Subsetting en R

Dr. Samuel D. Gamboa Tuz

Subsetting

- Sirve para extraer datos de algún objeto en R utilizándo **índices** (son vectores).
- Se utilizan tres operadores: `[]`, `[[]]` y `$`.
- Estos operadores pueden comportarse de manera diferente dependiendo del tipo de datos.

Vectores atómicos

- Se puede aplicar *subsetting* a vectores atómicos de varias formas:
  - Usando un vector con números positivos.
  - Usando un vector con números negativos (excluir elementos).
  - Usando un vector de caracteres (cuando los elementos están nombrados).
  - Usando un vector lógico (condicional).
- Se puede usar `[[]]` para seleccionar un solo elemento usando un índice numérico positivo o de caracter, pero es igual a `[]` (solo en vectores atómicos).
- No se puede usar el `$` en vectores atómicos.

- Con números positivos:

```
(x <- c(a = 1, b = 2, c = 3, d = 5, e = 5))
```

```
## a b c d e
## 1 2 3 5 5
```

```
x[1] # Seleccionar posición 1.
```

```
## a
## 1
```

```
x[1:3] # Seleccionar las posiciones 1, 2 y 3.
```

```
## a b c
## 1 2 3
```

```
x[c(1,3)] # Seleccionar las posiciones 1 y 3.
```

```
## a c
## 1 3
```

```
x[c(1, 1, 1, 1)] # Seleccionar la posición 1 cuatro veces.
```

```
## a a a a
## 1 1 1 1
```

- Con números negativos (excluir elementos):

```
x[-1] # Excluir la posición 1.
```

```
## b c d e
## 2 3 5 5
```

```
x[-c(1, 3)] # Excluir las posiciones 1 y 3.
```

```
## b d e
## 2 5 5
```

```
x[-1:-3] # Escluir las posiciones 1, 2 y 3.
```

```
## d e
## 5 5
```

- No se pueden combinar índices negativos y positivos.

- Con un vector de caracteres:

```
x["a"] # Seleccionar una posición por nombre.
```

```
## a
## 1
```

```
x[c("a", "d")] # Seleccionar varias posiciones por nombre.
```

```
## a d
## 1 5
```

- Con un vector lógico o con una condición:

```
x[c(TRUE, FALSE, TRUE, FALSE, FALSE)] # Seleccionar con un vector lógico.
```

```
## a c
## 1 3
```

```
x[x > 3] # Seleccionar con una condición.
```

```
## d e
## 5 5
```

Subsetting en matrices

- Se puede extraer o fitrar elementos de una matrix usando uno o dos índices dentro de `[]`.
- Cuando se extrae información con un solo índice, el operador `[]` se comporta como un vector atómico; siempre en dirección de las columnas.

```
(m <- matrix(1:9, ncol = 3, byrow = TRUE))
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

```
m[1]
```

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

```
m[1:3]
```

```
## [1] 1 4 7
```

- Cuando se extrae información con dos índices, el operador `[]` permite seleccionar filas y columnas en una matriz.

```
m[1:2, c(2,3)]
```

```
##      [,1] [,2]
## [1,]    2    3
## [2,]    5    6
```

- También se puede seleccionar por nombres:

```
dimnames(m) <- list(c("a", "b", "c"), c("A", "B", "C"))
m
```

```
##   A B C
## a 1 2 3
## b 4 5 6
## c 7 8 9
```

```
m[ "a", c("A", "B"), drop = FALSE]
```

```
##   A B
## a 1 2
```

- `drop = FALSE` sirve para evitar que el resultado sea un vector atómico cuando el resulado es un solo valor, fila o columna.

- También se pueden usar condicionales para filtrar por filas en una matriz con *subsetting*.

```
m[m > 1]
```

```
## [1] 4 7 2 5 8 3 6 9
```

```
m[m[,"A"] > 1, c("B", "C")]
```

```
##   B C
## b 5 6
## c 8 9
```

- **Subassignment**. Se pueden poner todos los resultados en 0 (o cualquier otro número) usando el opearador `[]` vacío. Puede ser útil al escribir funciones.

```
m2 <- m
m2[] <- 0
m2
```

```
##   A B C
## a 0 0 0
## b 0 0 0
## c 0 0 0
```

- Dejar un índice vacío es equivalente a seleccionar todas las filas o columnas:

```
m[, c("A", "B")] # Selecciona todas las filas
```

```
##   A B
## a 1 2
## b 4 5
## c 7 8
```

```
m[c("a", "b"), ] # Selecciona todas las columnas
```

```
##   A B C
## a 1 2 3
## b 4 5 6
```

Subsetting en listas

- El resultado de usar el operador `[]` en una lista es una sublista.

```
list1 <- list(1:5, letters[1:5], month.abb[1:5])
list1[1]
```

```
## [[1]]
## [1] 1 2 3 4 5
```

```
typeof(list1[1])
```

```
## [1] "list"
```

- Para acceder a los valores (como vector atómico) es necesario usar `[[]]`:

```
list1[[1]]
```

```
## [1] 1 2 3 4 5
```

```
typeof(list1[[1]])
```

```
## [1] "integer"
```

- Cuando la lista está nombrada se puede usar `$`, que prácticamente funciona igual que `[[]]`, pero puede "auto-acompletar" el nombre:

```
names(list1) <- c("numbers", "letters", "months")
list1$months
```

```
## [1] "Jan" "Feb" "Mar" "Apr" "May"
```

```
list1$m # Tener cuidado con esto, mejor usar nombre completo siempre.
```

```
## [1] "Jan" "Feb" "Mar" "Apr" "May"
```

```
list1[["months"]]
```

```
## [1] "Jan" "Feb" "Mar" "Apr" "May"
```

```
list1["months"] # Arroja una lista.
```

```
## $months
## [1] "Jan" "Feb" "Mar" "Apr" "May"
```

- Se puede usar un segundo índice usando el operador `[]` para extraer posiciones del vector.

```
list1[1][1] # Arroja una lista.
```

```
## $numbers
## [1] 1 2 3 4 5
```

```
list1[1][2] # Solo puede usarse una posición (1).
```

```
## $<NA>
## NULL
```

```
list1[[2]][2:3]
```

```
## [1] "b" "c"
```

```
list1[["months"]][2]
```

```
## [1] "Feb"
```

```
list1$letters[1:3]
```

```
## [1] "a" "b" "c"
```

Subsetting en *data frames*

- Se puede extraer con un solo índice como en las listas o con dos índices como en las matrices.

```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
head(iris$Species) # Igual que iris[["Species"]]
```

```
## [1] setosa setosa setosa setosa setosa setosa
## Levels: setosa versicolor virginica
```

```
head(
  iris[ iris$Sepal.Length > 5 , grep("^S", colnames(iris))],
  3)
```

```
##    Sepal.Length Sepal.Width Species
## 1           5.1         3.5  setosa
## 6           5.4         3.9  setosa
## 11          5.4         3.7  setosa
```

Selecciona un vector atómico

```
df$col2[1]
df[["col2"]][1]
df[[2]][1]
df[,"col2"][1]
df[,2][1]
```

	col1	col2	col3
row1	"a"	1	T
row2	"b"	2	F
row3	"c"	3	T
row4	"d"	4	F
row5	"e"	5	T

Selecciona un df

```
df[,c("col2, col3")][1]
df[,c(2,3)][1]
df["col2"]
df[2]
```

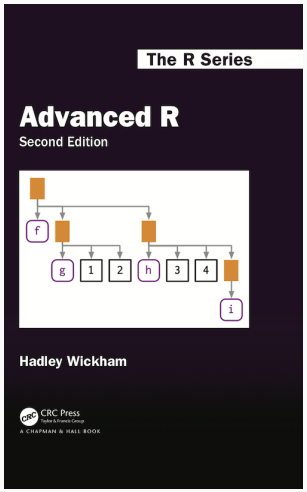
	col1	col2	col3
row1	"a"	1	T
row2	"b"	2	F
row3	"c"	3	T
row4	"d"	4	F
row5	"e"	5	T

Actividades recomendadas

- ¿Para qué sirve la función `subset()` y cómo se utiliza?
- ¿Cómo filtrarías las filas con `NA`s en una columna especifica de un *data frame*?
- ¿Cómo seleccionarías únicamente las columnas con vectores tipo *integer* o *double* de un *data frame*?

Bibliografía recomendada

Advanced R - Hadley Wickham



Capítulo 4 - Subsetting

https://adv-r.hadley.nz/subsetting.html