

# 5Vectores\_cheatsheet\_extendido.R

moka

2023-04-20

```
# Autor: Monika Avila Marquez, Ph.D.
# Referencia: Basado en R Programming Fundamentals, StanfordOnline XDFS112

# Definir directorio de trabajo
midirectorio<-setwd("~/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/5Vectores")
midirectorio

## [1] "/Users/moka/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/5Vectores"

# Crear un vector con c
vector1<-c(1,0,2,3)
# Consulte la clase
class(vector1)

## [1] "numeric"

# Comprobar si es un vector
is.vector(vector1)

## [1] TRUE

# Unir vectores
vector15vector1<-c(vector1,5,vector1)
# Obtener la longitud del vector
length(vector15vector1)

## [1] 9

# Ver que contiene el vector
str(vector15vector1)

##  num [1:9] 1 0 2 3 5 1 0 2 3

# Operaciones con vectores
# Suma
vector1+10

## [1] 11 10 12 13

# Potenciacion
vector1^2

## [1] 1 0 4 9

# Cuidado con pedir correctamente sumas matemáticas de vectores
vector2<-c(10,1,20,30)
vector1+vector2

## [1] 11 1 22 33
```

```

# Puedes equivocarte al sumar vectores no son sumables!
vector1+c(1,2) # ¡No es matemáticamente correcto!

## [1] 2 2 3 5
# Multiplicación con escalar
vector1*11

## [1] 11 0 22 33
# Multiplicación puntual con otro vector
vector1*vector2

## [1] 10 0 40 90
vector1

## [1] 1 0 2 3
vector2

## [1] 10 1 20 30
# Raíz cuadrada

# Indexación de vectores: se hace usando corchetes
# Tomar subconjuntos de vectores
# Acceder al primer elemento
vector1[1]

## [1] 1
# Acceder al máximo
# Obtener el índice del valor máximo
vector1

## [1] 1 0 2 3
which.max(vector1)

## [1] 4
vector1[which.max(vector1)]

## [1] 3
# Acceso al mínimo
# Obtener el índice del valor mínimo
vector1

## [1] 1 0 2 3
which.min(vector1)

## [1] 2
vector1[which.min(vector1)]

## [1] 0
# Nota: para las funciones vector redondo, vector cuadrado es para acceder a los elementos.
# Si utilizas vector redondo en su lugar, te dará el error de que la función vector1 no se encuentra
# Incorrecto vector1(1)
# Indexar elementos diferentes
vector1

```

```
## [1] 1 0 2 3
vector1[c(1,3)]

## [1] 1 2
vector1

## [1] 1 0 2 3
vector1[1:3]

## [1] 1 0 2
# Seq permite crear un índice con igual espacio entre elementos
seq(1,5,by=3)

## [1] 1 4
vector1[seq(1,5,by=3)]

## [1] 1 3
# Obtener un vector de booleanos
vector1>0

## [1] TRUE FALSE TRUE TRUE
# Obtiene el subvector cuyos elementos son mayores que 0
vector1[vector1>0]

## [1] 1 2 3
# Elimina un elemento, por ejemplo el elemento 2
vector1[-2]

## [1] 1 2 3
# Muestreo aleatorio sin reemplazamiento
# Obtener una muestra igual a 2
set.seed(1)
a=round(runif(2,1,4))
a

## [1] 2 2
vector1[a]

## [1] 0 0
# Utilizar la función sample
sample(vector1,size=2,replace=FALSE)

## [1] 2 1
sample(vector1,size=2,replace=TRUE)

## [1] 0 1
# Esta función permite crear permutaciones del vector original
sample(vector1,size=4,replace=FALSE)

## [1] 2 3 0 1
vector1
```

```
## [1] 1 0 2 3
```

```
# No quitar el comentario de la linea inferior. Solamente copiar en la consola para que ejecute  
#rmarkdown::render("5Vectores_cheatsheet_extendido.R",c("pdf_document","html_document"))
```