

INTRODUCCIÓN A LA PROGRAMACIÓN EN

R

OBJETIVO

Conocer los conceptos básicos del lenguaje R que nos permitan manipular objetos para realizar un primer análisis exploratorio de datos.

INTRODUCCIÓN
OBJETOS EN R
VECTORES
MATRICES
DATA FRAME
LISTAS
FACTORES
FECHAS Y TIEMPO
DATOS
GRÁFICOS
FUNCIONES

INTRODUCCIÓN AL LENGUAJE DE PROGRAMACIÓN R



ROSS IHAKA



Versión Libre

S-Plus

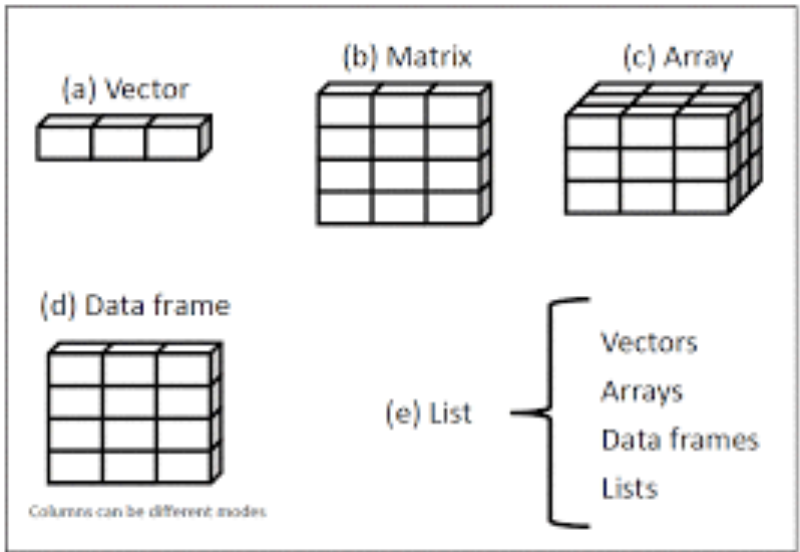


ROBERT GENTLEMAN

CARACTERÍSTICAS PRINCIPALES

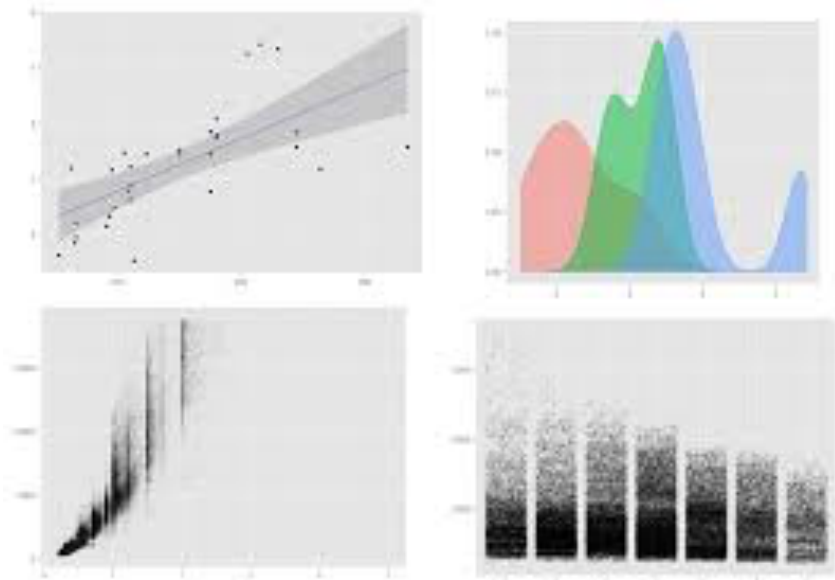
SINTAXIS Y SEMÁNTICA SIMILAR A S

ORIENTADO A OBJETOS

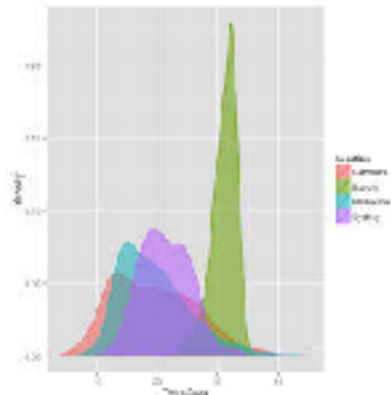


COMUNIDAD DE R

<http://cran.r-project.org/>



GRÁFICOS



SOFTWARE LIBRE

I ♥
SOFTWARE
LIBRE

UTILIZA PAQUETES



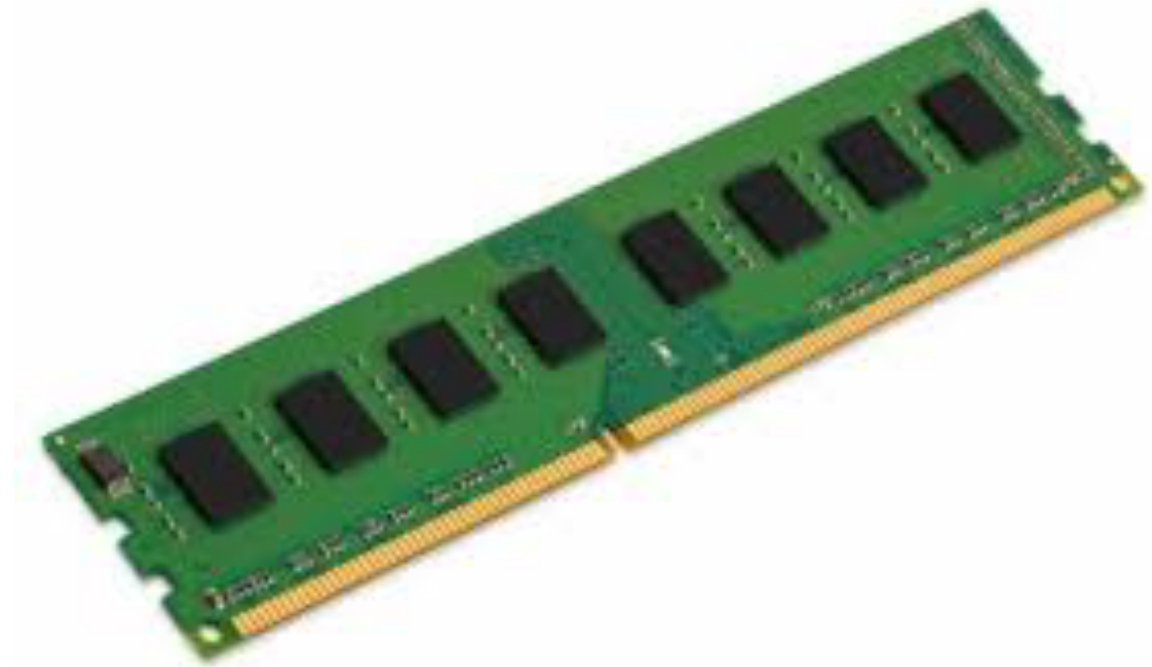
CORRE EN CUALQUIER SISTEMA OPERATIVO



SYNERGY
Vision

- Lenguaje de programación en líneas de comandos

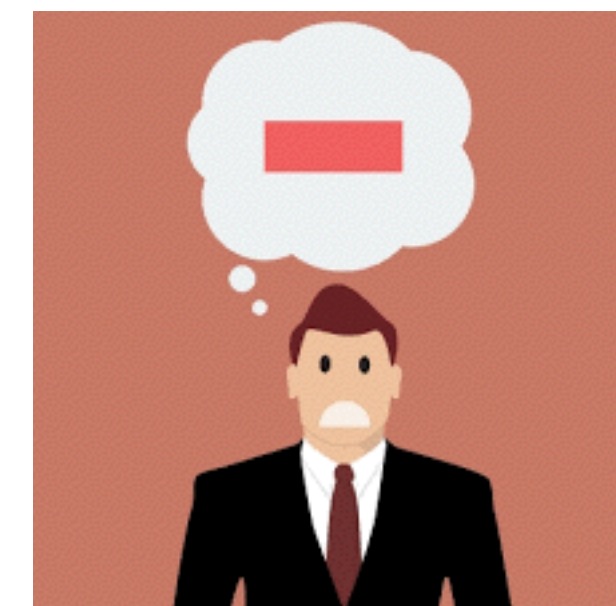
- Datos en memoria RAM



- Funciones basadas en usuarios



- No es ideal para usarlo con todo



Paquetes del CRAN

Paquetes Recomendados

R Base Core



[Home]

Download

CRAN

R Project

About R

Logo

Contributors

What's New?

Reporting Bugs

Development Site

Conferences

Search

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred [CRAN mirror](#).

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- [R version 3.5.1 \(Feather Spray\)](#) has been released on 2018-07-02.
- The R Foundation has been awarded the Personality/Organization of the year 2018 award by the professional association of German market and social researchers.
- [R version 3.5.0 \(Joy in Playing\)](#) has been released on 2018-04-23.

INSTALACIÓN DE PAQUETES

```
> install.packages("DT")
also installing the dependencies 'htmlwidgets', 'crosstalk'

probando la URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/htmlwidgets_1.2.tgz'
Content type 'application/x-gzip' length 761045 bytes (743 KB)
=====
downloaded 743 KB

probando la URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/crosstalk_1.0.0.tgz'
Content type 'application/x-gzip' length 595828 bytes (581 KB)
=====
downloaded 581 KB

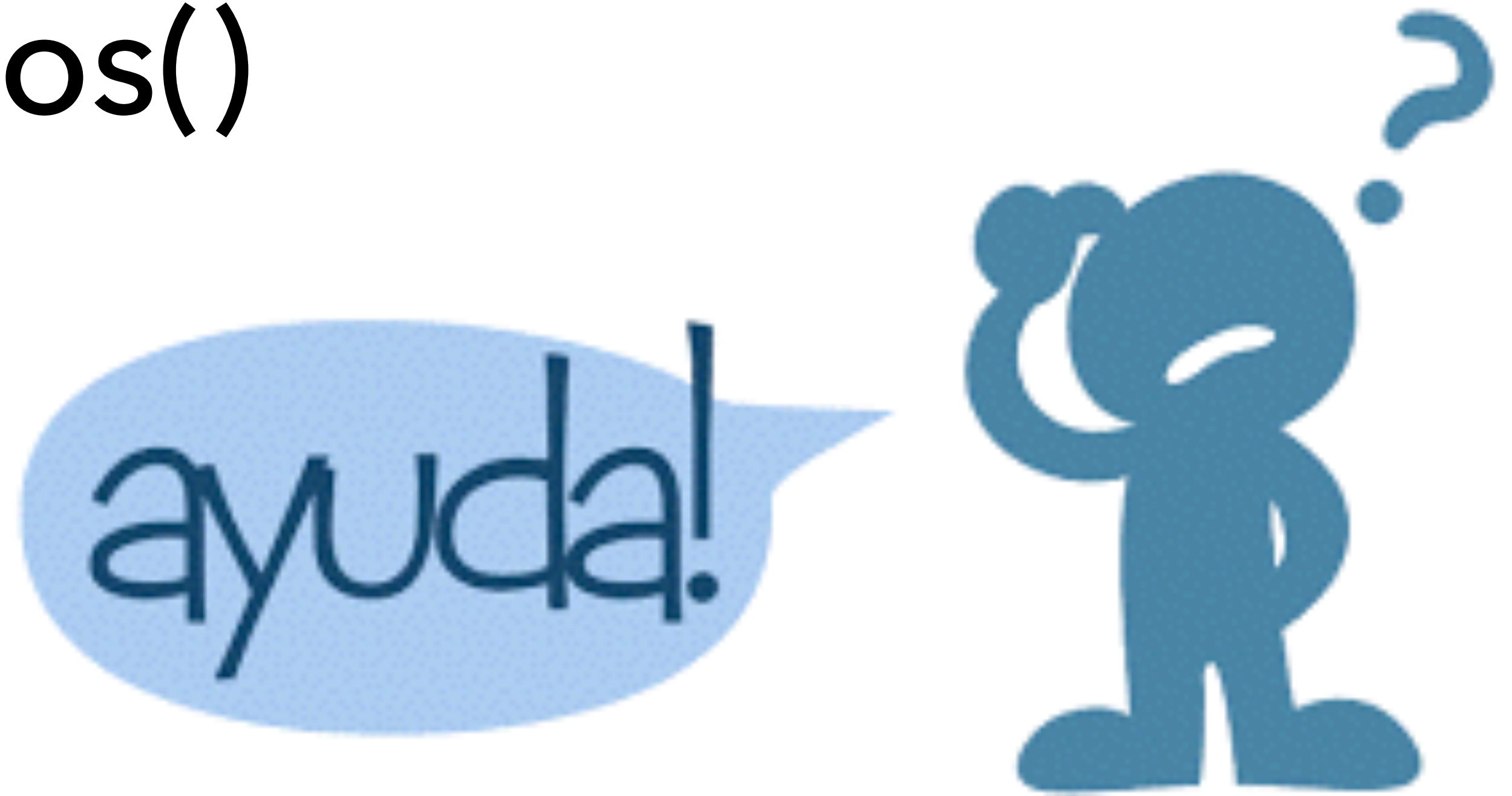
probando la URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/DT_0.4.tgz'
Content type 'application/x-gzip' length 892567 bytes (871 KB)
=====
downloaded 871 KB

The downloaded binary packages are in
  /var/folders/4r/d594lw55ldf494fqfxt7y180000gp/T//Rtmp1ZxfrB/downloaded_packages
> library("DT")
```

▶ help() ?

▶ apropos()

▶ example()





Ambiente de desarrollo integrado o
entorno de desarrollo interactivo
(IDE integrated development environment)

<https://www.rstudio.com/>

OBJETOS EN R

Entidades que combinan estado (atributo), comportamiento (método) e identidad; esto quiere decir, un objeto estará compuesto por datos también llamados atributos que tendrán valores concretos, con estos objetos se realizarán algunos procedimiento o método.

- ▶ Datos Numéricos
- ▶ Datos Categóricos
- ▶ Datos Lógicos
- ▶ Datos de Fecha y Hora
- ▶ Valores Perdidos, NA
- ▶ Inf
- ▶ NaN
- ▶ NULL

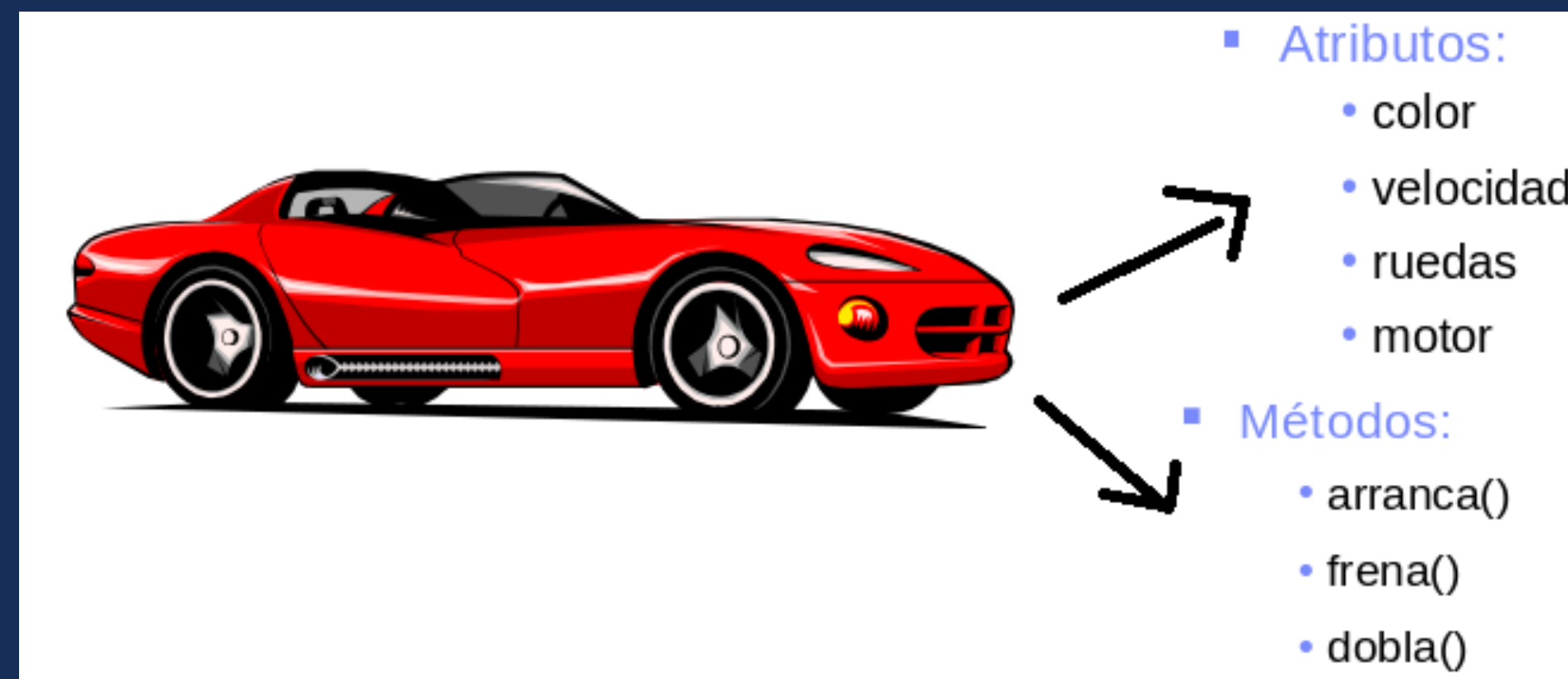


- Nombres (**names**)
- Dimensiones (**dimensions**)
- Dimnames (**dimnames**)
- Modo (**mode**)
- Tipo (**typeof**)
- Clase (**class**)

ATRIBUTOS INTRÍNSECOS

1. Tipo - **mode()**

2. Longitud - **length()**



- Combinaciones de letras números y caracteres especiales.
- Sensible a las mayúsculas.
- Colocar nombres distintos a funciones creadas.

Operador Asignación 

Entre las funciones más usadas con este lenguaje de programación tenemos:

- ▶ `ls()`
- ▶ `is.algo()` -> `is.vector()`, `is.list()`, `is.matrix()` ...
- ▶ `boxplot()`
- ▶ `ls.str()`
- ▶ `mean()`
- ▶ `sd()`
- ▶ `rm()`
- ▶ `median()`
- ▶ `var()`
- ▶ `paste()`
- ▶ `table()`
- ▶ `Summary()`
- ▶ `abs()`
- ▶ `range()`
- ▶ `apply()`
- ▶ `round()`
- ▶ `quantile()`
- ▶ `sapply()`
- ▶ `sort()`
- ▶ `IQR()`

► Operaciones aritméticas

							log()		sin()	asin()
+	-	*	sqrt()	^	%/%	%%	logb()	exp()	cos()	acos()
							log2()		tan()	atan()

► Operadores Comparativos

< > <= >= == !=

► Operadores Lógicos

& && | ||

VECTORES

CREACIÓN DE VECTORES

`c(2,4,6)`

2 4 6

Elementos de un vector

`2:6`

2 3 4 5 6

Sucesión de números enteros

`seq(2,3,by=0.5)`

2.0 2.5 3.0

Sucesión con un incremento

`rep(1,times=3)`

1 1 1

Repetir sucesiones

`sequence(nvec=3)`

1 2 3

Sucesión que inicia en 1

`gl(n=2,k=2)`

1 1 2 2

Series regulares de factores

SELECCIÓN DE ELEMENTOS Y OPERACIONES

`x <- c(2 , 1, 3 , -1, 10 , 0 , 0 , 1)`

`x[2]` 1 Extraer el segundo elemento

`x[-4]` 2 , 1, 3, 10 , 0 , 0 , 1 Eliminar el cuarto elemento

`x[c(1,3,5)]` 2310 Seleccionar varias posiciones

`x>2` FALSE FALSE TRUE FALSE
TRUE FALSE FALSE FALSE Vectores lógicos

`x+y` `x<-c(5,1,5); y<-c(2,1,3)`
7 2 8 Suma entre Vectores

`sum(x)` 11 Sumar los elementos de un vector

`x*y` 10 1 15 Otras operaciones

MATRICES

CREACIÓN Y SELECCIÓN DE ELEMENTOS

matrix(data,nrow,ncol,byrow=TRUE)

rownames()	Colocar nombres en las filas
colnames()	Colocar nombres en columnas
dim() nrow() ncol()	Dimensiones de la matriz
M[1,2]	Selección del elemento en la posición 1,2 de la matriz
M[1,]	Seleccionar una fila
M[,2]	Seleccionar una columna
M[1:2,2:3]	Selección de submatrices
M[-1,] M[,-1]	Eliminar fila o columna
M[1,2]<-4	Modificar un elemento

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

OPERACIONES CON MATRICES

$X1 + X2$ Suma de elementos entre matrices

$X1 + 2$ Sumar un Escalar

$X1 * X2$ Multiplicar elementos entre matrices

$x1 * 2$ Multiplicar por un escalar

$X1 \%*\% X2$ Multiplicación de matrices

`rowSums()` Sumar filas

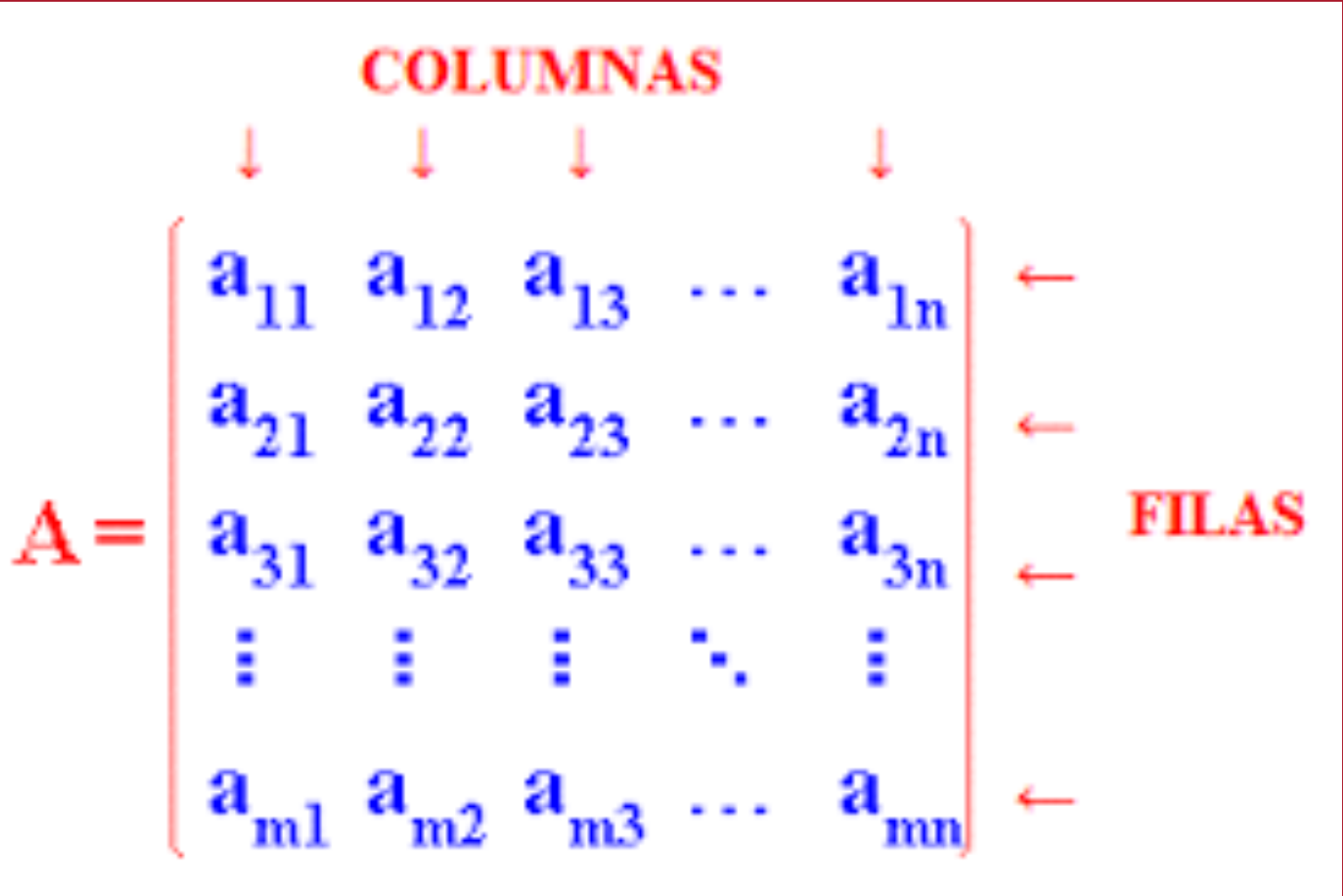
`colSums()` Sumar columnas

`rbin()` Agregar una fila

`cbin()` Agregar una columna

`rowMeans()` Promedio de filas

`colMeans()` Promedio de columnas



DATA FRAME

CREACIÓN DE UN DATA FRAME

```
data.frame( x = 1:3, y = c("a","b","c") )
```

colnames()

Colocar nombres a las columnas

rownames()

Colocar nombres a las filas

nrow()

ncol()

Número de filas y columnas

	state	color	food	age	height	score
Jane	NY	blue	Steak	30	165	4.6
Niko	TX	green	Lamb	2	70	8.3
Aaron	FL	red	Mango	12	120	9
Penelope	AL	white	Apple	4	80	3.3
Dean	AK	gray	Cheese	32	180	1.8
Christina	TX	black	Melon	33	172	9.5
Cornelia	TX	red	Beans	69	150	2.2

SELECCIÓN DE ELEMENTOS

`data_frame[i , j]`

Selección de elementos

`data_frame[i ,]`

Selección de una fila

`data_frame[, j]`

Selección de variable o columna

`data_frame$VARIABLE`

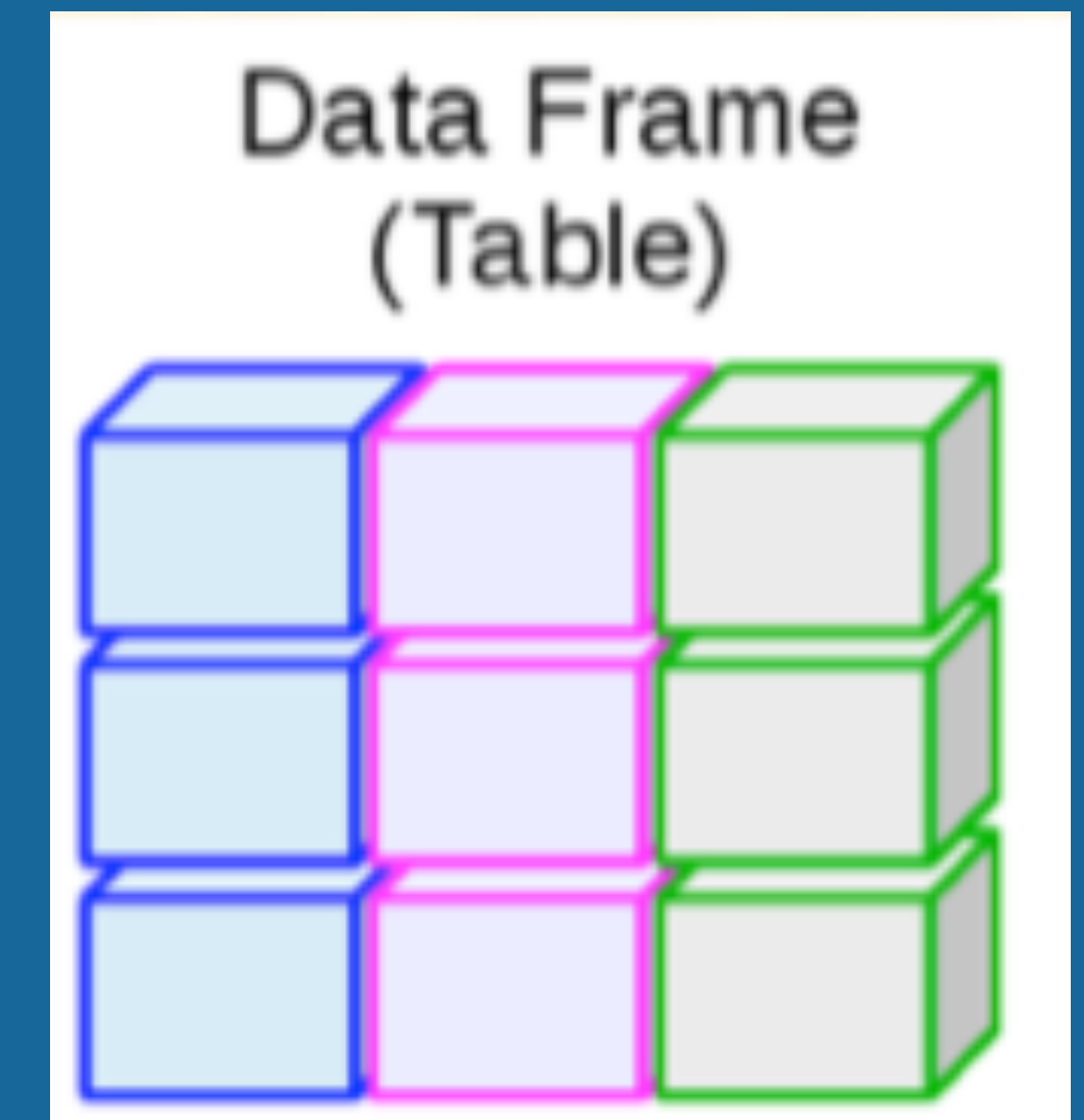
Otra forma de seleccionar variables

`head()`

Primeras 6 filas

`tail()`

Últimas 6 filas



LISTAS

CREACIÓN DE LISTAS

```
l <-list( x = 1:5, y = c("a","b"))
```

SELECCIÓN DE ELEMENTOS DE UNA LISTA

```
l[[2]]
```

Seleccionar el segundo elemento

VECTOR

```
l[1]
```

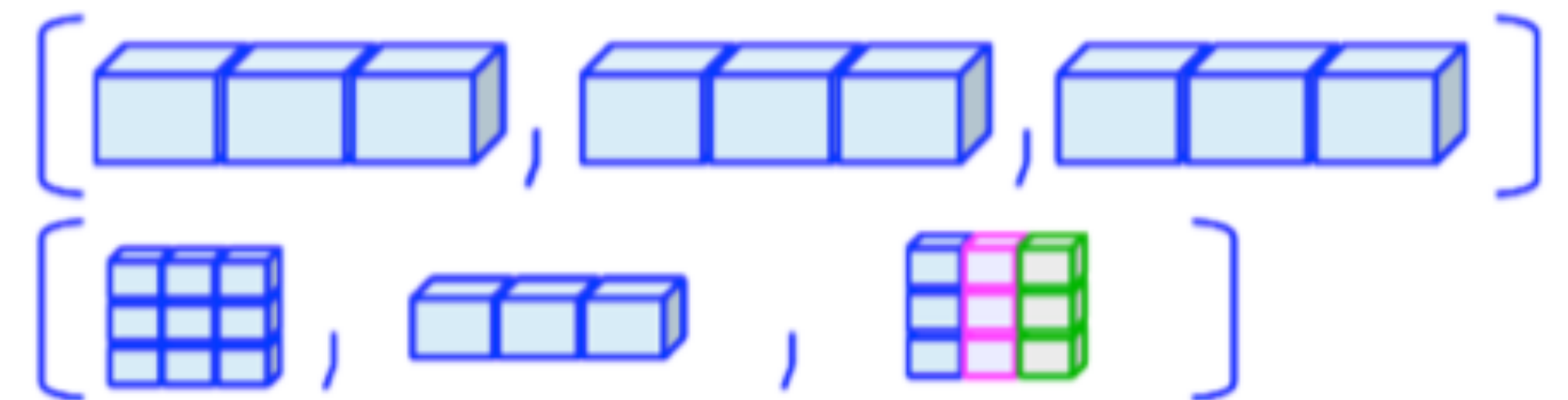
Seleccionar el primer elemento y lo guarda en una lista

LISTA

```
l$x
```

Selecciona el elemento llamado x

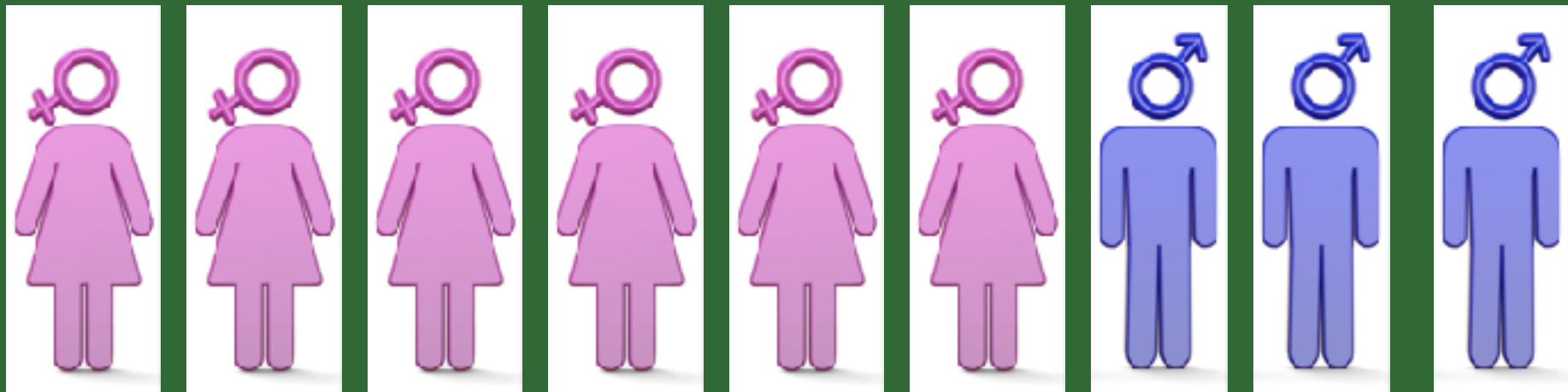
VECTOR



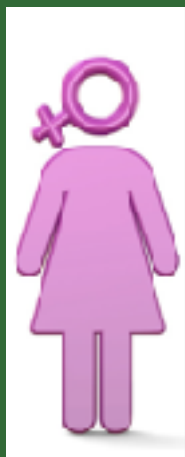
FACTORES

CREACIÓN DE FACTORES

fact(x, levels, labels)



levels(x)



table(x)

6 MUJERES

3 HOMBRES

FECHAS Y TIEMPO

DATE

`as.Date('2018-09-07', format='%Y-%m-%d')`

POSIX

POSIXct

`as.POSIXct()`

Guarda la fecha como un valor numérico

POSIXlt

`as.POSIXlt()`

Descompone el número como una lista

DATOS

IMPORTAR DATOS

<code>read.table()</code>	File, header, sep, dec	Leer un archivo .txt
----------------------------	------------------------	----------------------

<code>read.csv()</code>	File, header, sep, dec	Leer un archivo .csv
-------------------------	------------------------	----------------------

EXPORTAR DATOS

<code>write.table()</code>	Nombre del objeto, file, header, sep, dec	Escribir un archivo .txt
-----------------------------	---	--------------------------

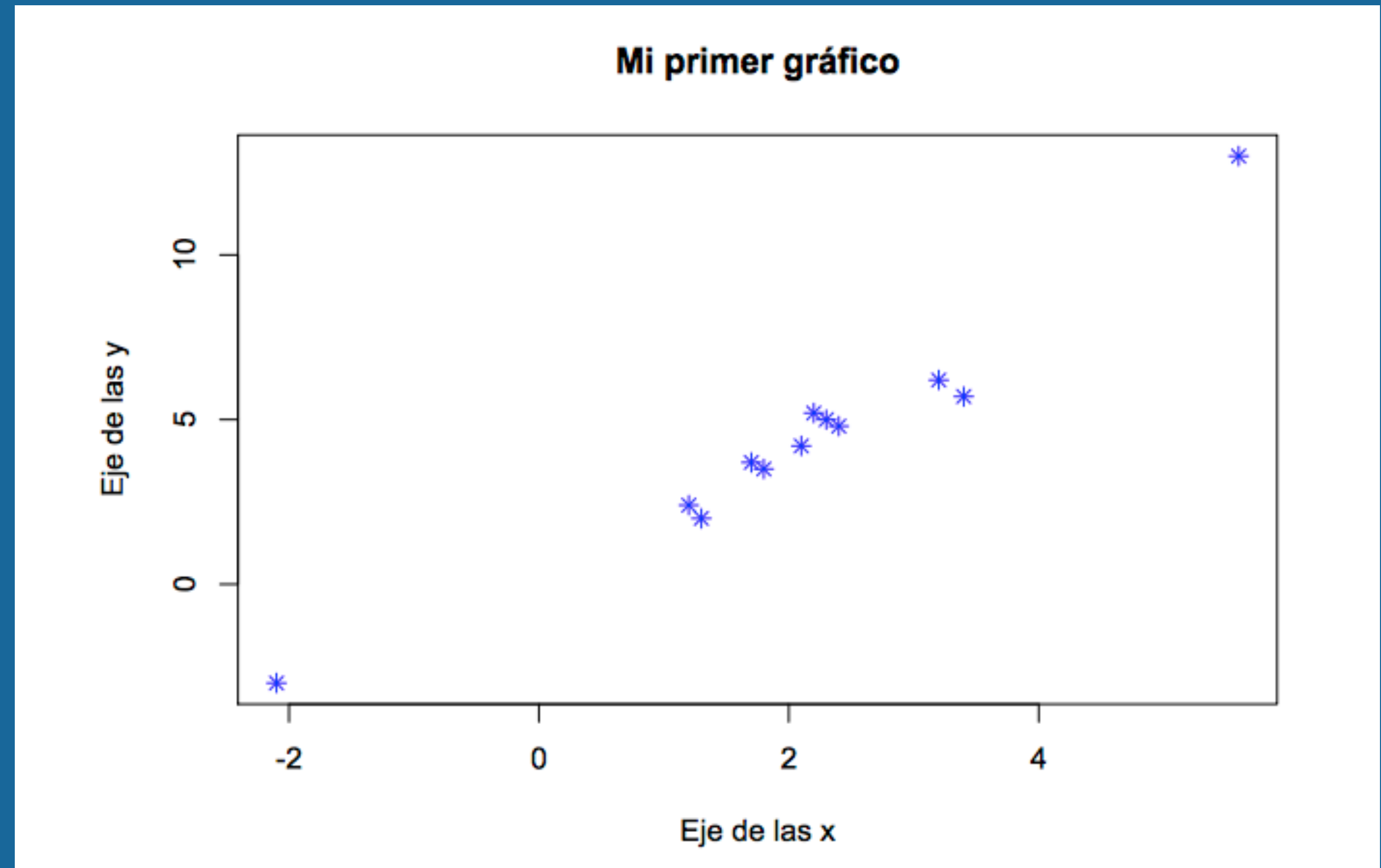
<code>write.csv()</code>	Nombre del objeto, file, header, sep, dec	Escribir un archivo .csv
--------------------------	---	--------------------------

GRÁFICOS

GRÁFICO DE PUNTOS

plot()

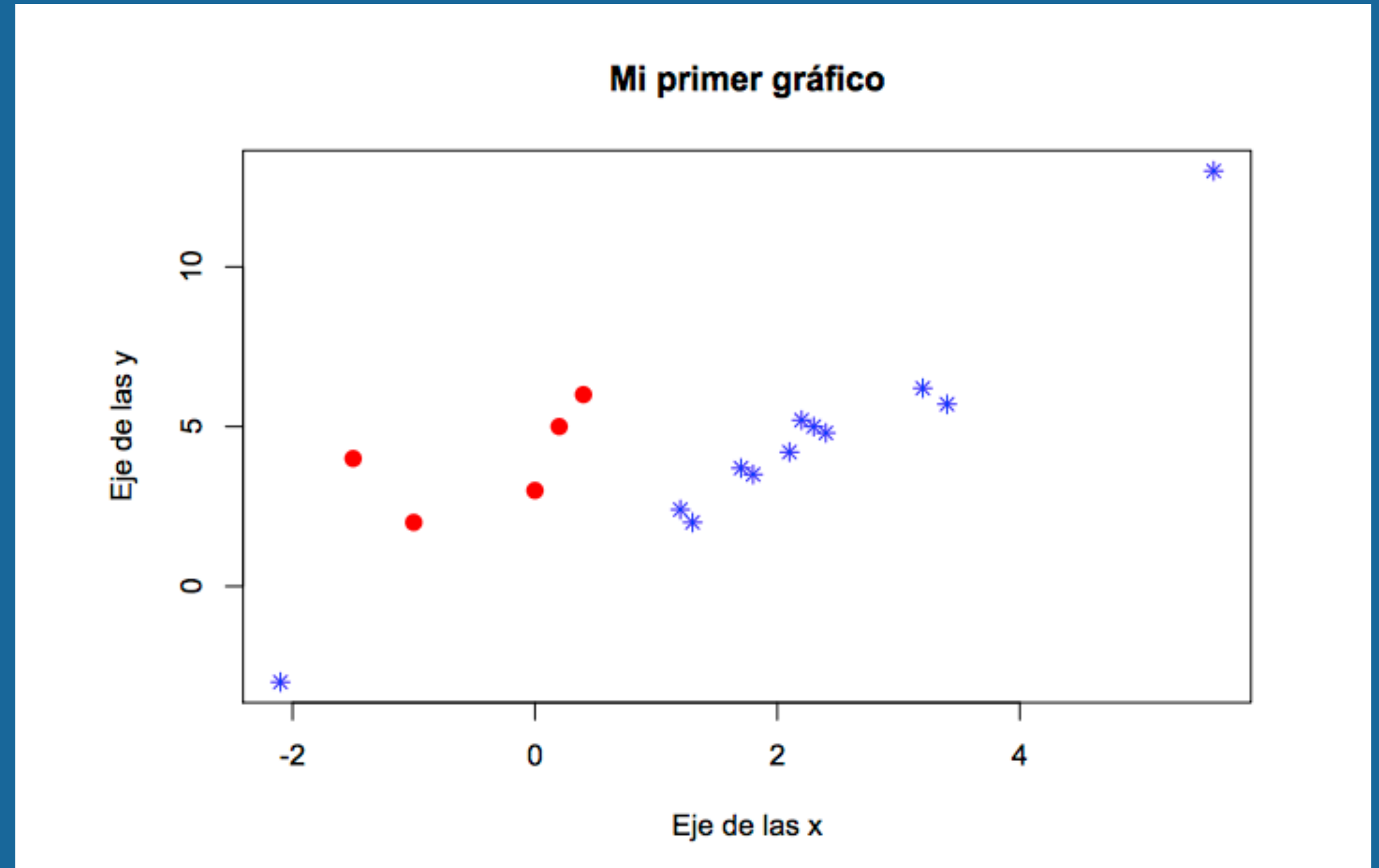
- ▶ x
- ▶ y
- ▶ pch
- ▶ col
- ▶ xlab
- ▶ ylab
- ▶ main



AGREGAR PUNTOS

points()

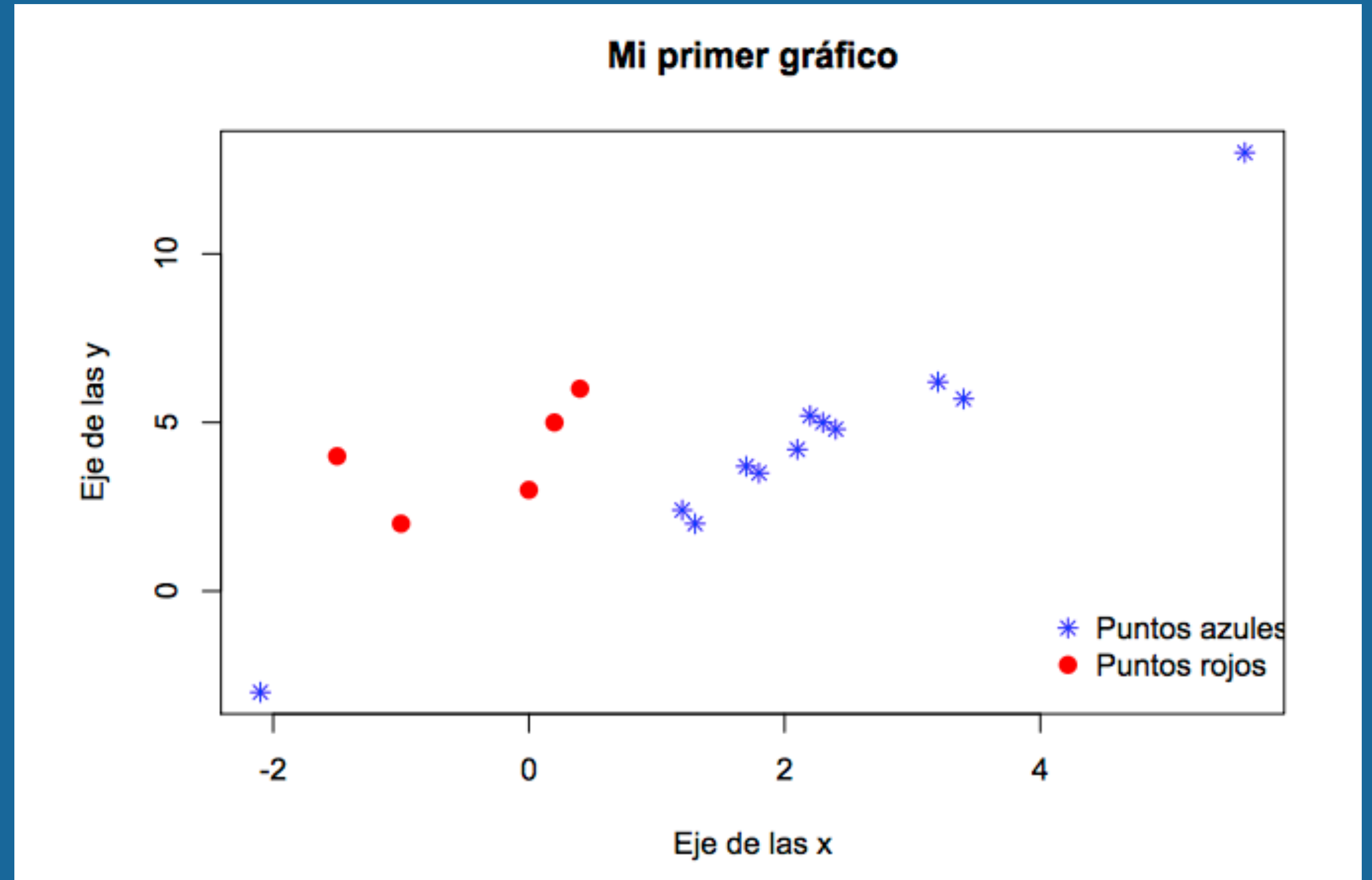
- ▶ x
- ▶ y
- ▶ pch
- ▶ col



AGREGAR LEYENDAS

legend()

- ▶ x
- ▶ y
- ▶ pch
- ▶ col
- ▶ Legend



GRÁFICOS DE LINEAS

`plot()` ▶ `type = 'l'`

`lines()`

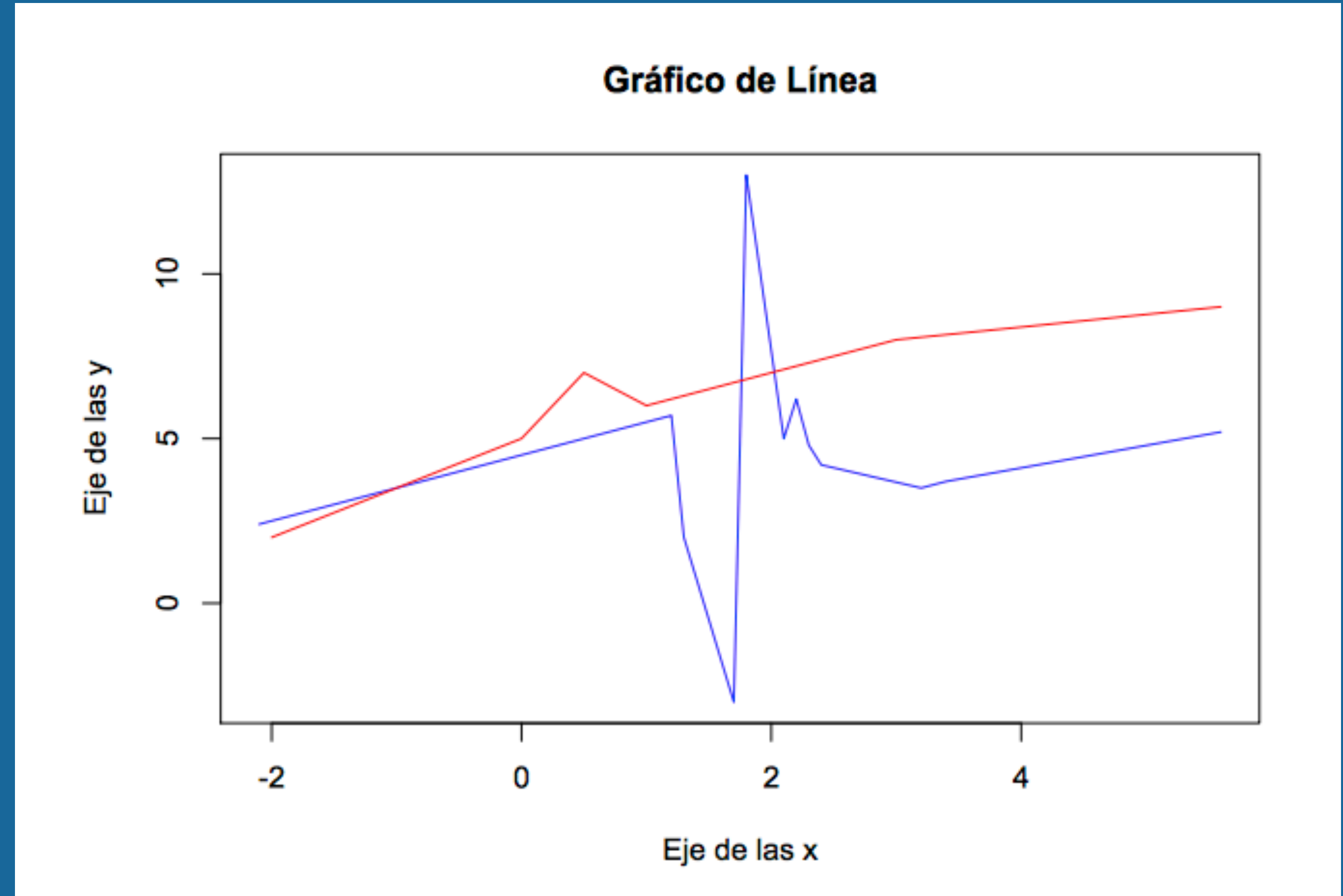


GRÁFICO DE TORTA

pie()

- ▶ x
- ▶ Labels
- ▶ col

Gráfico de Torta de las Ciudades

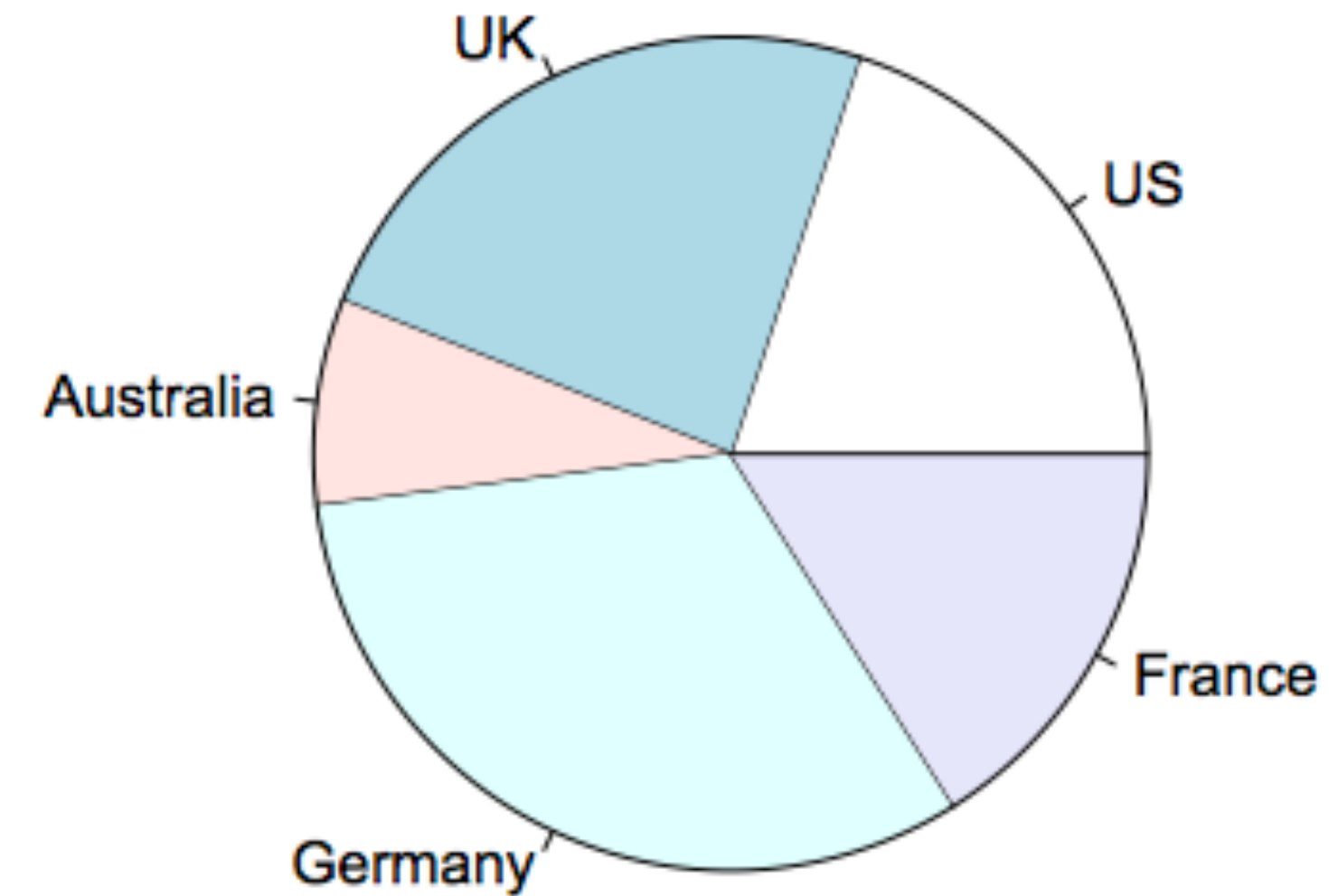
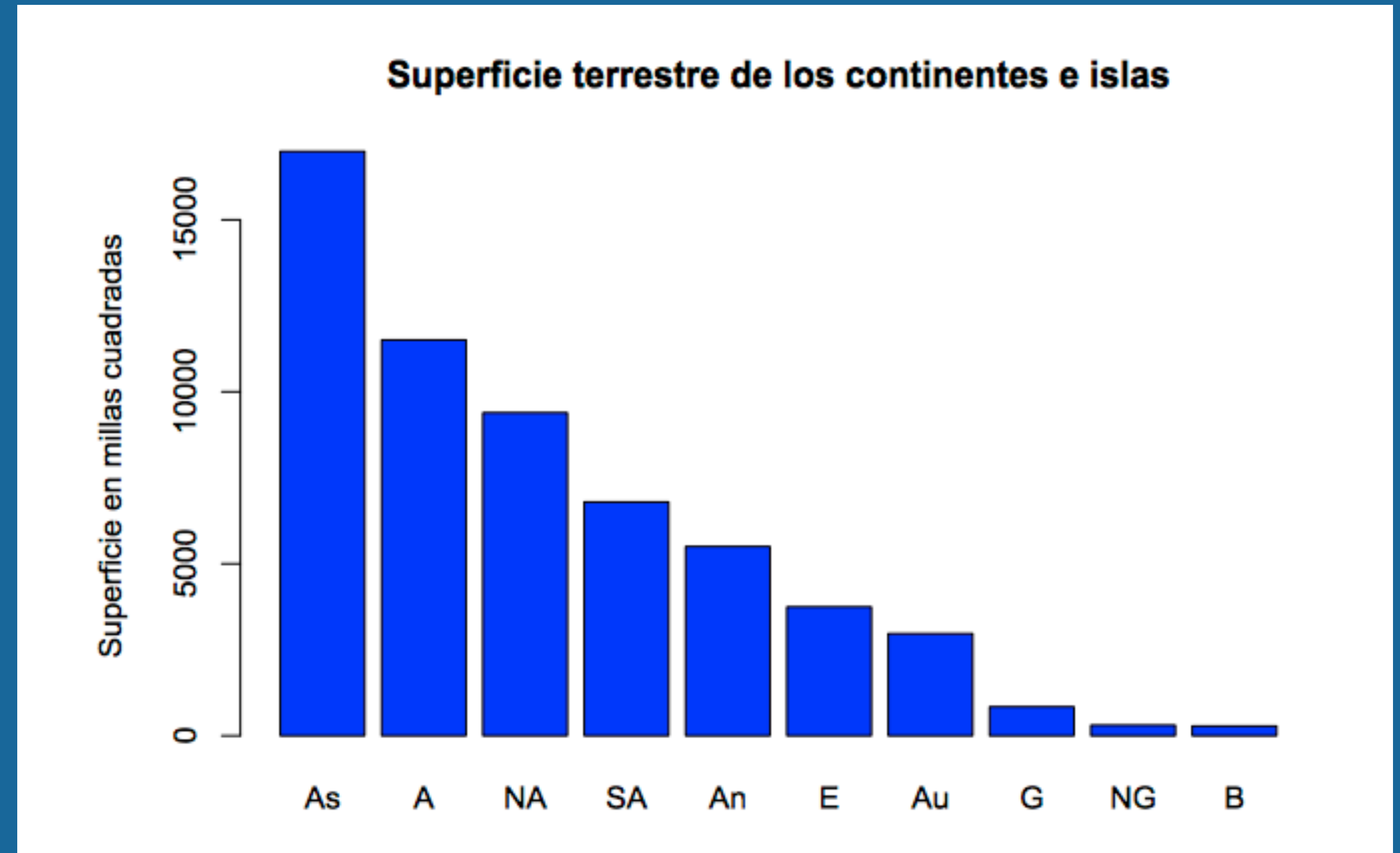


GRÁFICO DE BARRAS

barplot()

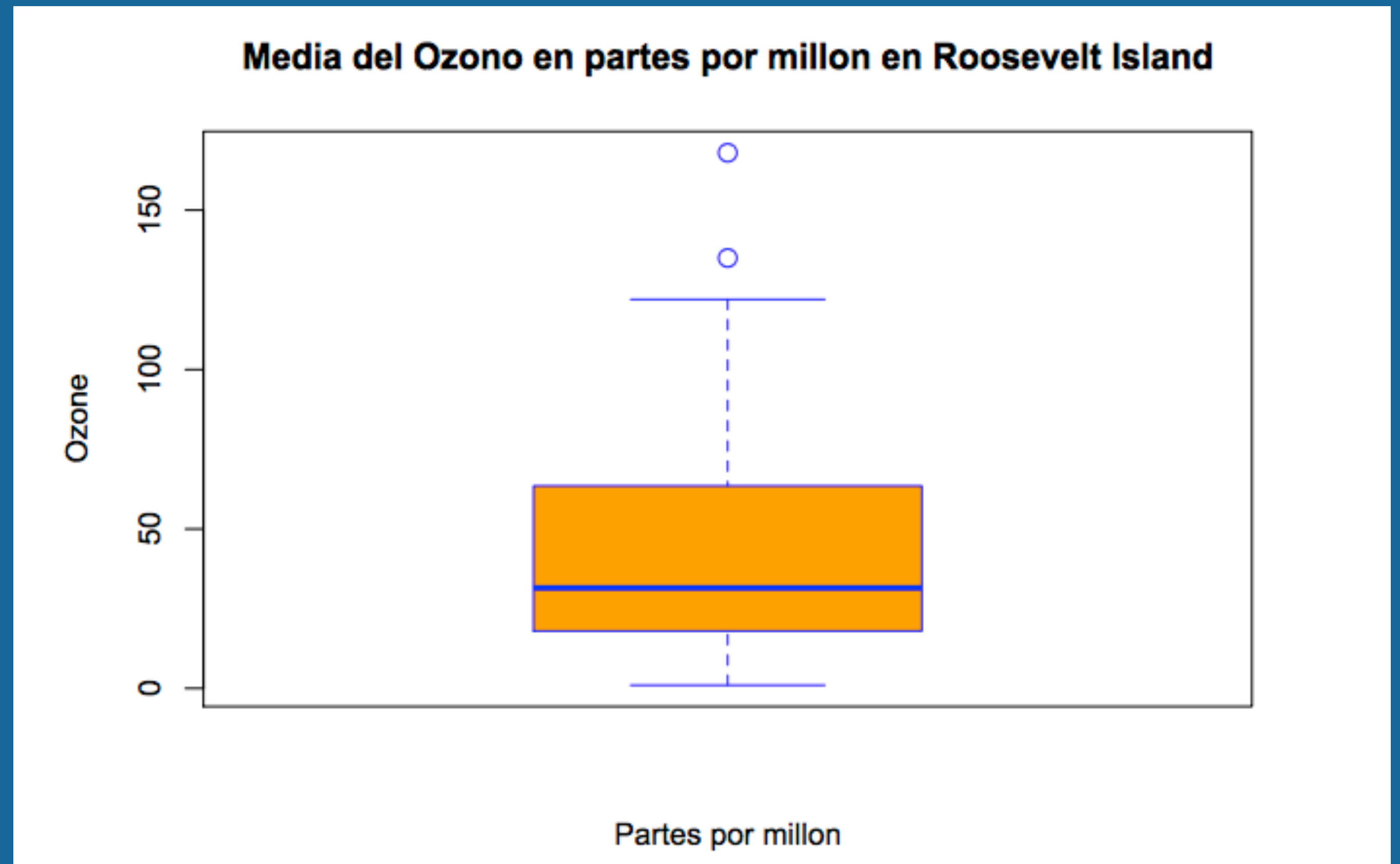
- ▶ height
- ▶ horiz
- ▶ names.arg



BOXPLOT

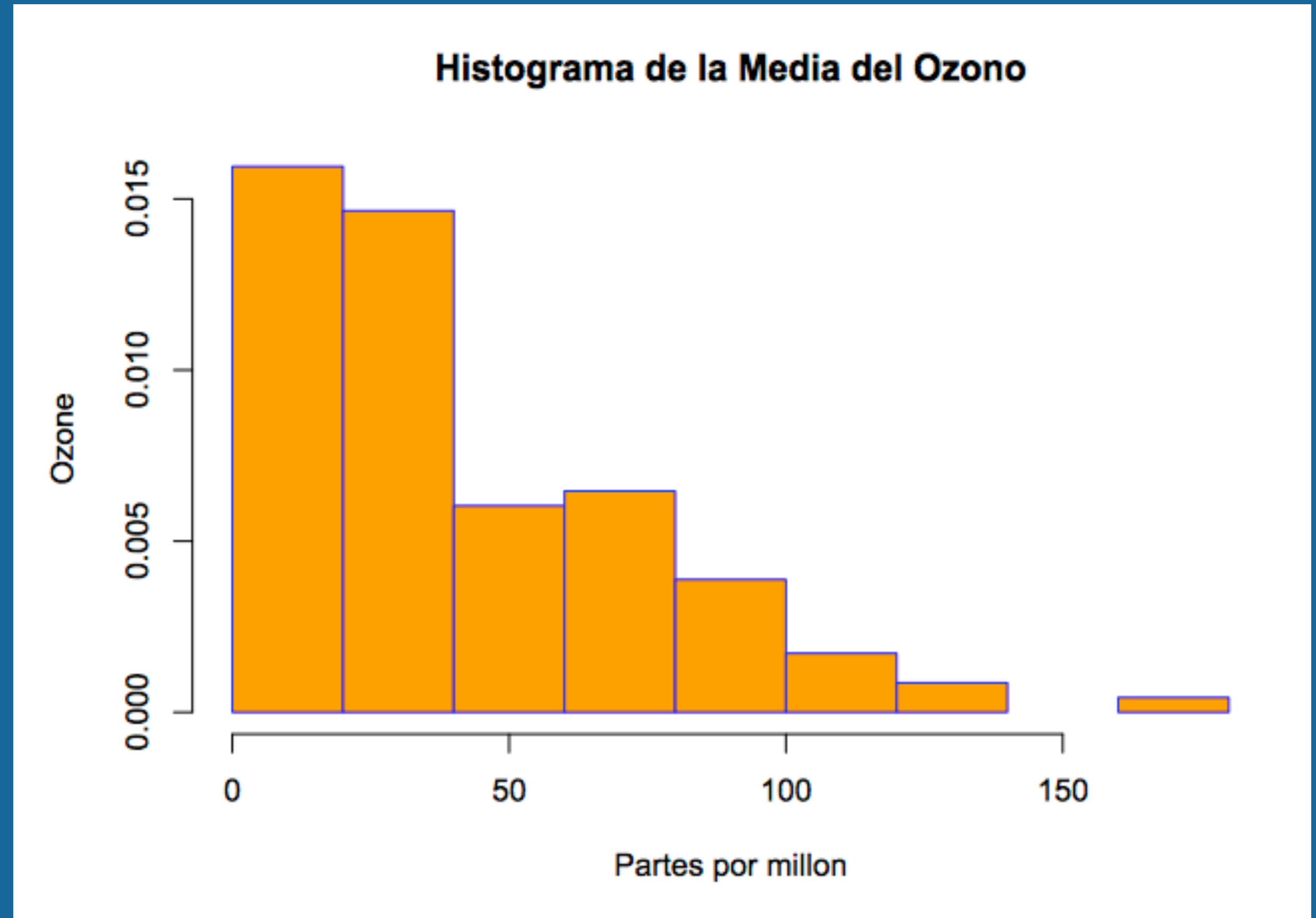
boxplot()

- Valor mínimo
- Valor máximo
- Primer cuartel
- Mediana
- Tercer cuartil



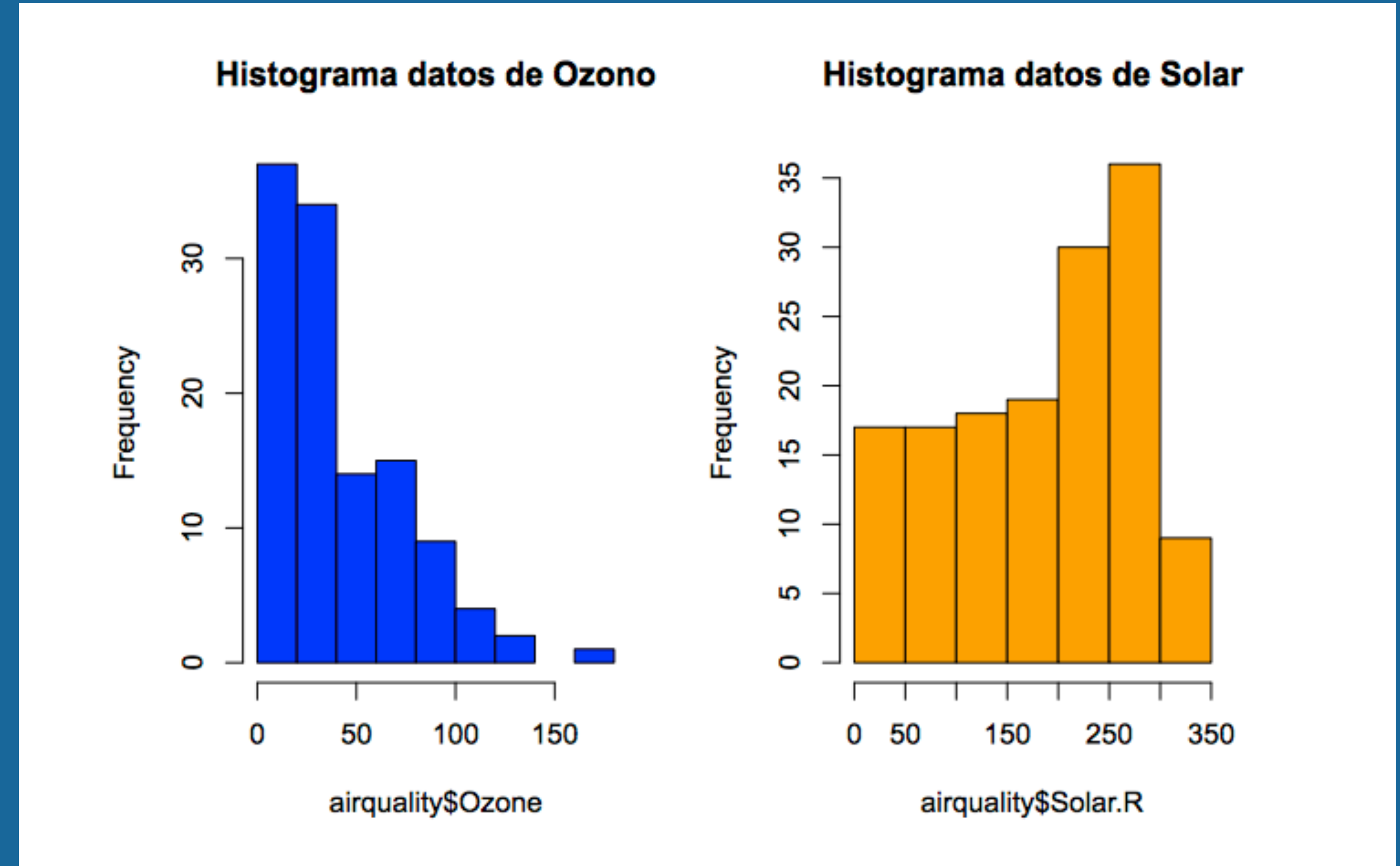
HISTOGRAMAS

barplot() ▶ breaks
▶ freq



MÚLTIPLES GRÁFICOS POR VENTANAS

```
par(mfrow=c(1,2))
```



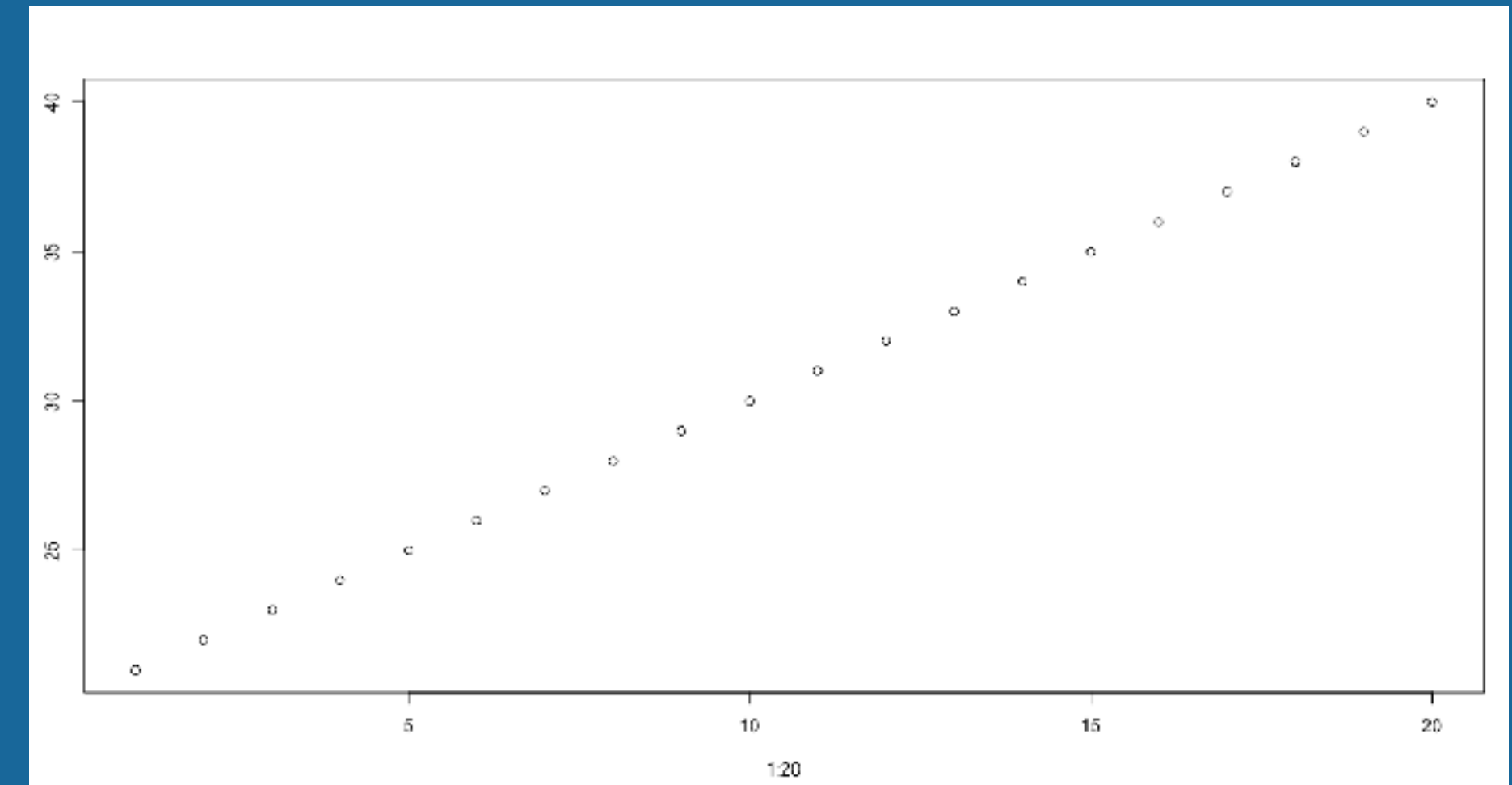
GUARDAR GRÁFICOS

jpeg()

- ▶ file
- ▶ quality

dev.off()

```
jpeg(file="imagen.jpg", quality = 60)  
plot(x=1:20,y=21:40)  
dev.off()
```



FUNCIONES

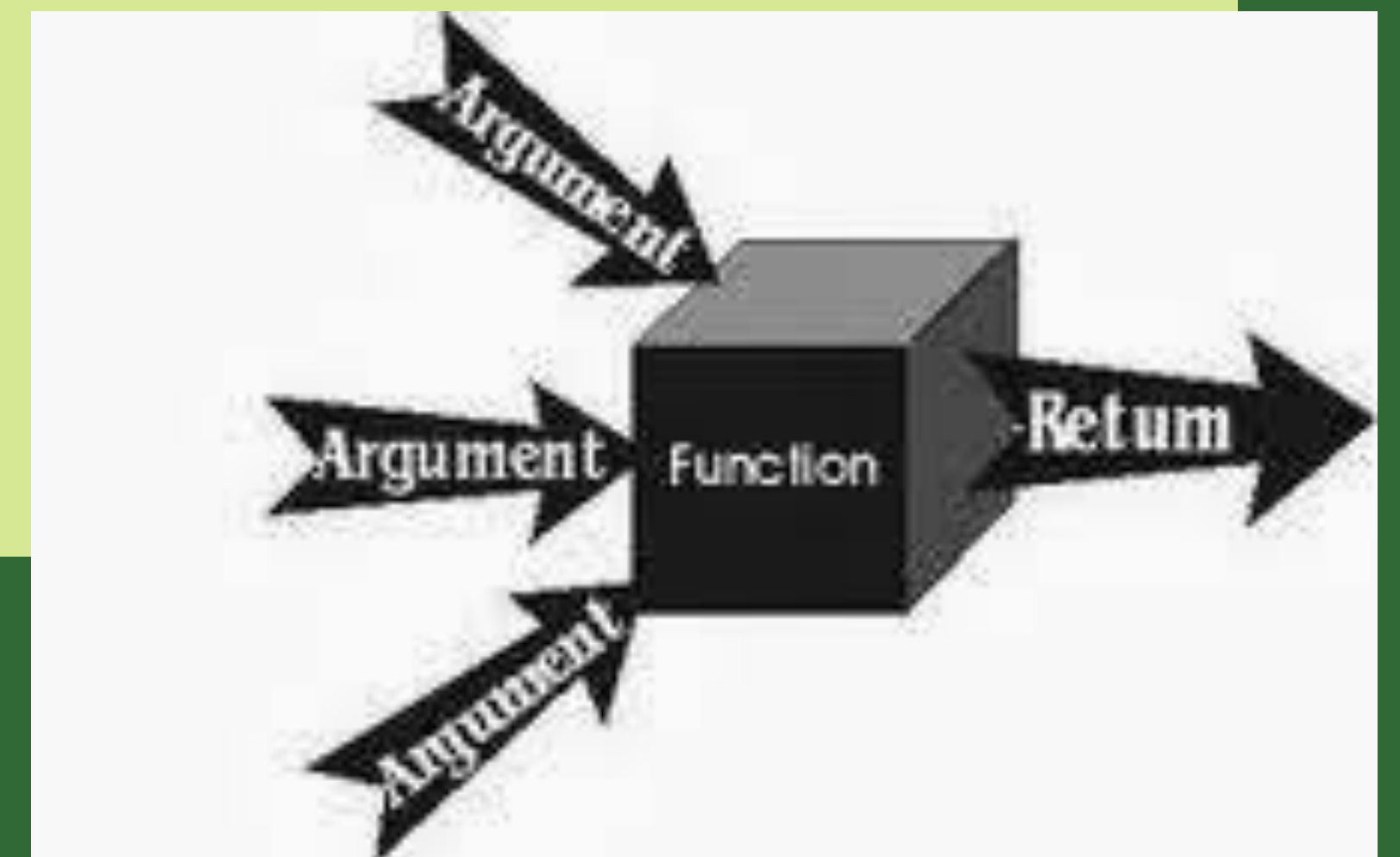
CREACIÓN DE FUNCIONES

```
Nombre_funcion <- function (argumento_1, argumento_2,...){
```

Expresión

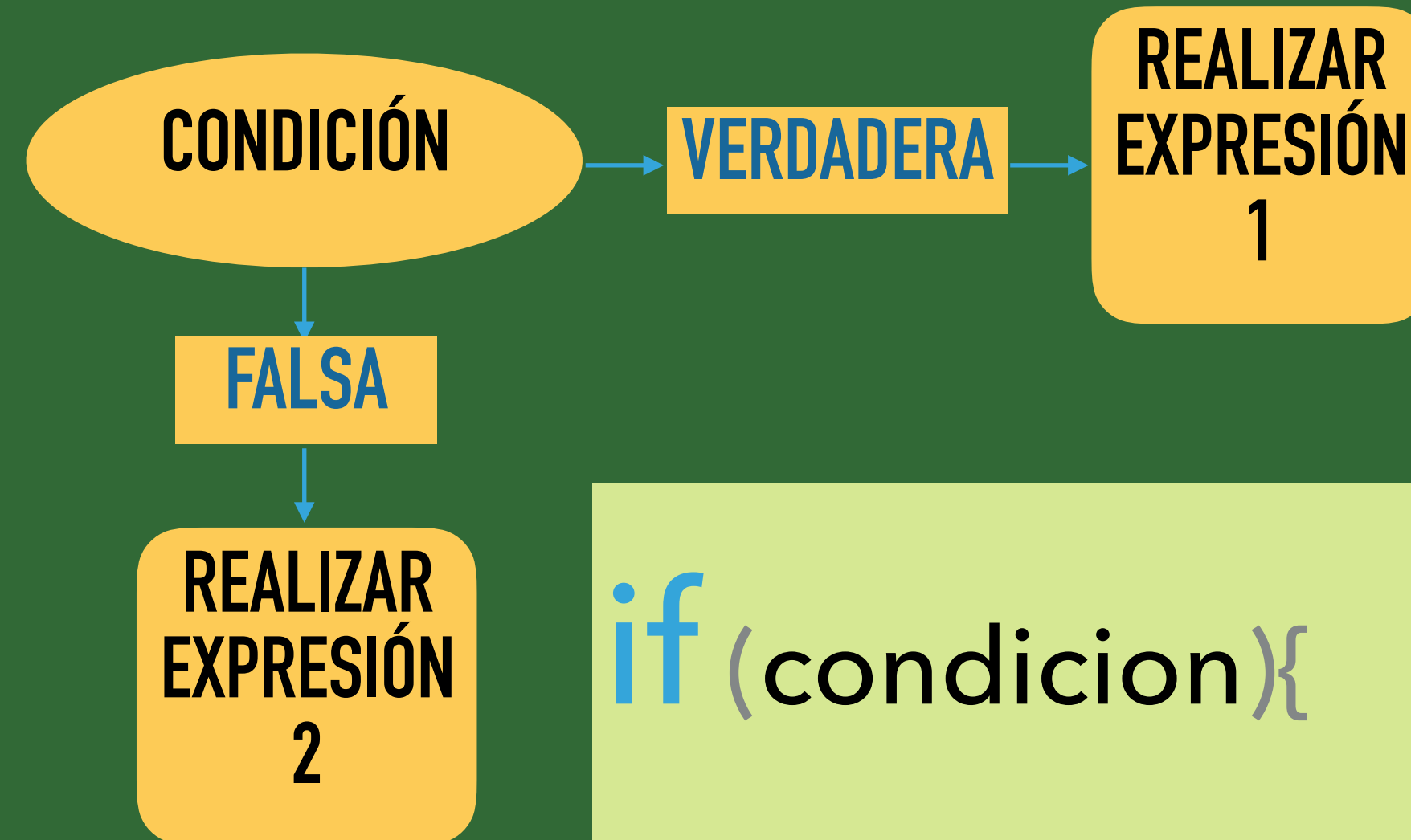
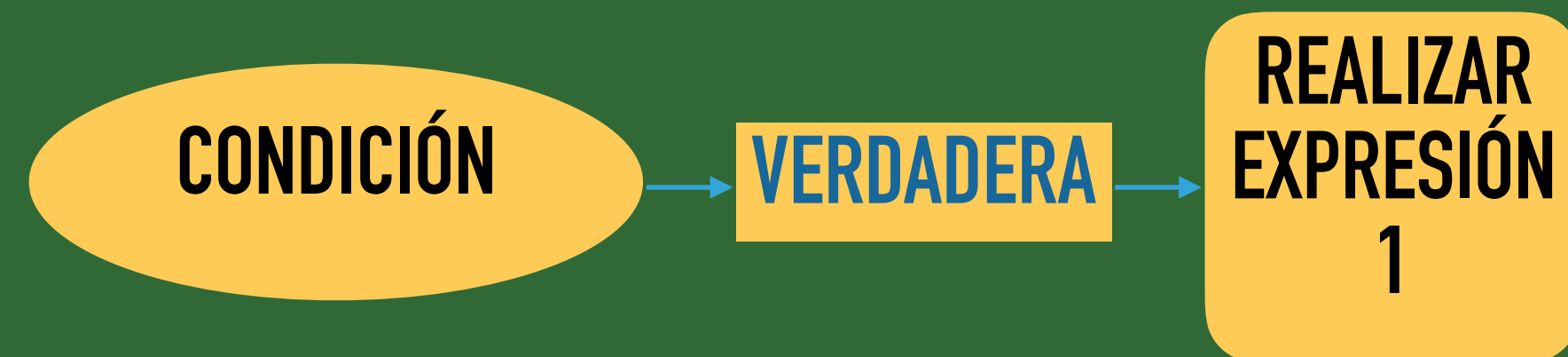
```
return( objeto )
```

```
}
```



CONDICIONALES

```
if (condicion){  
  Expresión  
}
```



```
if (condicion){  
  Expresión 1  
} else {  
  Expresión 2  
}
```

BUCLES

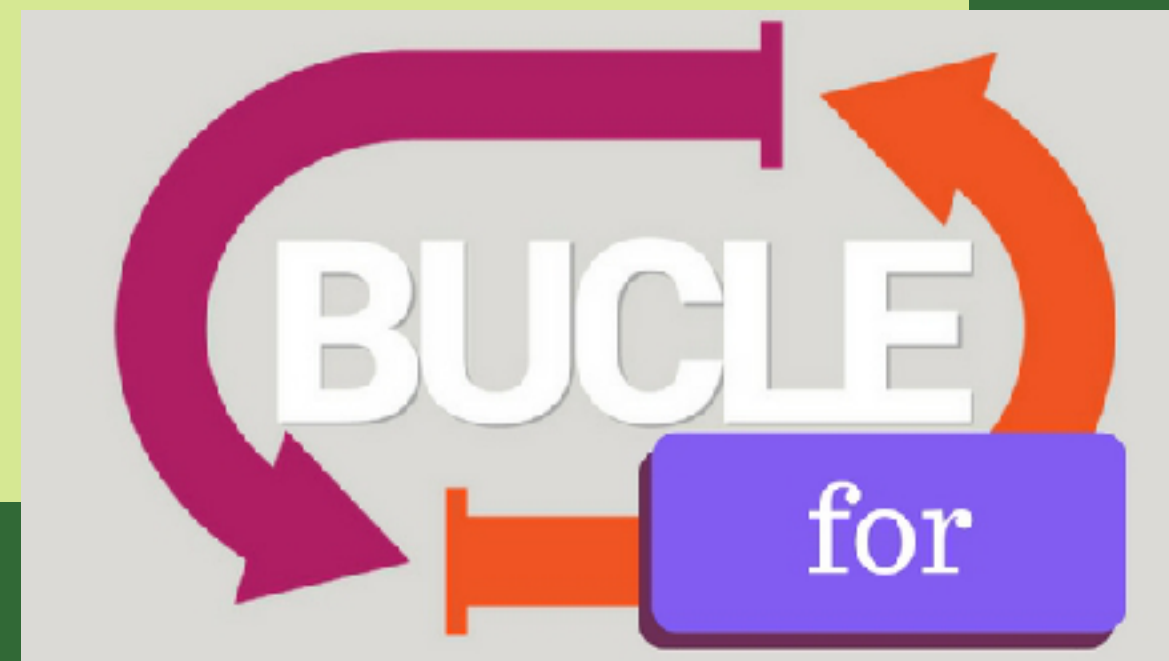
```
for (variable in secuencia){
```

Comando 1

Comando 2

...

```
}
```



```
while ( condicion){
```

Instrucciones

```
}
```



MUCHAS
GRACIAS