

Taller de R. Clase 03

Dr. Isaías Moreno Cruz

30-09-2024

Matriz

```
df <- read.table("./data/xyz.dat", sep="," , header=FALSE)
names(df) <- c("x", "y", "z")
head(df)
dim(df)
n<- sqrt(length(df$z))
M <- matrix(df$z, nrow=n, ncol=n)
```

Ejercicio II

```
library(plot.matrix)  
plot(M)
```

cbind-ing y rbind-ing

Las matrices pueden ser creadas por column-binding (unión de columnas) o row-binding con `cbind()` y `rbind()`

R

```
x <- 1:3  
y <- 10:12  
cbind(x,y)  
rbind(x,y)
```

La lista es un tipo especial de vector que puede contener elementos de diferente clase. Las listas son muy importantes en R.

R

```
x <- list(1, "a", TRUE, 1+4i)
```

```
x
```

Factors

Los factores son usados para representar una categoria.

R

```
x <- factor(c("yes", "yes", "no", "yes", "no"))
```

```
x
```

```
table(x)
```

```
unclass(x)
```

El orden de los niveles pueden ser fijados utilizando el argumento `levels` en la función `factor()`. Esto puede ser importante en modelos lineales porque el primer nivel es usado como linea base.

R

```
x <- factor(c("yes", "yes", "no", "yes", "no"),  
           levels=c("yes", "no"))
```

```
x
```

Missing Values

Los valores perdidos son denotados por NA o NaN para definir una operación matemática indefinida.

- `is.na()` usado para probar si hay Na en un objeto
- `is.nan()` usado para probar si hay NaN en un objeto
- El valor de NA tienen clase.
- El valor de NaN es también NA, pero no viceversa.

R

```
x <- c(1, 2, NA, 10, 3)
is.na(x)
is.nan(x)
```

Data Frame

Data frames son usados para almacenar datos de manera tabular

- Son representados como un tipo de lista especial donde cada elemento de la lista tiene la misma longitud
- A diferencia de las matrices, los data frame pueden almacenar diferentes tipos de clases en cada columna
- Data frames tienen un atributo especial llamado `row.names`
- Data frame son usualmente creados usando `read.table()` o `read.csv()`
- Pueden ser convertidos a una matriz al usar `data.matrix()`

R

```
x <- data.frame(foo=1:4, bar=c(T, T, F, F))  
x  
nrow(x)  
ncol(x)
```


Names

Los objetos de R pueden tener names, lo que es muy útil para escribir código leíble y auto-descriptible

R

```
x <- 1:3
names(x)
names(x) <- c("foo", "bar", "north")
x
names(x)
```

Las listas pueden tener nombres

R

```
x <- list(a=1, b=2, c=3)
x
```

Las matrices también pueden tener nombres

R

```
m <- matrix(1:4, nrow=2, ncol=2)
dimnames(m) <- list(c("a", "b"), c("c", "d"))
m
```

Subsetting (subconjunto)

Hay un numero de operadores que pueden utilizarse para extraer una subserie de objetos R

- [siempre regresa un objeto de la misma clase que el original
- [[es usado para extraer elementos de una lista o de un data frame
- \$ es usado para extraer elementos de una lista o de el nombre de un data frame

R

```
x <- c("a", "b", "c", "c", "d", "a")
x[1]
x[2]
x[1:4]
x[x> "a"]
u <- x>"a"
u
x[u]
```

Subsetting a Matrix

Las matrices pueden ser filtradas en la forma común de índices (i,j)

R

```
x <- matrix(1:6, 2,3)
```

```
x
```

```
x[1,2]
```

```
x[2,1]
```

```
## Los índices se pueden omitir
```

```
x[1,]
```

```
x[,2]
```

```
## Por default, return a vector de longitud uno. Este comportamiento  
puede ser cambiando mediante drop=FALSE
```

```
x[1,2]
```

```
x[1,2, drop=FALSE]
```

Subsetting Lists

R

```
x <- list(foo=1:4, bar=0.6)
```

```
x
```

```
x[[1]]
```

```
x$bar
```

```
x[["bar"]]
```

```
x["bar"]
```

Subsetting List II

El operador `[[` puede ser usado con índices; y `$` solo puede ser usado con nombres literales

R

```
x <- list(foo=1:4, bar=0.6, baz="hello")
name <- "foo"
x[[name]]
x$name
```

R

```
## El [[ puede tomar una secuencia
x <- list(a=list(10,12,14), b=c(3.14,2.81))
x[[c(1,3)]]
x[[1]][[3]]
```

Partial Matching

Coincidencia parcial de un nombre con `[[` o `$`

R

```
x<- list(aardvark=1:5)
x$a
x[["a"]]
x[["a", exact=FALSE]]
```

Remove NA Values

Una tarea común es remover los valores perdidos (NAs)

R

```
x <- c(1,2,NA, 4, NA, 5)
bad <- is.na(x)
x[!bad]
```

R

```
x <- c(1,2,NA, 4, NA, 5)
y <- c("a", "b", NA, "d", NA, "f")
good <- complete.cases(x,y)
good
x[good]
y[good]
```


Remove NA Values

R

```
airquality[1:6, ]  
good <- complete.cases(airquality)  
airquality[good,][1:6,]
```

- Apartir del dataframe `airquality`, filtrar los datos para el mes de agosto
- Apartir del dataframe `airquality`, filtrar los dias 21
- Apartir del dataframe `airquality`, filtrar las temperaturas más alta e indicar el mes.

R