperadas bajo la hipótesis de independencia de X e Y.

En R, la función para contrastar la independencia de dos variables categóricas es

```
chisq.test(x, correct)
```

donde

`x`	una matriz que contiene la tabla de contingencia observada.
`correct`	un valor lógico que indica si se aplica (`TRUE`) o no (`FALSE`) la corrección de continuidad al calcular el estadístico de contraste para tablas de 2×2: se resta la mitad de todas las diferencias entre los valores observados y esperados.

Caso práctico: La tabla de contingencia siguiente resume la información contenida en un estudio retrospectivo de control de casos para estudiar la relación de tener (caso) o no (control) cáncer de mama en mujeres y la edad a la que se tiene el primer hijo.

	Tipo	
Edad al tener el primer hijo	Caso	Control
$\geqslant 30$	683	1498
$\leqslant 29$	2537	8747

(a) Construye en R la correspondiente tabla de contingencia.

Las tablas de contingencia permiten organizar los datos en tablas de doble entrada, donde cada entrada representa un valor específico de la variable categórica. El código R necesario para su consideración es el que sigue.

```
M=matrix(c(683, 2537, 1498, 8747), nrow=2)
rownames(M)=c(">=30", "<30")
colnames(M)=c("Caso", "Control")
M
```

Recuerda que, por defecto, los elementos de una matriz se introducen por columna, tal y como se indica en la Sección 1.3.

(b) ¿Existe asociación entre el cáncer de mama y la edad a la que se tiene el primer hijo?

```
X=chisq.test(M, correct=TRUE)
X
```

La función `chisq.test` proporciona un p-valor muy próximo a cero ($< 2.2 \cdot 10^{-16}$), claramente menor que los niveles de significación habituales. Por tanto, existen evidencias para rechazar la hipótesis nula de independencia entre las variables "tener cáncer de mama" y "edad". En este ejemplo, el valor del estadístico asociado a este contraste, usando la corrección de continuidad (`correct=TRUE`), es 77.89. Esta corrección nos garantiza que la aproximación de la distribución del estadístico de contraste por la distribución Chi-cuadrado sea buena.

(c) ¿Cuales son los valores observados y los valores esperados bajo la independencia de ambas variables?

```
X$observed   # Valores observados
X$expected   # Valores esperados
```

El código anterior permite recuperar la matriz con los valores observados, que se corresponden con la matriz originalmente introducida, y los valores esperados. Por ejemplo, si ambas variables fuesen independientes, el número de casos esperados con edad superior o igual que 30 y que hayan sido diagnosticados de cáncer es 521.561.

5.2 Medidas de efecto: riesgo relativo y odds-ratio

La relación entre las variables se puede cuantificar mediante el cálculo de medidas de asociación como el riesgo relativo (RR) y la odds-ratio (OR). En R, se emplea el paquete `BioProbability` para la determinación de ambas medidas de asociación en tablas de contingencia 2×2.

```
install.packages("BioProbability")
library(BioProbability)
```

5.2.1 Riesgo relativo

El riesgo relativo (RR) es una razón que relaciona la incidencia en dos grupos de población, que difieren por el grado de exposición a un factor determinado.

En R, la función `relative.risk` permite el cálculo de este riesgo relativo.

```
relative.risk(M, conf.int, level)
```

donde

`M` una matriz 2×2 donde su primera fila contiene el número de casos expuestos a una condición (M_{11}) y los no expuestos (M_{12}). Su segunda fila contiene el número de los no considerados como casos: los expuestos (M_{21}) y los no expuestos (M_{22}).

`conf.int` indicador lógico de si se quiere (`TRUE`) o no (`FALSE`) calcular el intervalo de confianza para el riesgo relativo.

`level` valor de α para calcular el intervalo de confianza al nivel $1-\alpha$.
Por defecto, se realiza el intervalo al nivel 0.95.

Caso práctico (continuación): El estudio retrospectivo permite estudiar la relación del cáncer de mama en mujeres y la edad a la que se tiene el primer hijo.

(d) Calcula el riesgo relativo.

En este caso, se ha empleado la siguiente línea de código. Debemos indicar en la función `relative.risk` la tabla de contingencia en `M` como argumento.

```
relative.risk(M, conf.int=TRUE, level=0.05)
```

El riesgo relativo es igual a 1.393. Dado que el valor del riesgo relativo es mayor que 1, tener el primer hijo con más de 30 años incrementa el riesgo de padecer cáncer de mama.

(e) Determina el intervalo de confianza para el riesgo relativo al 95% de confianza.

La función `relative.risk` también proporciona dicho intervalo de confianza para el riesgo relativo, al considerar las opciones `conf.int=TRUE` y `level=0.05`. En este caso, el intervalo de confianza para el riesgo relativo al 95% de confianza es $(1.297, 1.495)$.

5.2.2 Odds-ratio

En la práctica, el número de individuos clasificados como enfermos es mucho más pequeño que el número de sujetos clasificados como no enfermos. Entonces, la Odds-ratio es una medida de riesgo alternativa. En R, la función `odds.ratio` permite el cálculo de este riesgo relativo.

```
odds.ratio(M,  conf.int, level)
```

donde

`M` una matriz 2×2 donde su primera fila contiene el número de casos expuestos a una condición (M_{11}) y los no expuestos (M_{12}). Su segunda fila contiene el número de los no considerados como casos: los expuestos (M_{21}) y los no expuestos (M_{22}).

`conf.int` indicador lógico de si se quiere (`TRUE`) o no (`FALSE`) calcular el intervalo de confianza para la Odds-ratio.

`level` valor de α para calcular el intervalo de confianza al nivel $1-\alpha$.
Por defecto, se realiza el intervalo al nivel 0.95.

Caso práctico (continuación): Considera de nuevo el estudio retrospectivo para estudiar la relación del cáncer de mama en mujeres y la edad a la que se tiene el primer hijo.

(f) Calcula la Odds-ratio.

Para su cómputo, se usa

```
odds.ratio(M, conf.int=TRUE, level=0.05)
```

Así, la Odds-ratio es igual a 1.572 en este caso. De nuevo, como su valor es mayor que 1, se comprueba el efecto del factor de riesgo "tener el primer hijo con más de 30 años" sobre "tener cáncer de mama".

(g) Determina el intervalo de confianza para el riesgo relativo al 95% de confianza.

La función `odds.ratio` también proporciona el intervalo de confianza para la Odds-ratio, al considerar las opciones `conf.int=TRUE` y `level=0.05`. En este caso, el intervalo de confianza para la Odds-ratio al 95% de confianza es $(1.421, 1.739)$.

5.3 Ejercicios propuestos

1. Se ha realizado un estudio prospectivo o de cohorte para estudiar si existe relación entre el infarto de miocardio y el uso de anticonceptivos orales. Para ello, se consideraron dos grupos de mujeres de 40 a 44 años de edad que no han padecido infarto de miocardio (IM). Un grupo toma habitualmente anticonceptivos orales (AO) y el otro no. Se controlaron durante 3 años a los dos grupos de mujeres y se observó la incidencia de infarto IM al cabo de 3 años:

	IM al cabo de 3 años	
Uso AO	Sí	No
Si	13	4987
No	7	9993

   (a) ¿Existe asociación entre ambas variables?

   (b) Calcula el riesgo relativo y la Odds-ratio.

2. En un estudio transversal o de prevalencia, se seleccionó una muestra de inviduos individuos en una población ($n = 1570$) que ya cumplieron los 40 años y se clasificaron atendiendo a la edad y a la diabetes:

	Diabetes	
Edad	Si	No
$\geqslant$55	84	1066
$<$55	11	409

   (a) ¿Existe asociación entre ambas variables?

   (b) Calcula el riesgo relativo y la Odds-ratio.

6 Modelo de regresión lineal simple

El objetivo de un modelo de regresión es tratar de explicar la relación que existe entre una variable dependiente (variable respuesta) Y un conjunto de variables independientes (variables explicativas). En particular, un modelo de regresión lineal simple trata de explicar la relación que existe entre la variable respuesta Y y una única variable explicativa X a través de la ecuación de una recta que permite estimar el valor medio de Y a partir de los valores de X y realizar predicciones acerca de la variable Y.

En la Sección 6.1, presentaremos las funciones de R básicas para determinar el modelo de regresión lineal simple, así como para predecir nuevas observaciones.

6.1 El modelo de regresión lineal

En el estudio de variables bidimensionales, la relación más sencilla entre un par de variables cuantitativas X e Y es la dependencia lineal, donde se supone que la relación entre ellas viene dada por una recta. La recta de regresión ajustada de Y sobre X se corresponde con

$$y = \hat{\beta}_0 + \hat{\beta}_1 x$$

donde $\hat{\beta}_0$ y $\hat{\beta}_1$ son los coeficientes estimados a partir de los pares de datos muestrales (x_i, y_i), con $i = 1, \cdots, n)$.

La sintaxis siguiente permite realizar el ajuste del modelo lineal y obtener resultados relacionados con el ajuste en R.

```
modelo=lm(y~x)
summary(modelo)
```

siendo `y` el vector de datos conteniendo la variable respuesta y `x` el correspondiente a la variable explicativa.

En la salida de la función `summary` podemos distinguir los siguientes elementos:

- Bajo el título `Call`, se presenta el modelo en lenguaje R.
- Bajo el título `Residuals`, se presenta un estudio descriptivo, con el mínimo, primer cuartil, mediana, tercer cuartil y máximo de los residuos.
- Bajo el título `Coefficients`, se ofrecen los elementos básicos de inferencia para los coeficientes de la recta.
  - `Estimate`: Son las estimaciones de los coeficientes $\hat{\beta}_0$ y $\hat{\beta}_1$.
  - `Std.Error`: Son los errores típicos de los coeficientes.
  - `t value`: Son los cocientes entre las estimaciones y los errores típicos. Es el estadístico para el contraste de que el coeficiente vale cero.
  - `Pr(>t)`: Son los p-valores para el contraste de que el coeficiente vale cero.

  Los asteriscos que acompañan a los p-valores de los coeficientes indican si hay significación según la línea titulada `Signif.codes`.

Caso práctico: El Volumen Expiratorio Forzado (VEF) es una medida de la función pulmonar. Se cree que el VEF está relacionado con la estatura. Nos interesa estudiar la variable bidimensional (X, Y) siendo X la estatura de niños de 10 a 15 años de edad e Y el VEF.

A continuación, se muestra la estatura (en cm) y el VEF (en l) de una muestra de 12 niños en ese rango de edad:

Estatura	134	138	142	146	150	154	158	162	166	170	174	178
VEF	1.7	1.9	2.0	2.1	2.2	2.5	2.7	3.0	3.1	3.4	3.8	3.9

(a) ¿Existe algún tipo de relación entre el VEF y la estatura?

Para comprobar la relación entre ambas variables en R, usamos el siguiente código para la definición de los datos de ambas variables.

```
Estatura=c(134, 138, 142, 146, 150, 154, 158, 162, 166, 170, 174, 178)
VEF=c(1.7, 1.9, 2.0, 2.1, 2.2, 2.5, 2.7, 3.0, 3.1, 3.4, 3.8, 3.9)
```

Gráficamente, el diagrama de dispersión permite representar los pares de las observaciones (x_i, y_i) y chequear la posible relación entre las variables. En R, es posible determinarlo usando la función `plot`.

```
plot(Estatura, VEF)
```

Como resultado, de la Figura 11, se observa una clara relación lineal entre ambas variables, de forma que aumentos de la estatura parece que provocan incrementos en el VEF, y además lo hacen de forma lineal.

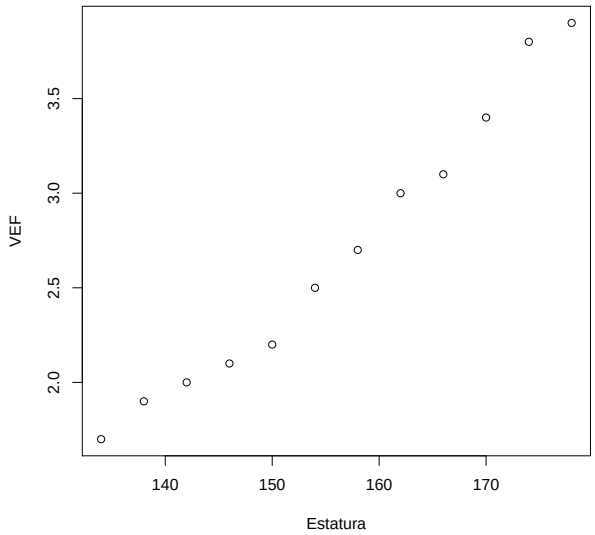


Figura 11: *Diagrama de dispersión de las variables Estatura y Volumen Expiratorio Forzado (VEF).*

Numéricamente, la covarianza entre X e Y puede interpretarse como una medida de relación lineal de ambas variables. En R, la covarianza se determina usando `cov(x,y)`, donde los argumentos `x` y `y` denotan a las dos variables cuantitativas involucradas.

La función `cov(x,y)` de R determina el valor de la cuasicovarianza de los vectores `x` e `y` dividiendo entre el número total de datos (denotado por n) menos 1, esto es,

$$\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y}).$$

En el ejemplo considerado, la cuasicovarianza es igual a 10.673, y se obtiene usando

```
cov(Estatura, VEF)
```

El signo de la cuasicovarianza indica la ya detectada relación directa entre las variables, es decir, a medida que aumenta la estatura aumenta el VEF. Sin embargo, en general, esta medida no permite comparar la relación entre pares de variables medidas en unidades muy diferentes.

El coeficiente de correlación lineal, que toma valores entre -1 y 1, sirve para investigar la relación lineal entre las variables en esos casos. En R,

```
cor(Estatura, VEF)
```

proporciona su valor. En este ejemplo, la correlación es 0.988 (próxima a 1) y, por lo tanto, la relación entre el Volumen Expiratorio Forzado y la Estatura es directa.

(b) Calcula y representa la recta de regresión del VEF sobre la estatura.

El siguiente código permite estimar los coeficientes de la recta de regresión del Volumen Expiratorio Forzado y la Estatura:

```
modelo=lm(VEF~Estatura)
summary(modelo)
beta0=modelo$coefficients[1]; beta0
beta1=modelo$coefficients[2]; beta1
```

Como resultado, los coeficientes de la recta de regresión de Y sobre X son

$$\hat{\beta}_0 = -5.313 \text{ y } \hat{\beta}_1 = 0.051.$$

Tal y como se puede ver en la salida de `summary`, concretamente en `Pr(>|t|)`, $\hat{\beta}_0$ y $\hat{\beta}_1$ son significativamente distintos de cero para los niveles de significación habituales (los p-valores asociados a los correspondientes contrastes son, respectivamente, $9.82\cdot 10^{-8}$ y $1.80\cdot 10^{-9}$). Además, como cabría esperar del cómputo de la cuasicovarianza y de la correlación, la pendiente de la recta $\hat{\beta}_1$ tiene signo positivo, indicando el carácter creciente de la recta de regresión.

Para completar el análisis descriptivo, las siguientes dos líneas en R permiten la representación de la recta de regresión obtenida, a través del comando `abline` sobre el diagrama de dispersión (ver Figura 12).

```
plot(Estatura, VEF)
abline(modelo, col="blue")
```

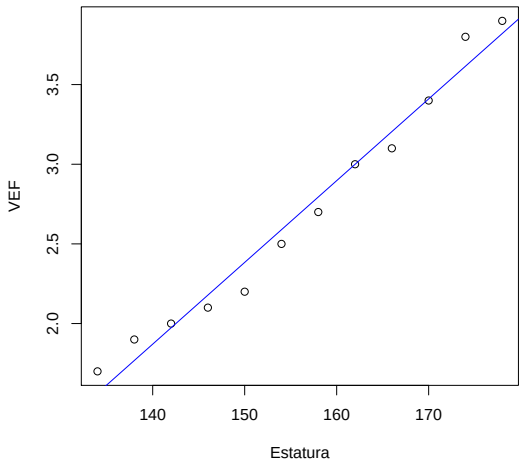


Figura 12: *Diagrama de dispersión de Estatura y Volumen Expiratorio Forzado (VEF) y su recta de regresión (en azul).*

(c) ¿Cuál es el porcentaje de variabilidad explicada por el modelo de regresión?

El coeficiente de determinación (R^2) es el porcentaje de la varianza total explicada por la recta de regresión. Se calcula como el cuadrado del coeficiente de correlación entre la variable explicativa y la respuesta, es decir, $R^2 = r_{xy}^2$.

Sobre la salida de `summary(modelo)` en R, es inmediato conocer que el coeficiente de determinación R^2 es 0.9765 (ver `Multiple R-squared`) para el ejemplo considerado. Es decir, con el modelo de regresión lineal simple asociado, la variable *Estatura* es capaz de explicar el 97.65% de la variabilidad de VEF.

(d) ¿Qué valor de Volumen Expiratorio Forzado cabe esperar para una niña de 140 cm de altura? ¿y para un niño de 150 cm de altura?

Si almacenamos el vector de las nuevas alturas observadas, basta con hacer en R:

```
Estatura0=c(140, 150)
beta0+beta1*Estatura0
```

para ver que las predicciones de Volumen Expiratorio Forzado para las estaturas de la niña y el niño son iguales a 1.871 y 2.384, respectivamente.

Alternativemente, la función `predict` permite obtener de manera inmediata en R dichas predicciones:

```
predict(modelo, data.frame(x=x0))
```

donde

`modelo`	el correspondiente modelo de regresión ya usado anteriormente.
`x`	la correspondiente variable independiente en `modelo`.
`x0`	la nueva observación de X sobre la que predecir. Nótese que `x0` puede ser incluso un vector de nuevas observaciones de X.

Obtenida la recta de regresión en el ejemplo, para predecir el Volumen Expiratorio Forzado para un niño y una niña con alturas de 140 cm y de 150 cm respectivamente, se usa

```
predict(modelo, data.frame(Estatura=Estatura0))
```

6.2 Ejercicios propuestos

1. Se registra la evolución del nivel de creatinina en pacientes tratados con Captopril después de ser sometidos a diálisis. Los resultados obtenidos se muestran en la tabla siguiente:

Días transcurridos	1	5	10	15	20	25	35
Creatinina (mg/dl)	5.7	5.2	4.8	4.5	4.2	4	3.8

   (a) A partir de los datos anteriores, ajusta un modelo de regresión lineal simple en el que el nivel de creatinina (Y) se exprese en función de los días transcurridos (X).

   (b) Calcula el porcentaje de variabilidad explicada por el modelo de regresión.

2. Se lleva a cabo un estudio, por medio de detectores radioactivos, sobre la capacidad corporal para absorber hierro y plomo. En el estudio participaron 6 personas y después de 10 días se obtuvieron los siguientes resultados.

Hierro	1.7	2.2	3.5	4.3	8.0	6.0
Plomo	2.1	3.0	1.8	2.5	4.2	4.0

   (a) Representa el diagrama de dispersión de los datos.

   (b) Calcula y representa la recta de regresión del valor del plomo sobre el valor del hierro.

   (c) ¿Cuál es el coeficiente de correlación lineal?

   (d) Calcula el porcentaje de variabilidad explicada por la recta de regresión.

   (e) ¿Qué valor de plomo cabe esperar para una persona con un nivel de hierro igual a 2.2?

A Listado de funciones principales

Función	Descripción	Página
`abline`	Representación de la recta de regresión	68
`barplot`	Diagrama de barras	28
`boxplot`	Diagrama de caja	33
`c`	Creación de un vector	13
`cbind`	Concatenar vectores	27
`chisq.test`	Test χ^2 de independencia	62
`colnames`	Nombres de columnas en una matriz/data frame	62
`cov`	Covarianza de dos muestras	67
`cor`	Coeficiente de correlación lineal	68
`cumsum`	Sumas acumuladas	27
`data.frame`	Creación de una base de datos	18
`dbinom`	Función de masa de probabilidad de la binomial	36
`dexp`	Función de densidad de la exponencial	42
`dnorm`	Función de densidad de la normal	39
`dpois`	Función de masa de probabilidad de la Poisson	37
`dweibull`	Función de densidad de la weibull	43
`exp`	Función exponencial	9
`fitdistr`	Ajuste de distribuciones	45
`head`	Primeras filas de un data frame	19
`help`	Ayuda de una función	20
`hist`	Histograma	32
`kurtosis`	Coeficiente de apuntamiento de una muestra	32
`install.packages`	Instalar un paquete	20
`length`	Número de elementos de un vector	14
`library`	Cargar un paquete	20
`lm`	Ajuste de modelos lineales de regresión	66
`log`	Logaritmo natural	9
`matrix`	Creación de una matriz	17
`max`	Máximo	14
`mean`	Media	31
`median`	Mediana	31
`min`	Mínimo	14
`names`	Nombres de un objeto	19
`ncol`	Número de columnas de una matriz/data frame	19
`nrow`	Número de filas de una matriz/data frame	19
`odds.ratio`	Odds-ratio	64
`pbinom`	Función de distribución de la binomial	36
`pexp`	Función de distribución de la exponencial	42
`pie`	Diagrama de sectores	28
`plot`	Diagrama de dispersión	67
`pnorm`	Función de distribución de la normal	39
`points`	Añadir puntos a un gráfico	41
`ppois`	Función de distribución de la Poisson	38
`predict`	Predicciones asociadas a un modelo de regresión	69
`prop.test`	Inferencia para la proporción	48
`pweibull`	Función de distribución de la weibull	44

Función	Descripción	Página
`qbinom`	Función cuantil de la binomial	36
`qexp`	Función cuantil de la exponencial	42
`qnorm`	Función cuantil de la normal	39
`qpois`	Función cuantil de la Poisson	38
`quantile`	Cuantiles de una muestra	31
`qweibull`	Función cuantil de la weibull	44
`quantile`	Cuantiles de una muestra	31
`read.table`	Lectura de un archivo .txt	21
`read.csv`	Lectura de un archivo .csv	21
`range`	Mínimo y máximo de una muestra	31
`relative.risk`	Riesgo relativo	63
`rownames`	Nombres de filas en una matriz/data frame	62
`sd`	Cuasidesviación típica de una muestra	31
`setwd`	Cambiar el directorio de trabajo	21
`skewness`	Coeficiente de asimetría de una muestra	32
`sort`	Ordenación de un vector	14
`sqrt`	Raíz cuadrada	9
`sum`	Suma de todas las componentes de un vector	14
`summary`	Resumen de medidas	66
`t.test`	Inferencia para la media de una normal con varianza desconocida	54
`table`	Tablas de frecuencias	27
`var`	Cuasivarianza de una muestra	31
`z.test`	Inferencia para la media de una normal con varianza conocida	54

a

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA