

>> **Números con exponentes**

> 1.2e3

> 1.2e-2

> 3.9+4.5i

>> Módulo y Cociente Entero

> 119 %/% 13

> 119 %% 13

> 9 %% 2

> 8 %% 2

> 15421 %% 7 == 0

>> **Redondeo**

>> floor(5.7)

>> ceiling(5.7)

>> rounded_i-function(x) floor($x+0.5$)

>> rounded(5.7)

>> rounded(5.4)

C2

>> Infinito y cosas que no son números (NaN)

Inf

-Inf:

$3/0$

$-12/0$

$\exp(-\text{Inf})$

$0/\text{Inf}$

$(0 : 3)^{\text{Inf}}$

NaN

C2

0/0

Inf-Inf

Inf/Inf

NA

is.finite(10)

is.infinite(10)

is.infinite(Inf)

>> **Missing values NA**

```
xj-c(1:8,NA)
```

```
mean(x)
```

```
mean(x,na.rm=T)
```

```
is.na(x)
```

```
vmvj-c(1:6,NA,NA,9:12)
```

```
vmv
```

```
seq(along=vmv)[is.na(vmv)]
```

```
which(is.na(vmv))
```

```
vmv[is.na(vmv)] <- 0  
vmv
```

```
vmv <- c(1:6,NA,NA,9:12)  
ifelse(is.na(vmv),0,vmv)
```

>> **Operadores**

+ - * / %% (exponente) aritméticos

> >= < <= == != relacionales

! & | logicos

modelo de formula

< - - > asignamiento

\$ indexamiento de listas

: crea secuencias

>> **Creando Vectores**

$y < -4.3$

$z < -y[-1]$

$\text{length}(z)$

$y < -10:16$

$y < -c(10, 11, 12, 13, 14, 15, 16)$

C2

```
y <- scan()
```

```
1: 10
```

```
2: 11
```

```
3: 12
```

```
4: 13
```

```
5: 14
```

```
6: 15
```

```
7: 16
```

```
8:
```

```
Read 7 items
```

C2

$A < -1:10$

$B < -c(2,4,8)$

$A*B$

[1] 2 8 24 8 20 48 14 32 72 20

>> **Nombrando los elementos del Vector**

```
counts <- c(25,12,7,4,6,2,1,0,2)
```

```
[1] 25 12 7 4 6 2 1 0 2
```

```
names(counts) <- 0:8
```

```
counts
```

```
0 1 2 3 4 5 6 7 8 25 12 7 4 6 2 1 0 2
```

```
(st <- table(rpois(2000,2.3)))
```

```
0 1 2 3 4 5 6 7 8 9
```

```
205 455 510 431 233 102 43 13 7 1
```

```
as.vector(st)
```

```
[1] 205 455 510 431 233 102 43 13 7 1
```

>> **Funciones Vectoriales**

max(x) - maximum value in x

min(x) - minimum value in x

sum(x) - total of all the values in x

mean(x) - arithmetic average of the values in x

median(x) - median value in x

range(x) - vector of min x and max x

var(x) - sample variance of x

cor(x,y) - correlation between vectors x and y

sort(x) - a sorted version of x

rank(x) - vector of the ranks of the values in x

order(x) - an integer vector containing the permutation to sort x into ascending order

quantile(x) - vector containing the minimum, lower quartile, median, upper quartile, and maximum of x

cumsum(x) - vector containing the sum of all of the elements up to that point

cumprod(x) - vector containing the product of all of the elements up to that point

cummax(x) - vector of non-decreasing numbers which are the cumulative maxima of the values in x up to that point

cummin(x) - vector of non-increasing numbers which are the cumulative minima of the values in x up to that point

colMeans(x) - column means of dataframe or matrix x

colSums(x) - column totals of dataframe or matrix x

rowMeans(x) - row means of dataframe or matrix x

rowSums(x) - row totals of dataframe or matrix x

C2

>> Trabajando con vectores y Subíndices Logicos

```
x < - 0:10
```

```
sum(x)
```

```
[1] 55
```

```
sum(x < 5)
```

```
[1] 5
```

```
sum(x[x<5])
```

```
[1] 10
```

```
x < 5
```

```
[1] TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE
```

```
[10] FALSE FALSE
```

```
1*(x < 5)
```

```
[1] 1 1 1 1 1 0 0 0 0 0 0
```

C2

```
x*(x < 5)
```

```
[1] 0 1 2 3 4 0 0 0 0 0 0
```

```
sum(x*(x < 5))
```

```
yj-c(8,3,5,7,6,6,8,9,2,3,9,4,10,4,11)
```

```
sort(y)
```

```
[1] 2 3 3 4 4 5 6 6 7 8 8 9 9 10 11
```

```
rev(sort(y))
```

```
[1] 11 10 9 9 8 8 7 6 6 5 4 4 3 3 2
```

```
rev(sort(y))[2]
```

```
[1] 10
```

```
rev(sort(y))[1:3]
```

```
[1] 11 10 9
```


C2

>> Direcccionamiento con Vectores

```
y  
[1] 8 3 5 7 6 6 8 9 2 3 9 4 10 4 11
```

```
which(y > 5)  
[1] 1 4 5 6 7 8 11 13 15
```

```
y[y > 5]  
[1] 8 7 6 6 8 9 9 10 11
```

```
length(y)  
[1] 15
```

```
length(y[y > 5])  
[1] 9
```

```
y <- rnorm(1000, 100, 10)
```

>> **Encontrando los valores más cercanos**

```
which(abs(xv-108)==min(abs(xv-108)))  
[1] 332
```

```
xv[332]  
[1] 108.0076
```

```
closest <- function(xv,sv) {  
  xv[which(abs(xv-sv)==min(abs(xv-sv)))] }  
}
```

```
closest(xv,108)  
[1] 108.0076
```

>> Aritmética Logica

$x < -0:6$

$x < 4$

[1] TRUE TRUE TRUE TRUE FALSE FALSE FALSE

$\text{all}(x > 0)$

[1] FALSE

$\text{any}(x < 0)$

[1] FALSE

$\text{sum}(x < 4)$

[1] 4

C2

>> Repeticiones

```
rep(9,5)  
[1] 9 9 9 9 9
```

```
rep(1:4, 2)  
[1] 1 2 3 4 1 2 3 4
```

```
rep(1:4, each = 2)  
[1] 1 1 2 2 3 3 4 4
```

```
rep(1:4, each = 2, times = 3)  
[1] 1 1 2 2 3 3 4 4 1 1 2 2 3 3 4 4 1 1 2 2 3 3 4 4
```

```
rep(1:4,1:4)  
[1] 1 2 2 3 3 3 4 4 4 4
```

C2

>> Generando secuencias de números

10:18

[1] 10 11 12 13 14 15 16 17 18

18:10

[1] 18 17 16 15 14 13 12 11 10

-0.5:8.5

[1] -0.5 0.5 1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5

seq(0,1.5,0.2)

[1] 0.0 0.2 0.4 0.6 0.8 1.0 1.2 1.4

seq(1.5,0,-0.2)

[1] 1.5 1.3 1.1 0.9 0.7 0.5 0.3 0.1

```
x.values <- seq(min(x),max(x),(max(x)-min(x))/100)
x <- rnorm(18,10,2)
seq(88,50,along=x)
```

```
sequence(5)
[1] 1 2 3 4 5
```

```
sequence(5:1)
[1] 1 2 3 4 5 1 2 3 4 1 2 3 1 2 1
```

```
sequence(c(5,2,4))
[1] 1 2 3 4 5 1 2 1 2 3 4
```

>> Sorting, Ranking and Ordering

```
houses <- read.table("c:
temp
houses.txt",header=T)
attach(houses)
names(houses)
[1] "Location" "Price"

ranks <- rank(Price)
sorted <- sort(Price)
ordered <- order(Price)
view <- data.frame(Price,ranks,sorted,ordered)
view
```

C2

```
y  
[1] 8 3 5 7 6 6 8 9 2 3 9 4 10 4 11  
sample(y)  
[1] 8 8 9 9 2 10 6 7 3 11 5 4 6 3 4  
sample(y)  
[1] 9 3 9 8 8 6 5 11 4 6 4 7 3 2 10  
sample(y,5)  
[1] 9 4 10 8 11  
sample(y,5)  
[1] 9 3 4 2 8  
sample(y,replace=T)  
[1] 9 6 11 2 9 4 6 8 8 4 4 4 3 9 3
```



```
p j- c(1, 2, 3, 4, 5, 5, 4, 3, 2, 1)
xj-1:10
sapply(1:5,function(i) sample(x,4,prob=p))
```

C2

```
>> Matrices
X <- matrix(c(1,0,0,0,1,0,0,0,1),nrow=3)
X
class(X)
[1] "matrix"
attributes(X)
$dim
[1] 3 3
vector <- c(1,2,3,4,4,3,2,1)
V <- matrix(vector,byrow=T,nrow=2)
V
dim(vector) <- c(4,2)
is.matrix(vector)
[1] TRUE
```

>> Nombrando filas y columnas en matrices

```
X <- matrix(rpois(20,1.5),nrow=4)
```

```
X
```

```
rownames(X) <- rownames(X,do.NULL=FALSE,prefix="Trial.")
```

```
X
```

```
drug.namesj-c("aspirin", "paracetamol", "nurofen", "hedex", "placebo")
```

```
colnames(X) <- drug.names
```

```
X
```

```
dimnames(X) <- list(NULL,paste("drug.",1:5,sep=""))
```

```
X
```


