

Taller de R. Clase 01

Dr. Isaías Moreno Cruz

Objetivo

R es un software de estadística y visualización para análisis de datos. El presente Taller tiene como objetivo conocer las generalidades básicas de R. En el curso aprenderas a instalar R, así las distintas interfases de usuario (GUI) disponibles, usar listas, manejar data frames, escribir y leer datos, gráficar, crear funciones y más. R es un software multiplataforma que puede ser instalado en Windows, Linux y Mac.

- ① Antecedentes
- ② Motivación
- ③ Instalación
- ④ Introducción a R
 - Objetos, números, atributos
 - Matrices
 - Funciones cbind y rbind
 - Listas, factores
 - Valores perdidos
 - Data Frame
 - Remover valores NA

- 1 Operaciones vectoriales
- 2 Leer y escribir datos
- 3 Estructuras de control
- 4 Funciones
- 5 Loop function
- 6 Números aleatorios
- 7 Gráficas (plot, lattice, ggplot2)
- 8 Colores
- 9 Fechas y horas
- 10 Expresiones regulares
- 11 Reproducibilidad

- 1976: S es un lenguaje desarrollado por John Chambers y otros en Bell Labs.
- 1991: R creado en Nueva Zelanda por Ross Ihaka y Robert Gentleman
- 1995: Martin Mächler convence a Ross y Robert de usar la licencia GNU General Public para hacer R software libre
- 1997: El R Core Group se forma, R Core.
- 2000: La versión 1.0.0 es R liberado

- R base, Instalación
- Packages. lattice, ggplot2, dplyr, ...
- Editores: terminal, jupyter, R Studio, emacs.
- Stack Overflow

Ejemplo

motivación

Todo en R es un objeto. Tiene cinco clases basicas de objetos:

- character
- numeric
- integer
- complex
- logical(True/False)

R

```
x <- 4.2  
class(x)
```

Numbers

- Inf; e.g. $1/0$
- NaN; e.g. $0/0$

R

```
x <- 1/0  
y <- 0/0  
print(x)  
print(y)
```

Los objetos de R pueden tener los atributos:

- names
- dimensions
- class
- length

El caracter # indica un comentario.

R

```
# Input
```

```
x <- 1
```

```
print(x)
```

El [1] indica que x es un vector y su primer elemento es 1.

Operador :

Operador : usado para crear una secuencia

R

```
# Input  
x <- 1:20  
x
```

Función c()

La función `c()` puede usarse para crear vectores de objetos

R

```
x <- c(0.5, 0.6)
x <- c(TRUE, FALSE)
x <- c(T, F)
x <- c("a", "b", "c")
x <- 9:29
x <- c(1+0i, 2+4i)
```

Objetos mixtos

R

```
y <- c(1.7, "a")  
class(y)  
y
```

Coerción explícita. Funciones as.*

Los objetos pueden ser forzados explícitamente a una clase usando la función as.*

R

```
x <- 0:6  
class(x)  
as.numeric(x)  
as.logical(x)  
as.character(x)  
as.complex(x)
```

R

```
x <- c("a", "b", "c")  
as.numeric(x)
```


Las matrices son vectores con un atributo de dimension. La dimensión es un vector entero con longitud 2 (nrow, ncol).

R

```
m <- matrix(nrow=2, ncol=3)
m
dim(m)
attributes(m)
```

Matrices II

Las matrices son construidas por columnas.

R

```
m <- matrix(1:6, nrow=2, ncol=3)
```

m

Las matrices pueden ser creadas a partir de un vector al agregar el atributo de dimension.

R

```
m <- 1:10
```

m

```
dim(m) <- c(2,5)
```

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
```