

# Introducción a R

author: Prácticas Radiación y Ruido date: Noviembre 2018 autosize: true

## ¿Qué es R?

El entorno R puede ser definido como un conjunto de herramientas que, entre otras cosas, facilitan:

- el manejo y el análisis de datos
- la representación gráfica
- un lenguaje de programación, R
- el cálculo de funciones estadísticas

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

R es software libre con licencia GNU-GPL.

## R Packages

R tiene actualmente más de 6000 paquetes disponibles.

- Un paquete es un conjunto de funciones agrupadas en librerías.
- Todos los paquetes de R están disponibles.
- Facilidad para encontrar documentacion. Comunidad activa.

## R Documentación

Algunos recursos

- **Manuales:** An introduction to R <https://cran.r-project.org/>
- Foros: **stackoverflow** <https://stackoverflow.com/>
- Rbloggers
- Recursos de expertos: <https://github.com/oscarperpinan/r-intro-eoi>

## Rstudio

## Primeros pasos

En R podemos definir ‘objetos’ y un conjunto de ‘clases’:

- vectores
- listas
- ...

Cada uno de los objetos definidos en R pertenecerá a una clase:

- numérico
- factor
- data.frame

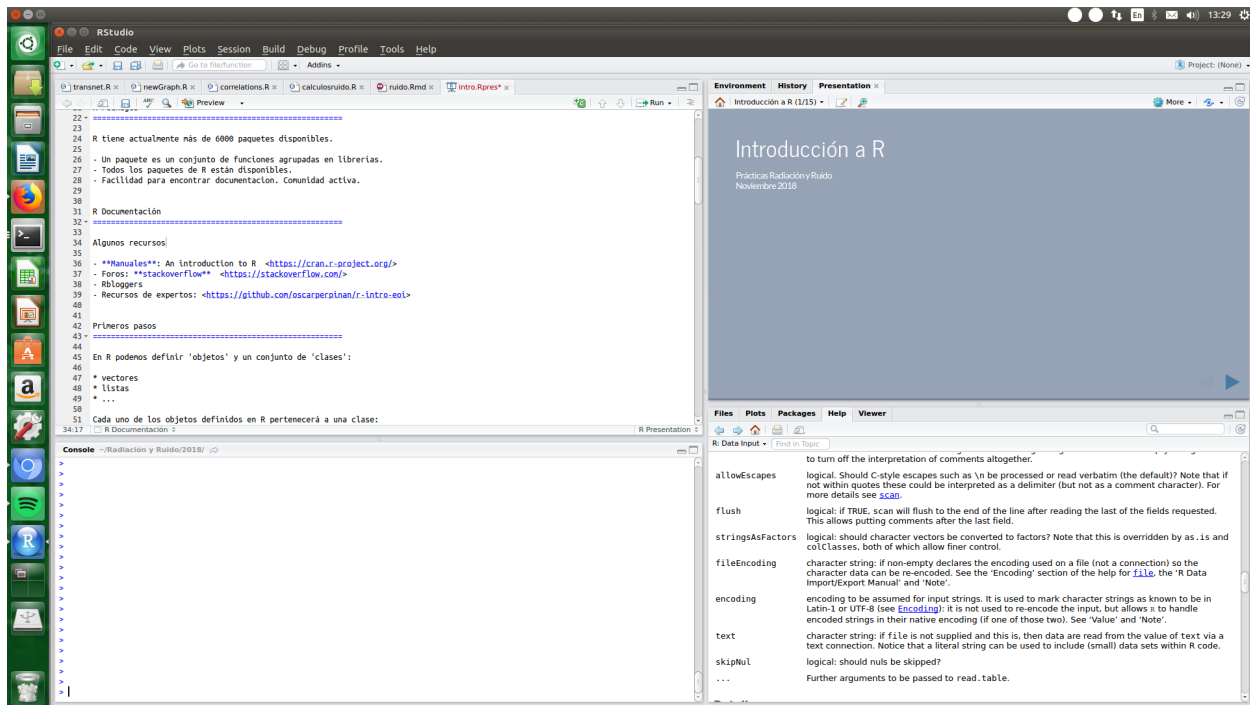


Figure 1: alt text

## Primeros pasos

Creando un vector en R:

```
x <- c(1, 3, 7, 8, 10)
x
```

```
[1] 1 3 7 8 10
```

Podemos ver la clase del objeto que hemos creado utilizando la función `class`.

```
class(x)
```

```
[1] "numeric"
```

## Primeros pasos

Otras formas de crear vectores:

```
x <- seq(1, 20, 2)
x
```

```
[1] 1 3 5 7 9 11 13 15 17 19
```

```
class(x)
```

```
[1] "numeric"
```

Busca otras formas de crear vectores

Crea un vector entre 1 y 500 de tal manera que entre 1 y 250 vaya de 1 en 1 y entre 250 y 500 de 10 en 10

## Primeros pasos

Creo una lista

```
x <- seq(1, 20, 2)
y <- c("a", "b")
```

```
lista <- list(x, y)
lista
```

```
[[1]]
[1] 1 3 5 7 9 11 13 15 17 19
```

```
[[2]]
[1] "a" "b"
```

```
class(lista)
```

```
[1] "list"
```

## Primeros pasos

Creo una matriz

```
z <- 1:12
M <- matrix(data=z, nrow=4, ncol=3)
M
```

```
      [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12
```

## Primeros pasos

Cargo datos por defecto en R y los leo:

```
data(cars)
head(cars)
```

```
  speed dist
1     4     2
2     4    10
3     7     4
4     7    22
5     8    16
6     9    10
```

## Primeros pasos

¿qué clase de objeto es 'cars'?

```
class(cars)
```

```
[1] "data.frame"
```

¿qué tamaño tiene 'cars'?

```
dim(cars)
```

```
[1] 50  2
```

## Primeros pasos

¿Existe otra manera de conocer el número de filas del data.frame 'cars'?

```
nrow(cars)
```

```
[1] 50
```

¿Existe una función equivalente para el número de columnas?

## Primeros pasos

¿Puedo conocer el nombre de las variables del data.frame?

```
names(cars)
```

```
[1] "speed" "dist"
```

**Encuentra otra función para conocer los nombres de las columnas en un data.frame o matriz**

Primeros pasos =====

Para obtener información sobre lo que hace una función:

```
?head
```

La descripción de la función y cómo usarla nos aparece en la ventana inferior derecha si usamos Rstudio.

- Puedes echar un vistazo a algunas funciones:
- sum
- max
- plot
- tail
- head
- ...

## Primeros pasos

```
summary(cars)
```

```
      speed      dist
Min.   : 4.0    Min.   : 2.00
1st Qu.:12.0    1st Qu.: 26.00
```

```

Median :15.0   Median : 36.00
Mean   :15.4   Mean     : 42.98
3rd Qu.:19.0   3rd Qu.: 56.00
Max.    :25.0   Max.     :120.00

```

## Primeros pasos

¿Puedo extraer la primera columna?

```

cars1 <- cars[,1]
cars1

```

```

[1]  4  4  7  7  8  9 10 10 10 11 11 12 12 12 12 13 13 13 13 14 14 14 14
[24] 15 15 15 16 16 17 17 17 18 18 18 18 19 19 19 20 20 20 20 20 22 23 24
[47] 24 24 24 25

```

¿Puedo extraer las filas entre 15 y 30?

```

carsR <- cars[15:30,]
carsR

```

```

      speed dist
15      12    28
16      13    26
17      13    34
18      13    34
19      13    46
20      14    26
21      14    36
22      14    60
23      14    80
24      15    20
25      15    26
26      15    54
27      16    32
28      16    40
29      17    32
30      17    40

```

## Primeros pasos

Una característica importante del lenguaje R es el indexado con \$.

Podemos acceder a las distintas variables de un data.frame por su nombre:

```

cars$speed

```

```

[1]  4  4  7  7  8  9 10 10 10 11 11 12 12 12 12 13 13 13 13 14 14 14 14
[24] 15 15 15 16 16 17 17 17 18 18 18 18 19 19 19 20 20 20 20 20 22 23 24
[47] 24 24 24 25

```

```

cars$dist

```

```

[1]   2  10   4  22  16  10  18  26  34  17  28  14  20  24  28  26  34
[18]  34  46  26  36  60  80  20  26  54  32  40  32  40  50  42  56  76

```

[35] 84 36 46 68 32 48 52 56 64 66 54 70 92 93 120 85

## Primeros pasos: representación

Representación de los datos de 'cars':

```
plot(cars)
```

## Primeros pasos: representación

Representación de los datos de 'cars' con algunos cambios:

```
plot(cars, type='p', col='red', main='cars')
```

Prueba a cambiar los títulos de los ejes

## Primeros pasos: directorio de trabajo

Es necesario conocer la ubicación de nuestro trabajo. Para esto dos funciones serán de mucha utilidad:

- `setwd()`
- `getwd()`

¿Podríaís decirme qué hace cada una de ellas?

## Primeros pasos: cargar datos

R permite cargar datos desde ficheros .csv, .xls o directamente desde su dirección url.

- Cargo un fichero de datos de ruido:

```
ruidoLunes <- read.csv("data/lun_cuba_10min.csv", dec=',')
```

¿Qué clase de objeto he creado?

## Primeros pasos: cargar datos

Cargo un nuevo fichero de datos:

```
ruidoMartes <- read.csv("data/mar_cuba_10min.csv", dec=',')
```

Función *cbind* y *rbind*:

```
data <- rbind(ruidoLunes,ruidoMartes)
```

## Primeros pasos: cargar datos

Miro las dimensiones de 'data' frente a los archivos aislados.

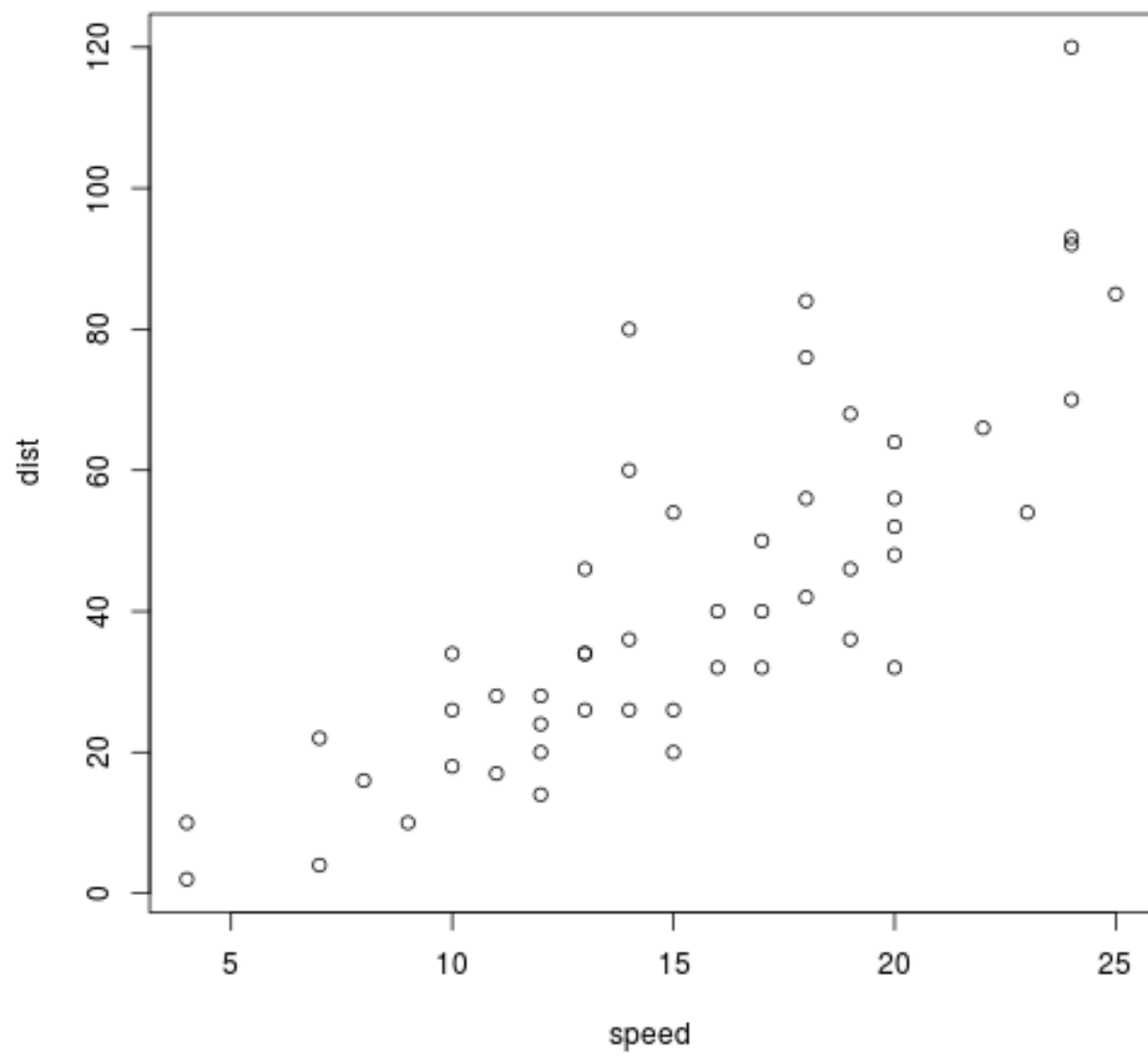


Figure 2: plot of chunk unnamed-chunk-19

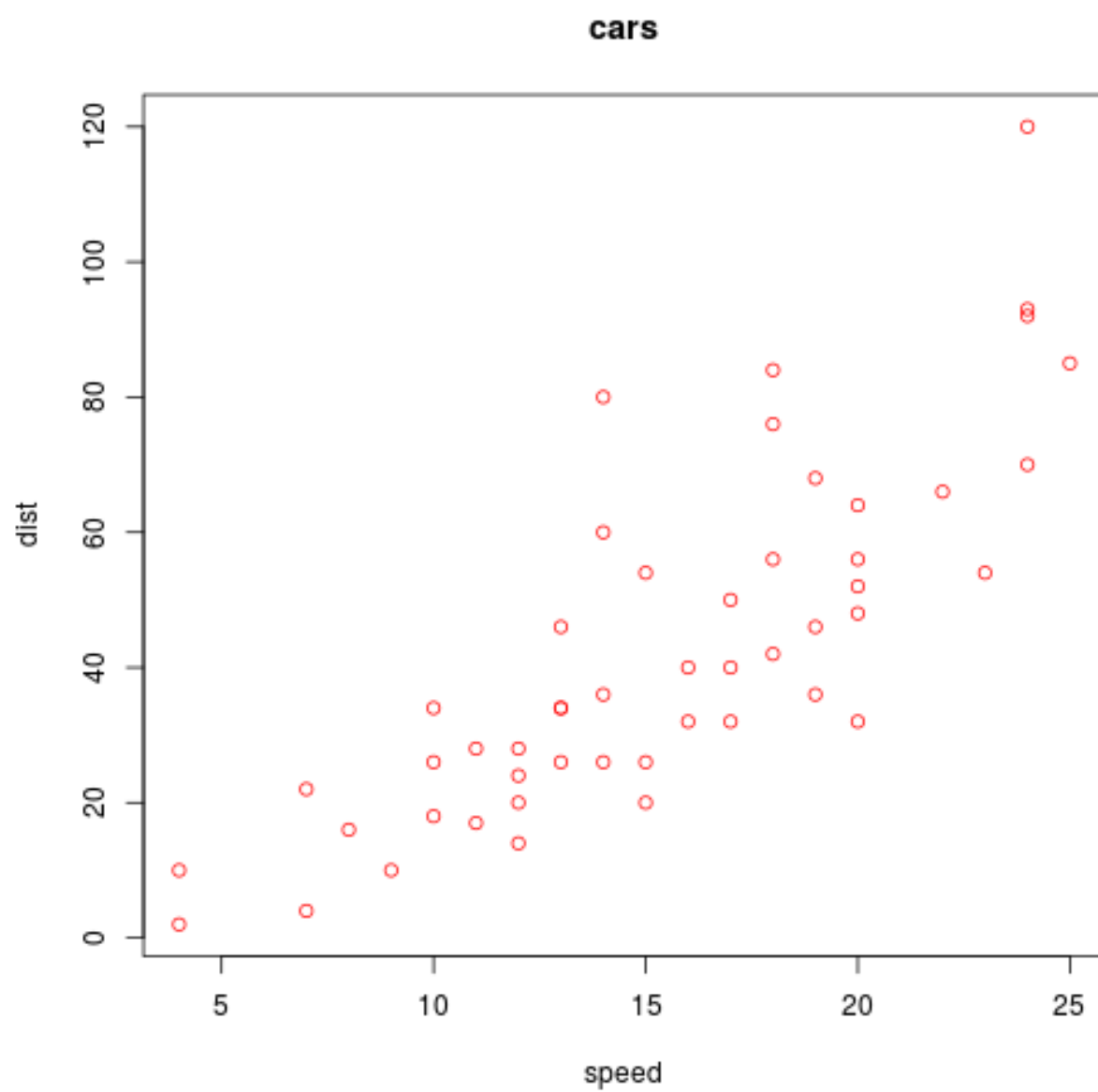


Figure 3: plot of chunk unnamed-chunk-20



```
dim(ruidoMartes)
```

```
[1] 144  2
```

```
dim(data)
```

```
[1] 288  2
```

Une los ficheros de lunes y martes manteniendo el número de filas del objeto de salida igual al de entrada