

Introducción: R

Giancarlo Sal y Rosas, Ph.D.

Departamento de Ciencias
Pontificia Universidad Católica del Perú
vsalyrosas@pucp.edu.pe

15 de marzo de 2017



Outline

- 1 Bases de datos
- 2 Funciones
- 3 Maximización



R

- Ambiente para estadística computacional y gráfica: Open source
- Versiones: Windows, MacOS, Linux, otros.
- Language de programación simple: Esta escrito en base a el lenguaje **C**
- Pagina principal: <https://cran.r-project.org/>



R: Calculadora

```
> 1 + 1
[1] 2
> 2 + 3 * 4
[1] 14
> 3^2
[1] 9
> exp(0.5)
[1] 1.648721
> sqrt(9)
[1] 3
> pi
[1] 3.141593
> 2*pi
[1] 6.283185
```



Variables

- Numérica
- Lógica: Verdadero o Falso
- String: Secuencia de caracteres
- El tipo es determinado cuando se crea la variable con el símbolo "<-"



Variables

```
> x <- 2  
> y <- 3  
> z <- 4  
> x + y*z  
[1] 14
```

```
> X + Y*Z  
Error: objeto 'X' no encontrado
```

```
> estado.civil <- "casado"  
> estado.civil  
[1] "casado"
```

```
> a <- 2 > 3  
> a  
[1] FALSE
```



Vectores

- Una ventaja de R es que de manera natural maneja vectores
- Existen al menos tres formas de crear vectores:
 - $c()$: Une valores usando comas
 - $rep(a, b)$: Repite **b** veces el valor **a**
 - $seq(a, b, by = c)$: Construye una secuencia de **a** hasta **b** con saltos de tamaño **c**
- Todos los operadores y funciones pre definidas se pueden aplicar directamente a estos



Vectores

```
> x <- c(1,2,3,4,5)
> x
[1] 1 2 3 4 5
> y <- rep(4,5)
> y
[1] 4 4 4 4 4
> z <- seq(4,5,0.1)
> z
[1] 4.0 4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5.0
> x^2
[1] 1 4 9 16 25
> x+y
[1] 5 6 7 8 9
> sqrt(x)
[1] 1.000000 1.414214 1.732051 2.000000 2.236068
```



Vectores

- Se puede acceder a los elementos de un vector con el operador "[]"

```
> z <- seq(4,5,0.2)
> z
[1] 4.0 4.2 4.4 4.6 4.8 5.0
> z[1]
[1] 4
> z[2:4]
[1] 4.2 4.4 4.6
> z[3] <- 20
> z
[1] 4.0 4.2 20.0 4.6 4.8 5.0
> z < 4.5
[1] TRUE TRUE FALSE FALSE FALSE FALSE
> z[-1]
[1] 4.2 20.0 4.6 4.8 5.0
```



Lectura

- Una de las funciones mas populares: `read.csv`"

```
dat <- read.csv(file.choose())
```

- La variable **dat** tiene estructura **data.frame**

```
> head(dat)
```

	ptid	t1	t2	event	arm
1	203000045	NA	NA	0	1
2	203000045	0	90	0	1
3	203000045	90	175	0	1
4	203000045	175	266	0	1
5	203000045	266	357	0	1
6	203000053	NA	NA	0	1

donde la primera fila son los nombres de las variables



Manipulación

```
> dat$t1[1:6]
[1] NA    0  90 175 266  NA
```

```
> dat$t2[1:6] + dat$t1[1:6]
[1] NA  90 265 441 623  NA
```

```
> dat[1,]
      ptid  t1  t2 event arm
1 203000045 NA  NA     0   1
```

```
> dat[1:4,1:3]
      ptid  t1  t2
1 203000045 NA  NA
2 203000045  0  90
3 203000045 90 175
4 203000045 175 266
```



Estructura

```
> fun1 <- function(x){  
+   x*x + x  
+ }  
>  
> fun1(3)  
[1] 12  
> fun1(c(1,2,3,4))  
[1] 2 6 12 20
```

- El nombre de la función **fun1**
- La variable a ser evaluada es **x** que puede ser un real o un vector



Estructura

```
> fun3 <- function(a,b){  
+   aux <- sum(a)  
+   z   <- b + aux  
+   return(z)  
+ }  
>  
> x <- seq(0,10,1)  
> y <- 11  
> h <- c(1,2,3)  
>  
> fun3(x,y)  
[1] 66  
  
> fun3(x,h)  
[1] 56 57 58
```

- La entrada son dos valores (números o vectores)



Maximización

- Una parte crucial en estadística es maximizar una función
- En **R** existen varias funciones pre-establecidas para hacer esto
 - **optimize**: Para funciones de una variable
 - **nlm**, **nlminb** y **optim**: Para funciones de mas de una variable



Optimize

- Sintaxis:

optimize(f, interval, ..., lower =, upper =, maximum = FALSE)

- *f*: La función a optimizar
- *interval*: el intervalo sobre el cual se va a optimizar *f*
- *lower* y *upper*: Se puede definir en lugar de *interval*
- *maximum*: Si deseamos maximizar (FALSE) o minimizar (TRUE)



Optimize

- Deseamos minimizar la función en $[0, 1]$

$$f(x) = \left(x - \frac{1}{3}\right)^2$$

- Usando R

```
> fun <- function (x, a) (x - a)^2  
> xmin <- optimize(fun, c(0, 1), a = 1/3)  
> xmin  
$minimum  
[1] 0.3333333  
  
$objective  
[1] 0
```



Optim

- Sintaxis:

optim(par, fn, gr = NULL, ..., method =, lower =, upper =, ...)

- *par*: Los parámetros a ser optimizados
- *fn*: La función a optimizar
- *gr*: La gradiente (opcional)
- *method*: El método (hay cuatro)
- Varias opciones mas...



Optimize

- Deseamos minimizar la función

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- Es sencillo corroborar a mano que la solución es

$$x_1 = x_2 = 1$$



Optimize

```
> fr <- function(x) {  
+   x1 <- x[1]  
+   x2 <- x[2]  
+   100 * (x2 - x1 * x1)^2 + (1 - x1)^2  
+ }  
>  
> res <- optim(c(-1.2,1), fr)  
> res  
$par  
[1] 1.000260 1.000506  
  
$value  
[1] 8.825241e-08  
  
$convergence  
[1] 0
```

