5. Tratamiento de datos ausentes

La existencia de datos ausentes, también conocidos como *missing values* y representados habitualmente como `NA` en R, es una casuística habitual en muchas bases de datos. La mayoría de las veces se deben a problemas durante la recopilación de datos, por ejemplo la incapacidad para obtener una cierta medida o respuesta, o fallos en la transcripción.

Al leer datos de una fuente externa, por ejemplo un archivo CSV, los datos ausentes pueden aparecer como comillas vacías, estar representadas por un cierto valor clave o, sencillamente, estar ausentes. Funciones como `read.table()` permiten indicar qué casos han de ser interpretados como valores ausentes y, en consecuencia, aparecer como `NA` en el *data frame*.

Tratar con datasets en los que existen datos ausentes puede generar diversos problemas. Por dicha razón en este capítulo aprenderemos a tratar con ellos en R, a fin de evitar los incovenientes que suelen producir.

5.1 Problemática

La presencia de datos ausentes dificulta la mayoría de operaciones matemáticas y de análisis. ¿Cuál es el resultado de sumar `NA` a cualquier número? ¿Es `NA` menor o mayor que un cierto valor? ¿Cómo se calcula la media de una lista de valores en los que aparece `NA`? Estos son algunos casos sin respuesta y, en consecuencia, el resultado de todos ellos es también `NA`.

El siguiente ejercicio genera un conjunto de valores que, hipotéticamente, se han obtenido de una encuesta. Cinco de los encuestados no han respondido, por lo que el valor asociado es `NA`. En los pasos siguientes se efectúan algunas operaciones cuyo resultado puede apreciarse en la consola:

Ejercicio 5.1 Algunas operaciones con datos ausentes

```
> # Número de horas trabajadas semanalmente en una encuesta
> valores <- as.integer(runif(50,1,10))
> indices <- as.integer(runif(5,1,50))  # Sin respuesta 5 casos
```

```
> valores[indices] <- NA
> valores

 [1]  1  4  6  4  1  5  4  3  6 NA  7 NA  3  2  5  8  9  4  3  1
[21]  9  4  1  6  1  6  5  4  9  8  5  3  4  7  6  7  4  7 NA  5
[41] NA  8  9  7  3  1  5  6  4  7

> valores > 5 # Los valores NA no pueden ser comparados

 [1] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE    NA
[11]  TRUE    NA FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE
[21]  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE
[31] FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE    NA FALSE
[41]    NA  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE

> valores + 10 # Ni se puede operar con ellos

 [1] 11 14 16 14 11 15 14 13 16 NA 17 NA 13 12 15 18 19 14 13 11
[21] 19 14 11 16 11 16 15 14 19 18 15 13 14 17 16 17 14 17 NA 15
[41] NA 18 19 17 13 11 15 16 14 17

> mean(valores)

[1] NA
```

5.2 Detectar existencia de valores ausentes

Antes de operar con un conjunto de datos, por tanto, deberíamos verificar si existen valores ausentes y, en caso afirmativo, planificar cómo se abordará su tratamiento. Con este fin podemos usar funciones como `is.na()` y `na.fail()`, entre otras.

Sintáxis 5.1 `is.na(objeto)`
Devuelve `TRUE` si el objeto es un valor ausente o `FALSE` en caso contrario. Si el objeto es compuesto, como un vector, una matriz o *data frame*, la comprobación se efectúa elemento a elemento.

Sintáxis 5.2 `na.fail(objeto)`
En caso de que el objeto facilitado como argumento contenga algún valor ausente, esta función genera un error y detiene la ejecución del guión o programa.

En caso de que solamente queramos saber si un objeto contiene valores ausentes o no, sin obtener un vector lógico para cada elemento, podemos combinar la salida de `is.na()` mediante la función `any()`, tal y como se muestra en el siguente ejercicio:

Ejercicio 5.2 Detectar la presencia de valores nulos antes de operar

```
> is.na(valores)

 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
[11] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
[21] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[31] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
[41]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

> any(is.na(valores))

[1] TRUE

>

> na.fail(valores)

Error in na.fail.default(valores) : missing values in object
```

5.3 Eliminar datos ausentes

Dependiendo de cómo vayamos a operar sobre los datos, es posible que antes de trabajar con ellos prefiramos eliminar los datos ausentes para evitar problemas como los antes expuestos. Con este fin recurriremos a la función `na.omit()`:

Sintáxis 5.3 `na.omit(objeto)`
Eliminará del objeto entregado como argumento cualquier dato ausente que exista, devolviendo un objeto del mismo tipo sin dichos valores. Los indices que ocupaban los datos ausentes se facilitan en un atributo asociado al objeto y llamado `na.action`.

Otra posibilidad consiste en utilizar la función `complete.cases()`. Esta resulta especialmente útil al trabajar con *data frames*, ya que verifica que ninguna de las columnas de cada fila contenga valores ausentes. El valor devuelto es un vector de lógicos, con `TRUE` en las filas completas (sin valores ausentes) y `FALSE` en las demás. Dicho vector puede ser utiliza para seleccionar las filas que interesen.

Sintáxis 5.4 `complete.cases(objeto)`
Devuelve un vector de valores lógicos indicando cuáles de las filas del objeto entregado como parámetro están completas, no conteniendo ningún valor ausente.

El dataset integrado `airquality` contiene 42 filas con valores ausentes de un total de 153 observaciones. En el siguiente ejercicio se muestra cómo obtener únicamente las filas sin valores nulos, ya sea utilizando `na.omit()` o `complete.cases()`:

Ejercicio 5.3 Eliminación de valores ausentes

```
> str(airquality)

'data.frame':        153 obs. of  6 variables:
 $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
 $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
 $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
 $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
 $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
 $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
> nrow(airquality)

[1] 153

> nrow(na.omit(airquality))

[1] 111

> nrow(airquality[complete.cases(airquality),])

[1] 111
```

5.4 Operar en presencia de datos ausentes

Algunas funciones R están preparadas para trabajar en presencia de datos ausentes, aceptando un parámetro que determina cómo han de ser tratados. Un par de ejemplos de este caso son las funciones `mean()` y `lm()`, usadas para obtener el valor promedio (media aritmética) y ajustar un modelo lineal. La primera acepta el parámetro `na.rm`, de tipo lógico, con el que se indica si los valores ausentes deben ser ignorados durante el cálculo o no. La segunda tiene un parámetro llamado `na.action` que, entre otros, acepta el valor `omit`, con exactamente el mismo resultado.

En ocasiones, en lugar de eliminar filas completas de datos de un *data frame* lo que se hace es sustituir los valores ausentes por el valor promedio de la columna en la que aparece, o bien con el valor más frecuente o bien algún valor especial. En el siguiente ejercicio se pone en práctica la primera técnica:

Ejercicio 5.4 Operar en presencia de valores ausentes

```
> promedio <- mean(valores, na.rm = TRUE)
> promedio

[1] 4.934783

> valores[is.na(valores)] <- promedio
> mean(valores)

[1] 4.934783

> lm(Solar.R ~ Temp, airquality, na.action=na.omit)

Call:
lm(formula = Solar.R ~ Temp, data = airquality, na.action = na.omit)

Coefficients:
(Intercept)         Temp
    -24.431        2.693
```