Introducción al paquete R

Dr. Edgar Ramírez Galeano

El entorno de trabajo de R

R es un potente lenguaje orientado a objetos y destinado al análisis estadístico y la representación de datos. Se trata de software libre que permite su utilización libre y gratuitamente. Podemos destacar las siguientes componentes:

- Funciones matemáticas para el proceso de vectores y matrices.
- Una gran cantidad de funciones estadísticas integradas en el sistema.
- Una amplia variedad de librerías especializadas para estadística y análisis de datos.
- Funciones de proceso gráfico orientadas al análisis de datos.
- Un lenguaje de programación completo basado en el lenguaje S.

Como podemos obtener ayuda en R

Funciones de ayuda en R

Función	Acción
help.start()	Ayuda general
help(sum) o ?sum	Función de ayuda
help.search("sum")	Ayuda online
data()	Muestra los conjuntos de datos de ejemplo que hay disponibles

R como calculadora

R puede ser utilizado como una calculadora de modo interactivo.

1+1	Suma de dos números
2*3	Multiplicación de dos números
3/2	Division real de dos numeros
3 %/ %2	Division entera: se devuelve la parte entera solamente
2^3	Potenciación
abs(-1.5)	Valor absoluto de un número
sqrt(2)	Raíz cuadrada de 2
pi	Número pi

Ayuda sobre funciones matemáticas en R

help("Math")
help(.Arithmetic")
help("Trig")

Variables en R

En R empleamos el operador $\leftarrow$ para hacer asignaciones, y una variable se crea durante la ejecución, esto es, en el instante en el que la asignas valores.

Asignación de valores a una variable

$>$ suma $\leftarrow$ 1+1
Ver el contenido de la variable
$>$ suma
2

Cualquier asignación a una variable crea un "objeto" de R.

Vectores

Son matrices de una dimensión que solamente pueden contener valores homogéneos, ya sean numéricos, alfanuméricos o valores lógicos. Emplearemos la función c().

Ejemplos de vectores

Crea un vector numérico de 4 elementos
c(10,2,7,90)
10 2 7 90
Crea un vector lógico de 5 elementos
c(T,F,T,T,F)
TRUE FALSE TRUE TRUE FALSE
Crea un vector de 2 cadenas de caracteres
c("Pedro","Juan")
"PedroJuan"

La letra c significa çoncatenar".

Extracción de elementos de un vector

Especificar los índices de los elementos a extraer

```
> x<-c(10,20,1,5,8,90)
```

Extrae el elemento en la posición 4 del vector

```
> x[4]
5
```

Extrae los elementos 1, 3 y 5 del vector

```
> x[c(1,3,5)]
10 1 8
```

Especificar una condición lógica.

```
> x>10
FALSE TRUE FALSE FALSE FALSE TRUE
> x[x>10]
20 90
```

Creación de patrones

Funciones para crear vectores de modo automático: **from:to seq y rep.**

Ejemplos

```
> 1:4
1 2 3 4
> seq(1,4)
1 2 3 4
> seq(1,4,by=0.5)
1.0 1.5 2.0 2.5 3.0 3.5 4.0
> seq(1,4,length=6)
1.0 1.6 2.2 2.8 3.4 4.0
```

Matrices

Una matriz en R es un conjunto de objetos indizados por filas y columnas.
Sintaxis General:
matrix(data, nrow, ncol, byrow=F)

data	Datos que forman la matriz
nrow	Número de filas de la matriz
ncol	Número de columnas de la matriz
byrow	Los datos se colocan por filas o por columnas según se van leyendo.

Ejemplo

```
> matrix(1:6)
[,1]
[1,] 1
[2,] 2
[3,] 3
[4,] 4
[5,] 5
[6,] 6
> matrix(1:6,nrow = 2,ncol = 3)
[,1] [,2] [,3]
[1,] 1 3 5
[2,] 2 4 6
```

Extracción de elementos de un matriz

```
> x
[,1] [,2] [,3]
[1,] 1 3 5
[2,] 2 4 20
```

Extrae el elemento de la fila 2 columna 3

```
> x[2,3]
[1] 20
```

Extrae todos los elementos de la fila 1

```
> x[1,]
[1] 1 3 5
```

Extrae todos los elementos de la columna 2

```
> x[,2]
[1] 3 4
```

Algunas funciones sobre matrices

```
> x<-matrix(c(10,20,30,40),nrow=2)
[,1] [,2]
[1,] 10 30
[2,] 20 40
```

Número de elementos de x

```
> length(x)
4
```

Tipo de datos de la matriz x

```
> mode(x)
numeric
```

Dimensiones de la matriz x

```
> dim(x)
2 2
```

Se añade una columna a x

x<-cbind(x,c(100,300))

[,1] [,2] [,3]

[1,] 10 30 100

[2,] 20 40 300

Se añade una fila a x

> x<-rbind(x,c(400,500,900))

[,1] [,2] [,3]

[1,] 10 30 100

[2,] 20 40 300

[3,] 400 500 900

Operaciones con Matrices

OPERACIÓN	SINTAXIS
Adición	+
Sustracción	-
Multiplicación por un escalar	*
Producto de Matrices	%*%
Matriz traspuesta	t()
Inversa de la matriz A	solve(A)

Ejercicios Matrices

$$A = \begin{pmatrix} 2 & 5 & 0 \\ 7 & 3 & 8 \\ 3 & 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 5 & 4 & 1 \\ 8 & 13 & 0 \\ 9 & 10 & 2 \end{pmatrix}$$

Calcular en R:

1. $A + B$
2. $A \cdot B$
3. $B \cdot A$
4. $(A^T)^T$
5. $(A)^{-1}$

Listas

Las listas sirven para concatenar objetos donde cada uno puede tener una estructura distinta. Esto no ocurre, por ejemplo, en los arrays, donde todos los elementos deben ser del mismo tipo (todos números, o todos carácter digamos).
Para crear una lista vamos a utilizar la función **list** y pasamos como argumentos los elementos que queremos incluir.

Código en R

```
familia<-list(padre="Jose",madre="Maria",num.hijos=2,
nombre_hijos=c("sofia",ruben"))
```

Nombres de los objetos dentro de la lista:

```
> names(familia)
[1] padre madre num.hijos nombre_hijos
```

Para acceder a componentes concretos se usa el operador $ seguido del nombre de la componente de la lista, o bien el número de la componente entre corchetes dobles [[ ]]:

```
 > familia$num.hijos
[1] 2
> familia[[3]]
[1] 2
```

Ejercicio Listas

Crear la lista alumno con la siguiente información:

- Matricula
- Nombre
- Apellido Paterno
- Apellido Materno
- Edad
- Materias

Data Frames

Los data frames son estructuras de datos de dos dimensiones (rectangulares) que pueden contener datos de diferentes tipos, por lo tanto, son heterogéneas. Esta estructura de datos es la más usada para realizar análisis de datos y seguro te resultará familiar si has trabajado con otros paquetes estadísticos.
Podemos entender a los data frames como una versión más flexible de una matriz. Mientras que en una matriz todas las celdas deben contener datos del mismo tipo, los renglones de un data frame admiten datos de distintos tipos, pero sus columnas conservan la restricción de contener datos de un sólo tipo.
En términos generales, los renglones en un data frame representan casos, individuos u observaciones, mientras que las columnas representan atributos, rasgos o variables.

```
> alumnos<- data.frame(
“nombre“ = c("jose","maria","luis","Guadalupe"),
“Sexo“ = c("M","F","M","F"),
“Edad“ = c(22,22,25,32)
)
 > alumnos
nombre Sexo Edad
1 jose M 22
2 maria F 22
3 luis M 25
4 Guadalupe F 32
 > alumnos$Edad
[1] 22 22 25 32
```

Ejercicio DataFrame

Crear una tabla alumnos con la siguiente información

Matricula	Nombre	Sexo	Calificacion
56	ACOSTA CLAUDIO	M	8
65	ACOSTA RAMON	M	9.5
15	ALBISU MONICA	F	8.5
21	ALE MARIA	F	5.5
15	ALIBERTI RAUL	M	6.2
11	ALLEVATO ANA	F	9
17	ALONSO IVANA	F	9.8

Scripts

Los scripts son documentos de texto con la extensión de archivo .R, por ejemplo mi_script.R.

Estos archivos son iguales a cualquier documentos de texto, pero R los puede leer y ejecutar el código que contienen.

Aunque R permite el uso interactivo, es recomendable que guardes tu código en un archivo .R, de esta manera puedes usarlo después y compartirlo con otras personas. En realidad, en proyectos complejos, es posible que sean necesarios mútiples scripts para distintos fines.