

Programación II

Grado en Estadística y Empresa

Funciones

Autores

Dr. Fuensanta Medina Domínguez

Dr. María Isabel Sánchez Segura

Dr. Antonio de Amescua Seco



Funciones

- ▶ Sintaxis:

```
Nombre_func <- function(arg_1,arg_2,...,arg_n)
{
  sentencias a ejecutar
  #devolver valor
}
```

- ▶ Para devolver un valor por el medio de una expresión y terminar la ejecución de la función:

```
return(variable/expressión)
```

- ▶ Llamada a funciones:

```
Nombre_func (expr_1, expr_2,...,expr_n)
```

Función: Variables

▶ Parámetros Formales:

- ▶ Argumentos de la función
- ▶ Cualquier modificación que se haga sobre los mismos se pierde al salir de la función.

▶ Variables locales:

- ▶ Aparecen en la función al serles asignado algún valor
- ▶ Desaparecen al salir de la función

▶ Variable globales: **¡¡¡¡¡CUIDADO CON VARIABLES GLOBALES!!!!**

- ▶ Aparecen fuera o dentro del cuerpo de cualquier función
- ▶ Ámbito: todas las funciones que componen el programa, cualquier función puede acceder a dichas variables para leer y escribir en ellas

Funciones

▶ Ejemplo:

```
sumaNotas<-function(a,b)
{
  resultado=a+b
  return(resultado)
}
```

```
> sumaNotas(3,3)
[1] 6
```

Funciones

- ▶ Todo argumento que no tenga valor por defecto es obligatorio

```
sumaNotas<-function(a=2,b)
{
  resultado=a+b
  return(resultado)
}
```

```
> sumaNotas(,3)
[1] 5
```

Funciones

- ▶ Permite colocar los argumentos en el orden que se quiera siempre que se especifique a qué argumento corresponde

```
sumaNotas<-function(a,b)
{
  resultado=a/b
  return(resultado)
}
```

```
> sumaNotas(4,2)
[1] 2
```

```
> sumaNotas(2,4)
[1] 0.5
```

```
> sumaNotas(b=2,a=4)
[1] 2
```

Funciones: Notación Infija

- ▶ Funciones con dos parámetros, se puedan llamar con notación infija, es decir, poniendo el nombre de la función entre los dos argumentos

Sintaxis: “%Nombre_func%”

Ejemplo: Notación Infija - Solución

Ejemplo: Función que reciba dos números, multiplique el primer número por la suma de los dos números entre 100.

```
"%operar%"<-function(a,b)
{
  (a*(a+b)/100)
}
> 4%operar%6
[1] 0.4
```

Función recursiva

- ▶ Funciones que se llaman a sí mismas durante su propia ejecución
- ▶ Necesita sentencia de parada (salida no recursiva) o se tendrá una **función recursiva infinita**.
- ▶ Proceso: se llama a la misma función con el resultado anterior

EJEMPLO FUNCIÓN RECURSIVA FACTORIAL

```
mifactorial<-function(n)
{
  if(n==1)
    return(1) # salida no recursiva
  if(n>0)
    return(n*mifactorial(n-1))
  else
    return(NULL) #salida caso negativo
}
```

```
> mifactorial(3)
[1] 6
```

```
> mifactorial(4)
[1] 24
```