

Apuntes Práctica 2

Bucles

Son estructuras repetitivas que nos son útiles para hacer cálculos. Es ir dando pasos y después de cada paso revisar una condición o variable para seguir en el bucle o no.

-Bucles secuenciales: For

Es un comando que utilizamos para repetir un mismo proceso de un número a otro que nosotros decidimos. Además por defecto el bucle por cada repetición que hace aumenta en 1 la variable, hasta llegar al número que nosotros hemos especificado.

Es muy útil para calcular por ejemplo sumatorios y productorios. También para generar vectores y matrices.

En R se utiliza: `for("variable" in "valor inicial":"valor final":"incremento") { "proceso" }`

- 1º Se pone la palabra "for".
- 2º Dentro del paréntesis se especifica en primer lugar la variable del bucle (por ejemplo "i") y se determina de qué valor a que valor va a variar (por ejemplo de 1 hasta n).
- 3º Si quisiéramos especificar que el incremento va a ser diferente de 1 habrá que especificarlo después del valor final. **(es opcional)**
- 4º Se abren una llave y se escribe el proceso dentro.
- 5º Se cierra la llave.

Nota: Nosotros recomendamos que pongáis un poco desplazado a la derecha el proceso y que lo hagáis en la siguiente línea de código, y el cierre de la llave colocarlo a la altura del inicio del bucle for. Tal y como se ve en la siguiente línea.

```
for("variable" in "valor inicial":"valor final":"incremento"){  
    "proceso"  
}
```

Parece algo inútil y que nos hará perder tiempo, pero cuando empezamos con bucles anidados veremos que nos ayudará a no liarnos.

Representación gráfica de funciones

Definir una función: En R utilizamos el comando `function(x){`
"función"
`}`

Por ejemplo: `f=function(x){`
`sin(x^2)/8`
`}`

`f(8)=sin(64)/8`

Estas funciones se pueden representar en R, utilizando de nuevo el comando "plot"

- 1º Utilizamos `rm(list=ls(all=TRUE))` para borrar todas las variables (opcional)
- 2º Vamos a definir una función (simple en este caso) -> `f=function(x){`
`x^2/4`
`}`
- 3º Definimos los puntos que vamos a utilizar como soporte (nuestro eje x) como un vector `xx=c(2,5,7,10)` (por ejemplo)

- 4º Utilizamos el comando "plot": Aquí se definen todos los comandos que podemos introducir en orden en la función para que sea como queramos

- 1) eje x: es un vector con los puntos soporte -> `xx` (en nuestro caso)
- 2) eje y: es el vector de los puntos soporte en la función -> `f(xx)`
- 3) `col`: es el color de la representación. Hay que ponerlo entre comillas en inglés. Por ejemplo `col="red"`
- 4) `xlim`: es la escala del eje de las x. Hay que ajustarla para observar mejor la función. Es un vector. Por ejemplo `xlim=c(0,20)`
- 5) `ylim`: es la escala del eje de las y. Hay que ajustarla para observar mejor la función. Por ejemplo `ylim=c(0,50)`
- 6) `type`: es el tipo de función. Hay que ponerlo entre comillas. Por ejemplo `type="b"`
- 7) `xlab`: es el título del eje x del gráfico. Por ejemplo `xlab="abscisas"`
- 8) `ylab`: es el título del eje y del gráfico. Por ejemplo `ylab="ordenada"`
- 9) `pch`: es el tipo de punto en la gráfica. Por ejemplo `pch="19"`

- 5° El comando sería:
plot(xx,f(xx),col="red",xlim=c(0,20),ylim=c(0,50),type="b",xlab="abscisas",ylab="ordenadas",pch="19")
- 6° Vemos la función

```
rm(list=ls(all=TRUE))  
f=function(x){  
  x^2/4  
}  
xx=c(2,5,7,10)  
plot(xx,f(xx),col="red",xlim=c(0,20),ylim=c(0,50),type="b",xlab="abscisas",ylab="ordenadas",pch="19")
```

