Instituto de Salud Carlos III

Introducción a 

3. Manejo básico de datos

3.1. Introducción objetos en R

Instituto de Salud Carlos III

Índice

3.1. Introducción objetos en R

3.1.1. ¿Qué es un objeto?

- Casi todo en R es un objeto, incluyendo funciones y estructuras de datos.
- Para saber los objetos que tenemos en el **espacio de trabajo (WORKSPACE)** utilizaremos **ls().**
- Escribir el nombre de un objeto muestra su contenido: mean.
- Para guardar el contenido del espacio de trabajo se pueden utilizar las funciones: **save.image()** y **save(<objetos>,file="nombre.RData")**
- Para acceder a objetos de la carpeta de trabajo (o del camino que especifiquemos) se pueden adjuntar:

```
> attach("misdatos.RData")
> ls(pos=2) # segunda posición en la ‘ ‘search list’ ’
```

3.1.2. Tipos de objetos (1/2)

a. Objetos del lenguaje:

1.1. llamadas
1.2. expresiones
1.3. nombres

b. Expresiones: colecciones de expresiones correctas no evaluadas

c. Funciones:

-Constan de:

a)lista de argumentos
b)código
c)entorno

d. Sin objeto: NULL

3.1.2. Tipos de objetos (2/2)

e. Objetos para los datos:

Tipo de Objeto	Definición	Ejemplo	Observaciones
Vector	Colección ordenada de elementos del mismo tipo	x<-c(1,2,3); z<-c(TRUE,FALSE,TRUE); y<-c("Low","Low","Medium","High")	Factor== tipo de vector de datos cualitativos
Array	Generalización multidimensional del vector. Elementos del mismo tipo.	matrix(rnorm(20),ncol=5)	x
Data.frame	Igual que el array pero puede tener columnas de distintos tipo.	dades <- data.frame(ID=c("gen0", "genB", "genZ"),subj1 = c(10, 25, 33), subj2 = c(NA, 34, 15), oncogen = c(TRUE, TRUE, FALSE),loc = c(1,30, 125))	x
List	Una colección ordenada de objetos conocidos y sus componentes.	una.lista<-list(un.vector = 1:10, una.palabra = “hola”, una.matriz = matrix(rnorm(20), ncol = 5), lista2 = c(a=5, b = factor(c(“a”,”b”))))	x

Instituto de Salud Carlos III

- Creación
- Carga
- Atributos (nombre, longitud, …)
- Dos objetos especiales:
  - -Graficos
  - -Funciones

3.1.3. Creación

NOMBRE OBJETO <- CONTENIDO

Ejemplos creación objetos

- **Vector=** Colección ordenada elementos del mismo tipo.

  > x <- c(5, 6, 7); y <- c("a", "b", "Enrique")
  > z1 <- c(FALSE, TRUE, FALSE)

- **Array=** Generalización multidimensional de vector. Elementos del mismo tipo.

- **data frame=** Como array, pero permiten elementos (columnas) de distintos tipos. El objeto más habitual para manejo de datos procedentes de experimentos.

  > my.data.frame <- data.frame(ID = c("gen0", "genB","genZ"),
  subj1 = c(10, 25, 33), subj2 = c(NA, 34, 15),
  oncogen = c(TRUE, TRUE, FALSE),
  loc = c(1,30, 125))

Ejemplos creación objetos

- **Factor=** Un tipo de vector para datos categóricos.
- **List=** Un "vector generalizado". Cada lista está formada por componentes (que pueden ser otras listas), y cada componente puede ser de un tipo distinto. Son unos "contenedores generales".

```
> una.lista <- list(un.vector = 1:10, una.palabra = "Hola",
una.matriz = matrix(rnorm(20), ncol = 5),
otra.lista = c(a = 5, b = factor(c("a", "b"))))
```

- **Funciones= function()**

Cosas variadas

(1) En los scripts para poder escribir comentarios y que no se ejecuten se utiliza el comando:

#

•Ejemplo:

```
# Voy por aquí sumado
1+1  # esto da 2
```

(2) Si escribimos ; en la misma línea es como escribir una nueva línea de comando. Ejemplo:

```
> datos.estudio <- c(1,2,3); datos.estudio2<-c("A", "B", "C")
```

Instituto de Salud Carlos III

PRACTICAMOS ESTAS CREACIONES

3.1.4. Carga de objetos

- Cargar objetos en el workspace previamente creados:

  a) Consola: load( )

  b) Menú de la Interfaz:

“VER IMPORTACIÓN”

Los objetos que tenemos creados o cargados en la sesión

- **Ver contenido de un objeto:** introducimos el nombre del objeto en la línea de comandos

- **COMANDOS** para saber los objetos que hemos creado o cargado:

```
> ls()
> objects()
> objects(pattern="a*")
```

- Para borrar objetos concretos

```
> rm(objetos)
```

- Para borrar todos los objetos del entorno de trabajo:

```
> rm(list = ls())
```

- Asignación de los valores de un objeto a otro

3.1.5. Atributos de los objetos (1/3)

1. **Modo:** Tipo básico en un vector o array: lógico, entero, real, carácter,... mode

2. **Tipo:** de los vectores o arrays: double,... typeof

3. **Nombres:** etiquetas de los elementos individuales de un vector o lista: names

4. **Dimensiones:** de los arrays (alguna puede ser cero) o vectores: dim, length

5. **Dimnames:** nombres de las dimensiones de los arrays: dimnames

6. **Clase:** vector alfanumérico con la lista de las clases del objeto: class

7. **Otros:** atributos de una serie temporal, estructura data.frame: str()

8. **Cambios de formato**: as.numeric, as.factor, as.character

3.1.5. Atributos de los objetos (2/3)

Ejemplos comandos:

```
> x <- 1:15; length(x)
> y <- matrix(5, nrow = 3, ncol = 4); dim(y)
> is.vector(x); is.vector(y); is.array(x)
> as.factor(x); as.numeric(x); as.character(x), as.numeric(as.character(x)),
> x1 <- 1:5; x2 <- c(1, 2, 3, 4, 5); x3 <- "patata"
> typeof(x1); typeof(x2); typeof(x3)
> mode(x); mode(y); z <- c(TRUE, FALSE); mode(z)
> attributes(y)
> w <- list(a = 1:3, b = 5); attributes(w)
> y <- as.data.frame(y); attributes(y)
> f1 <- function(x) {return(2 * x)}
> attributes(f1); is.function(f1)
```

3.1.5. Atributos de los objetos (3/3)

Cambio del tipo de vector (variable):

```
> x <- 1:15;
```

```
is.factor(x); is.numeric(x); is.character(x)
```

```
as.factor(x); as.numeric(x); as.character(x), as.numeric(as.character(x))
```

3.1.6. Nombres para los objetos

- Los nombres válidos para un objeto son combinaciones de letras, números y el punto (“.”).
- Los nombres no pueden empezar con un número.
- R es “case-sensitive”: x != X.
- Hay nombres reservados (“function”, “if”, etc.).
- Otras consideraciones:
  -El uso del “.” es distinto del de C++.
  -Mejor evitar nombres que R usa (ej., “c”)
  -Las asignaciones se hacen con **"<-"** y se recomiendan los espacios.
  -El signo **"="** se reserva para los argumentos de las funciones.

Consideraciones

- Sustitución de nombres de objetos
- Asignación de los valores de un objeto a otro
- No pueden coexistir dos objetos con el mismo nombre (siempre prevalece el más moderno)

Instituto de Salud Carlos III

EJERCICIOS DEL 1 AL 7

EJERCICIOS
"ejercicios.manejo.basico.datos.r"

3.1.7. Gráficos

- R incluye muchas y variadas funciones para hacer gráficos.
- El sistema permite desde gráficos muy simples a figuras de calidad para incluir en artículos y libros.
- Sólo examinaremos la superficie. Más detalles en días sucesivos y en el libro *R Graphics* de Paul Murrell por ejemplo.
- También podemos ver un buen conjunto de ejemplos con **demo**(graphics).
- El comando **plot** es uno de los más utilizados para realizar gráficos.

3.1.8. Funciones

- R es un lenguaje que permite crear nuevas funciones.
- **Definición:** Una función se define con una asignación de la forma:

  >nombre <- **function(**arg1,arg2,...**)**{expresión}

- La expresión es una fórmula o grupo de fórmulas que utilizan los argumentos para calcular su valor.
- El valor de dicha expresión es el valor que proporciona R en su salida y éste puede ser un simple número, un vector, una gráfica, una lista o un mensaje.