

MED - Tema introductorio: R - Teoría

Este es un tema introductorio al lenguaje de programación R, no incluido en el material original.

Asume experiencia en algún lenguaje de programación imperativo, como C/C++, Java, C# o Python.

Introducción

R es un lenguaje de programación orientado a la computación estadística cuya primera versión pública data de 1993.

Es un lenguaje interpretado, lo que significa que el código escrito en este lenguaje no se ejecuta directamente por el procesador, sino que requiere que haya instalado en el sistema operativo un programa llamado *intérprete* que traduce el código en R al procesador durante cada ejecución.

Los *paquetes* son contenedores de código añaden funcionalidades y datos a los programas en R. Los *paquete base* son aquellos que se consideran básicos para ejecutar un programa R y se cargan automáticamente, análogas a las bibliotecas básicas en otros lenguajes. Opcionalmente, el desarrollador puede cargar otros paquetes adicionales ofrecidos oficialmente por R o por otros desarrolladores.

Para ejecutar un programa en R se necesita tener instalado en el sistema un *entorno de ejecución* que incluya tanto un intérprete de R como los paquetes oficiales.

El software del entorno de ejecución de R es libre y de código abierto, y por tanto está disponible de manera gratuita para su instalación y uso.

The Comprehensive R Archive Network, más conocida como *CRAN*, es una web que incluye recursos relacionados con R, incluyendo los binarios (que son los ficheros a instalar en nuestro equipo para poder ejecutar R) y la documentación oficial.

Instalación

La instalación de R en el sistema local varía dependiendo del sistema operativo empleado.

- Windows: descarga el instalador de la distribución básica **R-base** desde este enlace y sigue los pasos del instalador.
- macOS: descarga el instalador .pkg que se corresponda con tu arquitectura (**x86_64** o **arm64**) desde este enlace y sigue los pasos del instalador.
- Linux
 - Debian/Ubuntu: ejecuta **apt install r-base** desde una terminal.

Para confirmar que ha sido instalado correctamente en tu sistema abre un terminal y escribe:

```
R --version
```

La salida debería ser algo como:

```
R version 1.2.3 (YYYY-MM-DD) -- "A text"  
Copyright (C) YYYY The R Foundation for Statistical Computing
```

Entornos de desarrollo

Para generar un código R no necesitas un entorno de desarrollo (también llamado IDE) dedicado. Sería suficiente con crear el programa, por ejemplo, con un editor de textos estándar.

No obstante, emplear un IDE ofrece ventajas como resaltado de sintaxis, sugerencias de código o ejecución integrada en el entorno.

Algunos IDEs populares para R son:

- **RStudio**: IDE dedicado para R, libre y de código abierto.
- **Visual Studio Code**: IDE de uso general con extensiones disponibles para R.
- **JupyterLab**: para poder ejecutar los programas R y no solo escribirlos se requiere la instalación adicional del kernel de R **IRkernel**.

Para instalar **IRkernel** abre un terminal, ejecuta R y después introduce los siguientes comandos:

```
install.packages("devtools")  
devtools::install_github("IRkernel/IRkernel")  
IRkernel::installspec()
```

Debes reabrir JupyterLab para que los cambios sean efectivos.

Ejecución de código

Existen distintas maneras de ejecutar un programa de R.

- Desde línea de comando: emplea el comando **R** (en mayúscula) desde un terminal seguido del nombre del programa.
- Desde RStudio.
- Desde JupyterLab: ejecuta la celda con el código en R.

Documentación

La documentación oficial de R está disponible en <https://cran.r-project.org/manuals.html>.

La documentación destacada sería:

- **An Introduction to R**: introducción al lenguaje de tipo didáctico. Útil como primer contacto o repaso al lenguaje.

- **The R language definition:** descripción del lenguaje más extensa que la introducción. No es necesario leer para la asignatura.
- **The R Reference Index:** documentación detallada y extensa. Útil para consultas específicas sobre alguna función o paquete.

Existen varias versiones de la documentación dependiendo a qué versión se está refiriendo. Para el uso que se da en la asignatura, la versión empleada es R-release.

Cada uno de estos documentos se ofrece en formato HTML, PDF o EPUB para comodidad de los usuarios.

Conceptos básicos

Se introducen algunos conceptos básicos de R y su sintaxis.

Asignaciones

Las asignaciones de valores a una variable se pueden realizar mediante los operadores `=` y `<-`.

Funciones del sistema básicas

Se introducen las funciones básicas de carácter general y no directamente relacionado con matemáticas o estadística.

print

La función `print()` es la primera que probablemente aprendas y ejecutes. Sirve para mostrar en pantalla un texto, por ejemplo el típico programa de prueba “Hola mundo”.

```
print("Hello world!")
```

ls

La función `ls()` devuelve un vector de cadenas de caracteres (*strings*) con los nombres de las variables de memoria almacenadas en el sistema. Es una abreviatura de *list*.

rm

La función `rm()` sirve para borrar de memoria una variable. Es una abreviatura de *remove*.

`rm()` habitualmente espera como parámetro el nombre de una variable a borrar.

Se pueden borrar varias variables a la vez si se llama a `rm()` con el parámetro `list` asociado a vector de strings que contienen los nombres de variables a borrar.

Un comando habitual por el que empiezan todos los programas es:

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

Este comando borra de memoria todas las variables almacenadas en memoria. Sirve para resetear la memoria y evitar que variables empleadas anteriormente en otros programas persistan en el nuevo.

Funciones matemáticas básicas

runif

La función `runif()` sirve para generar un vector de números aleatorios. Los argumentos son:

- Número de elementos a generar aleatoriamente.
- Límite inferior (incluido) de los números a generar.
- Límite exterior (incluido) de los números a generar.

Ejemplo que genera un vector de 3 elementos con valores entre 0 y 1:

```
runif(3,0,1)
```

sqrt

La función `sqrt()` calcula la raíz cuadrada.