

Clase 2

Práctica 1

DCAO

2do cuatrimestre 2023

Repaso de la clase pasada ...

¿Cómo funciona R?

1. Le doy un programa/scrip en lenguaje R
2. Ejecuta un programa/una línea /una función
3. Resultado/Número/Gráfico

¿Cómo ejecutamos un programa/línea en Rstudio ?

- * Escribimos los comandos en la consola
- * Para correr un programa completo: `source('script.R')` o `ctrl+alt+R`
- * También se puede correr un conjunto de comandos seleccionados `ctrl+ENTER`.

Primeros pasos en R

“File” -> “New File” -> R Script **abro un nuevo scrip en blanco**

Primeros pasos en R

1. Es importante saber donde estamos parados cuando comenzamos un programa

`getwd()` El directorio de trabajo en que nos encontramos

`setwd("Mi directorio de trabajo")` fijar uno nuevo

Primeros pasos en R

2. Comenzar con el espacio limpio

Si quiero limpiar el espacio de trabajo (Environment)

`rm(list=ls())` borra todos los elementos guardados

Es una buena forma de comenzar un código

`rm(a,b)` borro a y b

Primeros pasos en R

The screenshot displays the RStudio IDE interface. The main window shows a script editor with the following R code:

```
1  
2 ### Este el programa de la clase 2 de Laboratorio de R ###  
3  
4 rm(list=ls())#Borro el environment  
5 setwd("/home/rocio/Documents/LaboR/") # Establezco mi lugar de trabajo  
6
```

The interface includes a menu bar (File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help) and a toolbar. The top right panel shows the Environment and History tabs, with the Global Environment visible. The bottom panel shows the Console, Terminal, and Jobs tabs, with the Console displaying the R prompt >.

Primeros pasos en R

3. Hacer secciones y poner títulos

“comento” el código y esa línea no va a ser leída como código en R

“Code” -> “Insert Section” Armo una sección dentro del programa

Primeros pasos en R

4. Guardar los códigos que hicimos en clase

Desde el Rstudio :

“File” -> “Save” **guardo el archivo con una extension .R**

Te lleva a guardarlo directamente a tu directorio de trabajo.

Guardamos el archivo con extensión .R con algún nombre que nos oriente sobre el contenido y que no tenga caracteres raros, ni espacios

Elijan la codificación UTF-8 para guardarlo. En Windows UTF-8 no suele ser la opción por defecto.

El comando de asignación pueden ser el signo $=$, pero es mejor usar $<-$

> a <- 2 asigno a la variable a el valor 2

> b <- -4 asigno a b el valor -4

> c <- a*b asigno a c el valor del resultado de a*b

> Texto <- "hola mundo"

OBJETOS DE R

R tiene cinco tipos básicos de objetos:

- * Numeric (número real con doble precisión)
- * Integer
- * Complex
- * Logical (TRUE/FALSE)
- * Character

NÚMERO DE MÁQUINA/MACHINE o PUNTO FLOTANTE

La memoria de los ordenadores es limitada

No se pueden almacenar números con precisión infinita

NÚMERO DE MÁQUINA/MACHINE o PUNTO FLOTANTE

La memoria de los ordenadores es limitada

No se pueden almacenar números con precisión infinita

Para entender esta limitación debemos saber que:

- * Las PCs usan el sistema binario. En notación científica base = 2 \rightarrow dígitos son 0 y 1 (bits)

e.g:

$$13,25 = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 + 1 * 2^{-1} + 1 * 2^{-2} = (1101,01)_2$$

- * Las computadoras representan los números con punto o coma flotante

NÚMERO DE MÁQUINA/MACHINE o PUNTO FLOTANTE

La memoria de los ordenadores es limitada

No se pueden almacenar números con precisión infinita

Para entender esta limitación debemos saber que:

- * Las PCs usan el sistema binario.
- * Las computadoras representan los números con punto o coma flotante

El número se descompone de la siguiente forma:

$$\begin{array}{c}
 \text{signo} \qquad \qquad \qquad \text{exponente} \\
 \underbrace{\hspace{1.5cm}} \qquad \qquad \underbrace{\hspace{1.5cm}} \\
 + \, 1.01110 \cdot 2^{-1101} \\
 \underbrace{\hspace{1.5cm}} \qquad \underbrace{\hspace{1.5cm}} \\
 \text{mantisa} \qquad \qquad \text{base}
 \end{array}$$

NÚMERO DE MÁQUINA/MACHINE o PUNTO FLOTANTE

Para entender esta limitación debemos saber que:

- * Las computadoras representan los números con punto o coma flotante

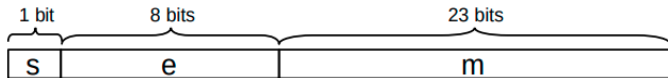
Las ventajas de escribir el número de esta forma:

- Puede representar números de órdenes de magnitud enormemente dispares (limitado por la longitud del exponente).
- Proporciona la misma precisión relativa para todos los órdenes (limitado por la longitud de la mantisa).
- Permite cálculos entre magnitudes: multiplicar un número muy grande y uno muy pequeño conserva la precisión de ambos en el resultado.

NÚMERO DE MÁQUINA/MACHINE o PUNTO FLOTANTE

Casi todo el hardware y lenguajes de programación utilizan números de punto flotante (float) en los mismos formatos binarios.

- precisión simple (32 bits o 4 bytes): 1 bit de signo, 8 de exponente, 23 de mantisa
- precisión doble (64 bits o 8 bytes): 1 bit de signo, 11 de exponente, 52 de mantisa



Veamos el Ejercicio 1 de la P1

Veamos el Ejercicio 1 de la P1

Machine double.eps

R: [1] 2.220446e-16, la distancia (epsilon) desde el 1 hasta el siguiente numero de precisión doble

Machine double.neg.eps

R: [1] 1.110223e-16, el epsilon tal que $1 - x \neq 1$

Machine double.xmin

R: [1] 2.225074e-308, el número mas pequeño que puede representar la maquina

Machine double.xmax

R: [1] 1.797693e+308, el número mas grande que puede representar la maquina

Machine double.base

R: [1] 2, La base para la representación de punto flotante: normalmente 2

Machine double.digits

R: [1] 53, la cantidad de dígitos en el significando del numero de punto flotante

Machine integer.max

R: 2147483647, entero máximo que es capaz de representar la maquina

Ejercicio para resolver en clase

Veamos algunos ejemplos en las compus ...