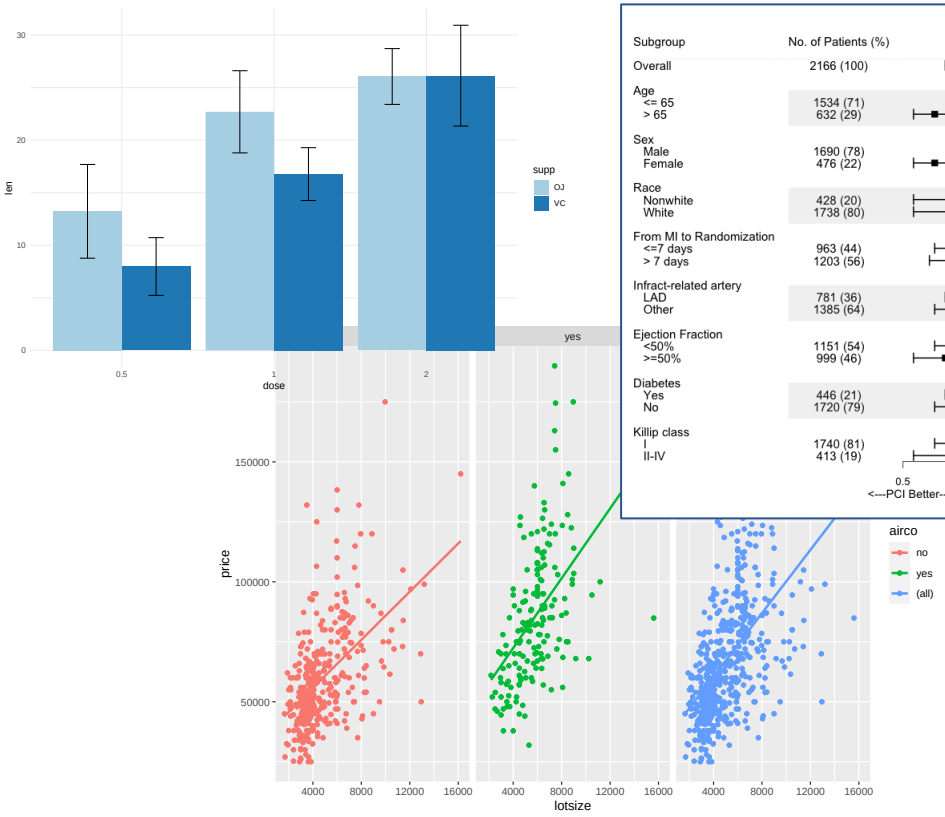
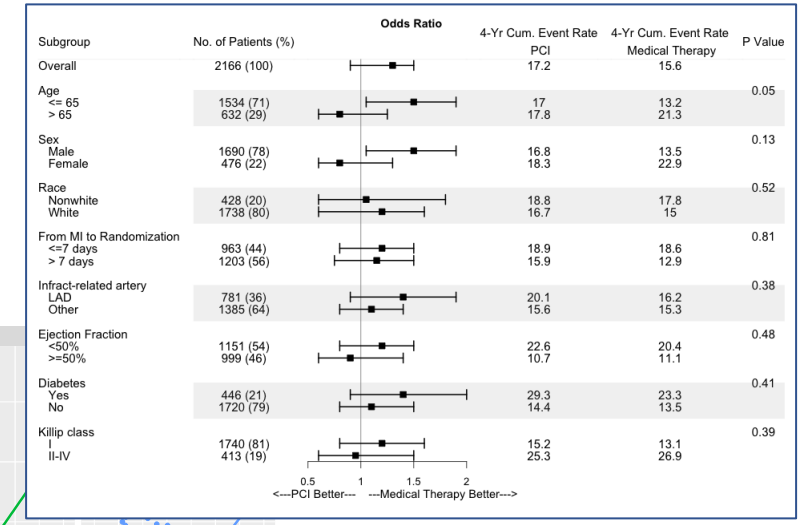
Análisis Estadístico con





Introducción a R

Índice

Introducción a R

1. Que es R

- **R** es un **lenguaje** y **entorno de programación** para análisis estadístico y gráfico **gratuito** (“high-level language)

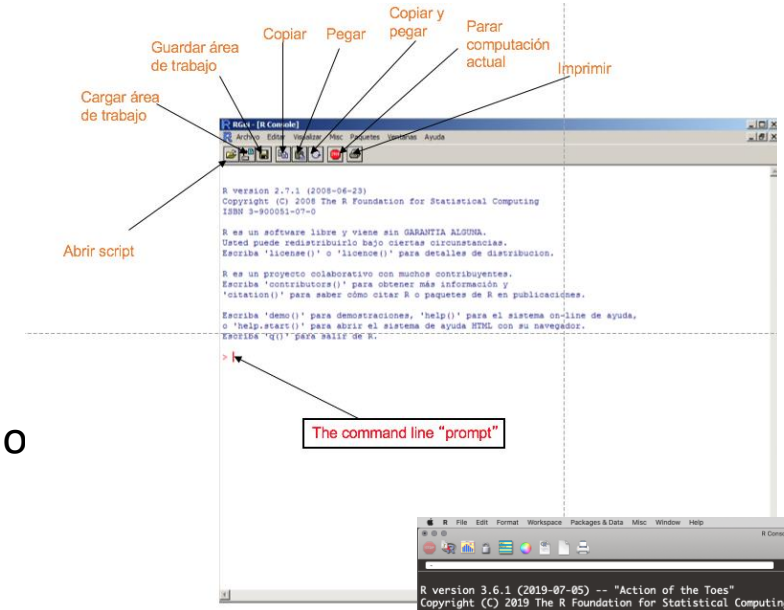
- **Lenguaje de programación orientado a objetos**.

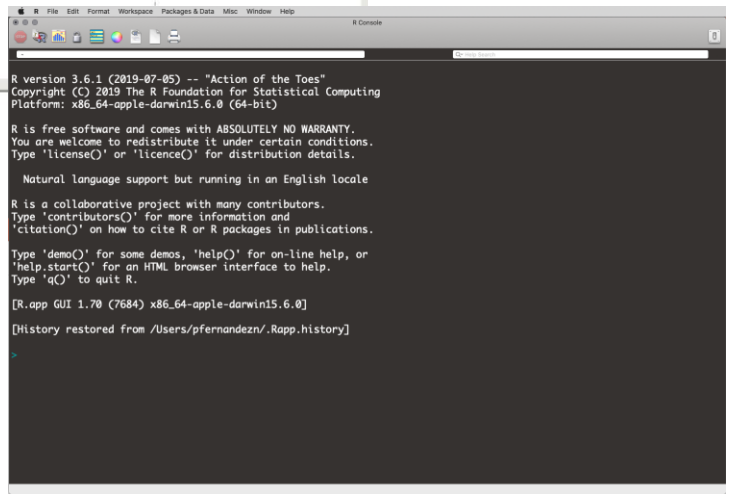
- Proyecto de software libre, resultado de la implementación GNU del premiado lenguaje “S” y “Scheme”.

Introducción a R

2. Aspectos relacionados con el software

- Menús y prompt
- Workspace o área de trabajo
- Ayuda





Introducción a R

2. Aspectos relacionados con el software

Ayuda (comandos)

>?rnorm

>help.start()

>?help.search

>help.search("normal")

>?apropos

>apropos("normal")

>?demo

>demo(graphics); demo(persp);

>demo(lm.glm)

Introducción a R

3. Objetos

- Tipos de objetos
- Creación
- Nombres
- Comandos
- Atributos

Introducción a R

3. Objetos: *Tipos de objetos*

Tipo de Objeto	Definición	Ejemplo	Observaciones
Vector	Colección ordenada de elementos del mismo tipo	x<-c(1,2,3); z<-c(TRUE,FALSE,TRUE); y<-c("Low","Low","Medium","High")	Factor== tipo de vector de datos cualitativos
Array	Generalización multidimensional del vector. Elementos del mismo tipo.	matrix(rnorm(20),ncol=5)	x
Data.frame	Igual que el array pero puede tener columnas de distintos tipo.	Edades<-data.frame(ID=c("gen0", "genB", "genZ"),subj1 = c(10, 25, 33), subj2 = c(NA, 34, 15), oncogen = c(TRUE, TRUE, FALSE),loc = c(1,30, 125))	x
List	Una colección ordenada de objetos conocidos y sus componentes.	una.lista <- c(un.vector = 1:10,una.palabra = "hola",una.matriz = matrix(rnorm(20), ncol = 5),lista2 = c(a = 5,b = factor(c("a", "b"))))	x

Introducción a R

3. Objetos: *Creación*

NOMBRE OBJETO <- CONTENIDO

Introducción a R

3. Objetos: *Nombres*

- Los nombres válidos para un objeto son combinaciones de letras, números y el punto (“.”).
- Los nombres no pueden empezar con un número.
- R es “case-sensitive”: x != X.
- Hay nombres reservados (“function”, “if”, etc.).
- Otras consideraciones:
  -El uso del “.” es distinto del de C++.
  -Mejor evitar nombres que R usa (ej., “c”)
  -Las asignaciones se hacen con **"<-"** y se recomiendan los espacios.
  -El signo **"="** se reserva para los argumentos de las funciones.

Introducción a R

3. Objetos: *Comandos*

- **Ver contenido de un objeto:** introducimos el nombre del objeto en la línea de comandos

- **COMANDOS** para saber los objetos que hemos creado o cargado:

  > **ls**()

- Para borrar objetos concretos

  > **rm**(nombre de objetos separados por comas)

- Para borrar todos los objetos del entorno de trabajo:

  > **rm**(list = ls())

Introducción a R

3. Objetos: *Atributos*

```
> x <- 1:15
> length(x) # [1] 15
> class(x) # [1] "integer"

> y <- matrix(5, nrow = 3, ncol = 4
> dim(y) ) # [1] 3 4
> class(y) # [1] "matrix" "array"

> datos<-data.frame(ID=c("a1", "a2", "a3"), edad=c(15,22,7))
> str(datos)        $ ID  : chr  "a1" "a2" "a3"
                    $ edad: num  15 22 7

> dim(datos) #[1] 3 2
> class(datos) #[1] "data.frame"
```

Cambio del tipo de vector (variable):

```
> x <- 1:15

> class(x) # [1] "integer"

> as.factor(x)
# [1] 1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
# Levels: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

> as.numeric(x); as.character(x)
# [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
# [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11"
"12" "13" "14" "15"

> as.numeric(as.character(x))
# [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
```

Introducción a R

4. Gráficos y Funciones: *Gráficos*

- R incluye muchas y variadas funciones para hacer gráficos.
- El sistema permite desde gráficos muy simples a figuras de calidad para incluir en artículos y libros.
- Sólo examinaremos la superficie. Más detalles en días sucesivos y en el libro *R Graphics* de Paul Murrell por ejemplo.
- También podemos ver un buen conjunto de ejemplos con **demo**(graphics).
- El comando **plot** es uno de los más utilizados para realizar gráficos.

Introducción a R

5. Librerías

- Tipos
- Instalar; Cargar
- Comandos

Introducción a R

5. Librerías: *Tipos*

A) POR DEFECTO (YA INSTALADOS con el software básico): (ej. stats).

No hay que instalarlas ni cargarlas

> library()

B) ADICIONALES (NO INSTALADOS con el software básico): (ej. maptools).

Hay que instalarlas

Hay que cargarlas en cada nueva sesión.

Introducción a R

5. Librerías: *Instalar; Cargar*

- Repaso de los conceptos de Instalar y Cargar librerías para poder utilizar las funciones que contienen estas.

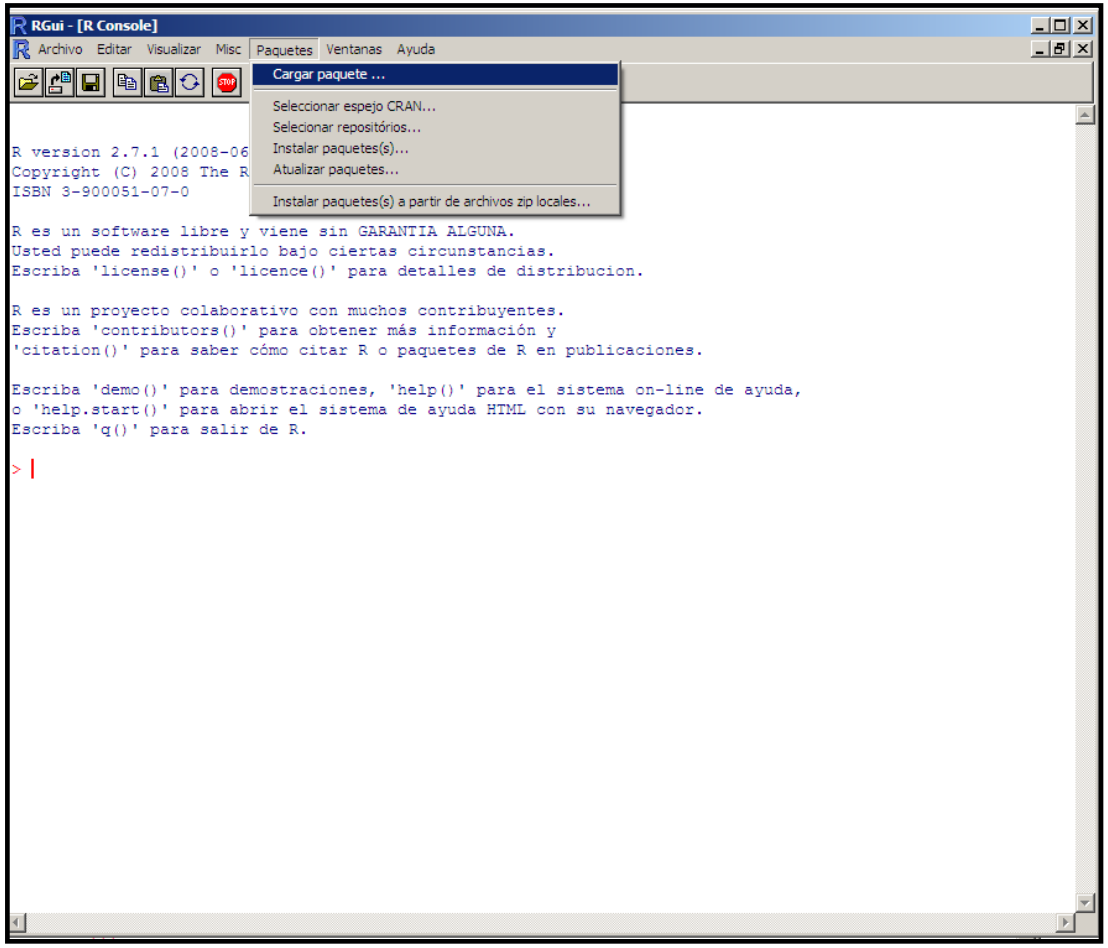
- Cuidado con la instalación de librerías que ya estén instaladas (versión más reciente)

- Instalación de librerías CRAN (ej. gdata) y NO CRAN!!!
  - https://cran.r-project.org/ (Software/Packages)
  - Fuera del cran, por ejemplo: https://www.bioconductor.org/install/

- Instalación de librerías en formato zip, tar.gz, etc.

Introducción a R

5. Librerías: *Comandos*



INSTALACIÓN:

install.packages()

CARGA:

library()

Introducción a R

6. Importación:

Funciones generales

> read.table

> read.csv

Funciones especiales

> read.xlsx de la librería "**openxlsx**"

> read.dta13 de la librería "**readstata13**"

Formato R

Formato: RData o Rdata

> load(file="unos.datos.salvados.previamente.RData")

Built-in data

> data(package = "plyr")

> library("plyr")

> **data**(ozone)

Introducción a R

7. Data.frame

- Un data frame está compuesto por vectores.
- Pueden contener datos de diferentes tipos.

- **Creacción de un data.frame:**

```
> my.data.frame<-data.frame(ID=c(”paciente1",”paciente2",”paciente3"),
edad=c(10,25,33),altura=c(NA,150,180), cancer=c(TRUE,TRUE,FALSE),
bmi=c(22,25,20))
```

- **Carga de un data.frame:**

```
> load(“datos.curso1.RData”)
```

Introducción a R

7. Data.frame

Visualización de atributos y contenido del data.frame:

```
> str(datos)
'data.frame':200 obs. of 11 variables:
$ ID : num 137 174 200 23 39 90 40 115 72 27 ...
$ edad : num 37 85 29 13 49 12 85 31 39 70 ...
$ sexo : chr "Mujer" "Mujer" "Hombre" "Hombre" ...
$ estado.civil : chr "Casado" "Soltero" "Casado"
"Divorciado" ...
$ nivel.estudios : chr "Bajo" "Alto" "Bajo" "Alto" ...
$ peso : num 59.6 60 79.2 80.8 80.8 ...
$ altura : num 151 149 169 171 171 ...
$ fumador : chr "No" "No" "No" "Si" ...
$ diabetes : chr "No" "Si" "Si" "Si" ...
$ cancer.mama : chr "Si" "No" "Si" NA ...
$ cancer.prostata: chr NA NA "Si" "Si" ...
```

> datos

	ID	edad	sexo	estado.civil	nivel.estudios	peso	altura	fumador	diabetes	cancer.mama	cancer.prostata
1	137	37	Mujer	Casado	Bajo	59.58221	150.7163	No	No	Si	<NA>
2	174	85	Mujer	Soltero	Alto	59.95427	149.2075	No	Si	No	<NA>
3	200	29	Hombre	Casado	Bajo	79.20674	168.9795	No	Si	Si	Si
4	23	13	Hombre	Divorciado	Alto	80.78347	171.1568	Si	Si	<NA>	Si
5	39	49	Hombre	Divorciado	Bajo	80.76036	170.5682	Si	Si	<NA>	No
6	90	12	Hombre	Casado	Alto	79.83426	170.8565	No	No	<NA>	Si
7	40	85	Hombre	Casado	Alto	80.69636	168.5586	No	No	<NA>	Si
8	115	31	Mujer	Soltero	Alto	61.28985	150.0667	Si	No	Si	<NA>
9	72	39	Mujer	Divorciado	Bajo	60.70871	150.4091	Si	Si	Si	<NA>
10	27	70	Hombre	Soltero	Medio	78.89874	167.8307	No	Si	<NA>	Si
11	19	24	Mujer	Divorciado	Alto	60.06984	149.5328	No	No	Si	<NA>
12	133	45	Hombre	Soltero	Medio	79.57263	170.7013	Si	Si	<NA>	Si
13	15	42	Mujer	Soltero	Bajo	60.39387	150.7226	Si	No	Si	<NA>
14	44	74	Mujer	Casado	Medio	60.33449	148.2137	No	Si	Si	<NA>
15	179	16	Mujer	Divorciado	Medio	60.01237	151.4407	Si	Si	Si	<NA>
16	148	6	Mujer	Soltero	Bajo	57.93049	149.7306	No	No	Si	<NA>
17	192	31	Mujer	Casado	Bajo	57.99527	148.3326	Si	Si	Si	<NA>
18	186	83	Mujer	Soltero	Bajo	60.44401	150.3675	Si	Si	Si	<NA>
19	18	32	Hombre	Divorciado	Alto	79.39825	168.9841	Si	No	<NA>	No
20	106	67	Hombre	Soltero	Medio	80.07338	170.6679	No	Si	<NA>	Si
21	86	54	Mujer	Divorciado	Medio	60.02079	150.1170	Si	Si	No	<NA>
22	55	18	Hombre	Soltero	Bajo	78.30574	170.9642	No	Si	<NA>	Si
23	20	84	Hombre	Casado	Bajo	80.35391	168.2667	Si	Si	<NA>	No
24	102	40	Mujer	Soltero	Bajo	60.34935	150.3949	Si	No	No	<NA>
25	140	6	Hombre	Divorciado	Bajo	79.47181	170.0953	Si	No	<NA>	Si
26	112	63	Mujer	Soltero	Medio	60.05865	150.2747	No	Si	No	<NA>
27	183	79	Hombre	Soltero	Alto	79.69627	169.4480	Si	Si	<NA>	Si
28	120	27	Mujer	Divorciado	Alto	58.60144	150.2412	No	Si	Si	<NA>
29	117	34	Hombre	Soltero	Alto	80.76927	168.8787	No	No	<NA>	Si
30	130	53	Hombre	Divorciado	Alto	79.05485	170.1690	Si	No	<NA>	Si
31	144	23	Mujer	Divorciado	Alto	59.16448	149.9304	No	No	Si	<NA>
32	41	66	Hombre	Casado	Medio	79.35361	170.4784	Si	No	<NA>	No
33	193	29	Mujer	Casado	Alto	58.42305	149.1501	Si	No	Si	<NA>
34	12	54	Hombre	Soltero	Medio	78.61265	169.0950	Si	Si	<NA>	Si
35	108	32	Mujer	Casado	Bajo	60.56363	150.0661	No	Si	Si	<NA>
36	17	7	Mujer	Soltero	Bajo	58.65925	148.7323	Si	Si	Si	<NA>
37	157	42	Hombre	Soltero	Bajo	79.90425	169.8202	No	Si	<NA>	Si
38	122	68	Mujer	Casado	Alto	61.06287	149.0318	No	No	Si	<NA>
39	25	15	Hombre	Soltero	Alto	81.42743	169.8150	No	Si	<NA>	Si
40	50	6	Hombre	Casado	Bajo	80.68350	168.3866	Si	No	<NA>	Si
41	146	38	Mujer	Soltero	Bajo	58.74115	150.0431	No	Si	Si	<NA>
42	29	28	Hombre	Casado	Medio	80.60408	170.8330	Si	No	<NA>	Si
43	187	27	Mujer	Soltero	Alto	60.28175	149.2774	No	No	Si	<NA>

Introducción a R

7. Data.frame

> fix(datos)

R Data Editor

	ID	edad	sexo	estado.civil	nivel.estudios	peso	altura	fumador	diabetes	cancer.mama	cancer.prostata
1	137	37	Mujer	Casado	Bajo	59.58221	150.7163	No	No	Si	
2	174	85	Mujer	Soltero	Alto	59.95427	149.2075	No	Si	No	
3	200	29	Hombre	Casado	Bajo	79.20674	168.9795	No	Si	Si	Si
4	23	13	Hombre	Divorciado	Alto	80.78347	171.1568	Si	Si		Si
5	39	49	Hombre	Divorciado	Bajo	80.76036	170.5682	Si	Si		No
6	90	12	Hombre	Casado	Alto	79.83426	170.8565	No	No		Si
7	40	85	Hombre	Casado	Alto	80.69636	168.5586	No	No		Si
8	115	31	Mujer	Soltero	Alto	61.28985	150.0667	Si	No	Si	
9	72	39	Mujer	Divorciado	Bajo	60.70871	150.4091	Si	Si	Si	
10	27	70	Hombre	Soltero	Medio	78.89874	167.8307	No	Si		Si
11	19	24	Mujer	Divorciado	Alto	60.06984	149.5788	No	No	Si	
12	133	45	Hombre	Soltero	Medio	79.57263	170.7413	Si	Si		Si
13	15	42	Mujer	Soltero	Bajo	60.39387	150.2226	Si	No	Si	
14	44	74	Mujer	Casado	Medio	60.73149	148.2137	No	Si	Si	
15	179	16	Mujer	Divorciado	Medio	60.10237	151.4407	Si	Si	Si	
16	148	6	Mujer	Soltero	Bajo	59.93049	149.7306	No	No	Si	
17	192	31	Mujer	Casado	Bajo	57.99527	148.3326	Si	Si	Si	
18	186	83	Mujer	Soltero	Bajo	60.44401	150.3675	Si	Si	Si	
19	18	32	Hombre	Divorciado	Alto	79.39825	168.9841	Si	No		No
20	106	67	Hombre	Soltero	Medio	80.07338	170.6679	No	Si		Si
21	86	54	Mujer	Divorciado	Medio	60.02079	150.117	Si	Si	No	
22	55	18	Hombre	Soltero	Bajo	78.30574	170.9642	No	Si		Si
23	20	84	Hombre	Casado	Bajo	80.35391	168.2667	Si	Si		No
24	102	40	Mujer	Soltero	Bajo	60.34935	150.3949	Si	No	No	
25	140	6	Hombre	Divorciado	Bajo	79.47181	170.0953	Si	No		Si
26	112	63	Mujer	Soltero	Medio	60.05865	150.2747	No	Si	No	
27	183	79	Hombre	Soltero	Alto	79.69627	169.448	Si	Si		Si
28	120	27	Mujer	Divorciado	Alto	58.60144	150.2412	No	Si	Si	

Cuidado con "fix"

> head(datos,3); tail (datos,3)

```
> head(datos,3)
   ID edad   sexo estado.civil nivel.estudios     peso   altura fumador diabetes cancer.mama cancer.prostata
1 137   37  Mujer       Casado           Bajo 59.58221 150.7163      No       No          Si             <NA>
2 174   85  Mujer      Soltero           Alto 59.95427 149.2075      No       Si          No             <NA>
3 200   29 Hombre       Casado           Bajo 79.20674 168.9795      No       Si          Si               Si
> tail(datos,3)
     ID edad   sexo estado.civil nivel.estudios     peso   altura fumador diabetes cancer.mama cancer.prostata
198  57   41 Hombre      Soltero          Medio 79.45077 168.4157      No       No          <NA>             No
199 169   44  Mujer      Soltero           Bajo 59.69171 149.0113      Si       No            No           <NA>
200 143   54  Mujer   Divorciado          Medio 61.02373 150.8797      Si       Si            Si           <NA>
```

Introducción a R

7. Data.frame: Acceso a elementos

a) A una variable: nombre.data.frame$”nombre.variable”

> datos$”sexo”; sexo.mirar<-datos$”sexo”

b) Notacion matricial (Nombre.data.frame[Filas,Columnas])

> datos[,3];datos[1:3,3];datos[1:10,1:3]

c) Tanto en Filas como en Columnas: vector de índices (posiciones), vector lógico, condición lógica sobre otros vectores (variables), etc

Ejemplos:

> datos[datos$”sexo”==“Mujer”,]; datos[datos$”sexo”==“Mujer”&datos$”estado.civil”==“Casado”,1:10]
> datos[datos$”sexo”%in%“Mujer”&datos$”estado.civil”%in%“Casado”,1:10];
> datos.new<-datos[datos$”sexo”==“Mujer”,]
> datos.new2<-datos; datos.new2[1,3]<-"Hombre”
> datos.new2[datos.new2$”ID”==200,sexo]<-"Mujer"
> datos.new2$”sexo”[datos.new2$”ID”==200]<-"Mujer"

Introducción a R

7. Data.frame

Añadir variables (columnas):

a) nombre.data.frame$”nombre.nueva.variable”<-valores

```
> datos$”caso.cancer” <-c(1,1,1…....)
```

b) Utilizar la funcion cbind()

```
> datos.nuevos1 <-datos[ ,1:3]
> datos.union1 <-cbind(datos,datos.nuevos1)
```

Añadir registros (filas):

a) Utilizar la funcion rbind()

```
> datos.nuevos2 <-datos[1:5, ]
> datos.union2 <-rbind(datos,datos.nuevos2)
```

Introducción a R

7. Data.frame

Eliminar variables o registros:

a) nombre.data.frame [-vector de posiciones de las filas a eliminar, -vector de posiciones de las columnas a eliminar]

```
> datos.nuevos3 <-datos[-c(2,3), ]; datos.nuevos4 <-datos[-c(2,3) , -c(14,18)]
```

b) nombre.data.frame [! condiciones lógicas, ! Condiciones logicas]

```
> indice5 <-datos$”sexo”!="Mujer”; datos.nuevos5 <-datos[indice5, ]
> datos.nuevos6 <-datos[ , names(datos)!="peso"]
> indice7 <-which(datos$”sexo”=="Mujer"); datos.nuevos7 <-datos[-indice7, ]
```

Modificación de contenido del data.frame:

a) Missing

```
>datos$cancer.mama[is.na(datos$cancer.mama)]<-”No”
```

b) Condiciones lógicas
c) Operaciones aritméticas

Introducción a R

7. Data.frame

Otra forma de obtener subsets de data frames (sin incluir los NA):

```
> datos.mujer1 <-datos[datos$”sexo”%in%"Mujer", -c(6,7) ]
> datos.mujer2 <-subset(datos, sexo=="Mujer",select=-c(peso, altura))
```

	ID	edad	sexo	estado.civil	nivel.estudios	fumador	diabetes	cancer.mama	cancer.prostata
2	174	85	Mujer	Soltero	Alto	No	Si	No	<NA>
8	115	31	Mujer	Soltero	Alto	Si	No	Si	<NA>
9	72	39	Mujer	Divorciado	Bajo	Si	Si	Si	<NA>
11	19	24	Mujer	Divorciado	Alto	No	No	Si	<NA>
13	15	42	Mujer	Soltero	Bajo	Si	No	Si	<NA>
14	44	74	Mujer	Casado	Medio	No	Si	Si	<NA>

Introducción a R

7. Data.frame

Observaciones generales

```
> names(datos)
[1] "ID" "edad" "sexo" "estado.civil" "nivel.estudios" "peso"
[7] "altura" "fumador" "diabetes" "cancer.mama" "cancer.prostata”
> names(datos)[3]
[1] "sexo”
> names(datos)[3]<-"sex”
> nombres.variables<-names(datos)
> row.names(datos)
[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12" "13" "14" "15" "16" ...
> mujeres<-datos[datos$sexo%in%"Mujer",]
> row.names(mujeres)
[1] "1" "2" "8" "9" "11" "13" "14" "15" "16" "17“...
> row.names(mujeres)<-NULL
> row.names(mujeres)
[1] "1" "2" "3" "4" "5" "6" "7" "8" "9"
```

Introducción a R

8. Operaciones aritméticas

- suma +, resta -, multiplicación *, división /, potencia ^, raíz cuadrada sqrt
- %/%división entera, %%módulo: resto de la división entera
- logaritmos log, log10, log2, logb(x, base), exponencial exp
- trigonométricas sin, cos, tan, asin, acos, atan

- otras:
max, min, range, pmax, pmin, mean, median, var, sd, quantile, sum, prod, diff, cumsum, cumprod, cummax, cummin.

***FUNDAMENTAL*:** R puede operar *sobre vectores enteros de un golpe.*

```
>x<-c(1:10); y<-x/2; z<-x^2; w<-y+z
```

• Si un elemento es más corto, se "recicla". Es "intuitivo" si la operación es escalar con vector. Pero también ocurre en operaciones entre vectores.

```
>x+15; x2<-c(1:5)
>x+x2
```

• El reciclado cuando (vector,escalar) suele ser lo que queremos. Entre vectores no siempre: cuidado. (R da un warning, de momento – S4 clases dan un error).

Introducción a R

8. Operaciones lógicas

• <, >, <=, >=, ==, !=

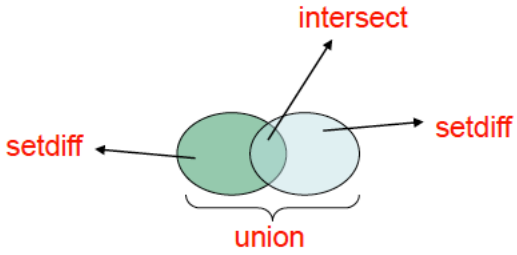
• !, &, |, xor() y los parecidos &&, ||

```
x <-5; x < 5; x >= 5; x == 6; x != 5
y <-c(TRUE, FALSE); !y; z <-c(TRUE, TRUE)
xor(y, z)
y & z; y | z
```

• Las formas &&, || son "short-circuit operators" que se suelen usar dentro de if statements. Se aplican a vectores de longitud uno y sólo evalúan el segundo argumento si es preciso.

Operaciones de conjuntos

> x <- c(1:5); y <- c(1, 3, 7:10)
> union(x, y)
> intersect(x, y)
> setdiff(y, x)



> v <- c("bcA1", "bcA2", "blX1")
> w <- c("bcA2", "xA3")
> unión (v, w)
> intersect (v, w)
> setdiff (w, v)
> setdiff (v, w)

Introducción a R

9. Missing

- NA es el código de “Not available”.

```
> v <-c(1,6,9,NA)
> is.na(v); which(is.na(v))
> w <-v[!is.na(v)] # sin los valores perdidos
> v == NA # !No funciona!
```

- Sustituir NA por, p.ej., 0:

```
> v[is.na(v)] <-0
```

- El infinito y NaN (“not a number”) son diferentes de NA.

```
> is.infinite(-5/0); is.nan(0/0); is.na(5/0)
```

- Con algunas funciones

```
> xna <- c(1, 2, 3, NA, 4); mean(xna) ## Esto da “NA”
> mean(xna, na.rm = TRUE)
```

Introducción a R

9. Missing

is.na()

```
> sum(is.na(datos$sex))
[1] 0

> datos.con.missing<-datos

> datos.con.missing$peso[datos.con.missing$”peso”<60]<-NA

> sum(is.na(datos.con.missing$”peso”))
[1] 51

summary(datos.con.missing)
```

```
> summary(datos.con.missing)
       ID              edad            sexo           estado.civil       nivel.estudios         peso            altura     
 Min.   :  1.00   Min.   : 1.00   Length:200         Length:200         Length:200         Min.   :60.01   Min.   :147.8  
 1st Qu.: 50.75   1st Qu.:21.00   Class :character   Class :character   Class :character   1st Qu.:61.17   1st Qu.:150.1  
 Median :100.50   Median :42.00   Mode  :character   Mode  :character   Mode  :character   Median :79.39   Median :159.8  
 Mean   :100.50   Mean   :42.17                                                           Mean   :73.67   Mean   :159.8  
 3rd Qu.:150.25   3rd Qu.:63.00                                                           3rd Qu.:80.25   3rd Qu.:169.7  
 Max.   :200.00   Max.   :85.00                                                           Max.   :82.28   Max.   :172.9  
                                                                                          NA's   :51                     
   fumador            diabetes         cancer.mama        cancer.prostata   
 Length:200         Length:200         Length:200         Length:200         
 Class :character   Class :character   Class :character   Class :character   
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   
```

Introducción a R

Modificación de contenido del data.frame:

a) Recodificación Missing

```
>datos$cancer.mama[is.na(datos$cancer.mama)]<-”No”
```

b) Condiciones lógicas

```
>datos$obeso<-”No”
>datos$obeso[datos$peso>15]<-”Si”
```

c) Operaciones aritméticas

```
>datos$ratio<-datos$altura/datos$edad
```

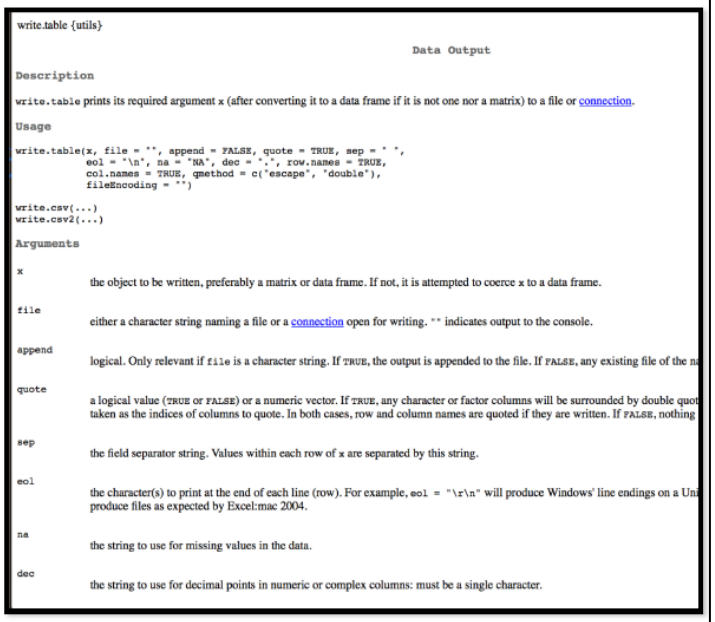
Introducción a R

10. Exportación: *Funciones generales*

> write.table

> wirte.csv

> ?write.table



write.table {utils}

Data Output

Description

`write.table` prints its required argument `x` (after converting it to a data frame if it is not one nor a matrix) to a file or connection.

Usage

```
write.table(x, file = "", append = FALSE, quote = TRUE, sep = " ",
            eol = "\n", na = "NA", dec = ".", row.names = TRUE,
            col.names = TRUE, qmethod = c("escape", "double"),
            fileEncoding = "")

write.csv(...)
write.csv2(...)
```

Arguments

`x`	the object to be written, preferably a matrix or data frame. If not, it is attempted to coerce `x` to a data frame.
`file`	either a character string naming a file or a connection open for writing. `""` indicates output to the console.
`append`	logical. Only relevant if `file` is a character string. If `TRUE`, the output is appended to the file. If `FALSE`, any existing file of the n
`quote`	a logical value (`TRUE` or `FALSE`) or a numeric vector. If `TRUE`, any character or factor columns will be surrounded by double quot taken as the indices of columns to quote. In both cases, row and column names are quoted if they are written. If `FALSE`, nothing
`sep`	the field separator string. Values within each row of `x` are separated by this string.
`eol`	the character(s) to print at the end of each line (row). For example, `eol = "\r\n"` will produce Windows' line endings on a Uni produce files as expected by Excel:mac 2004.
`na`	the string to use for missing values in the data.
`dec`	the string to use for decimal points in numeric or complex columns: must be a single character.

Ejemplo:

> data(airquiality)

> head(airquality)

Ozone Solar.R Wind Temp Month Day
1 41 190 7.4 67 5 1
2 36 118 8.0 72 5 2
3 12 149 12.6 74 5 3
4 18 313 11.5 62 5 4
5 NA NA 14.3 56 5 5
6 28 NA 14.9 66 5 6

> write.table(x=airquality, file="aiquality.exportado.txt",sep="\t",row.names=F,quote=F)

Introducción a R

10. Exportación:

Funciones especiales

```
> library("readstata13")
> save.dta13(mydata, "c:/mydata.dta")

> library(”openxlsx")
> write.xlsx(mydata, "c:/mydata.xlsx")
```

Formato R

- Formato: Rdata o RData

- Guardar datos, funciones, etc, para ser usados en otras sesiones de R de cualquier sistema operativo.

```
> a1 <- rnorm(10); a2 <- 1:10; a3 <- letters[10:20]
> save(a1, a2, file="unos.datos.guardados.RData")
```

- Podemos salvar todos los objetos con:

```
> save.image() # salvado como ".RData"
> save.image(file = "un.nombre.RData")
```

EJERCICIOS

"ejercicios.introduccion_R_a_realizar.pdf"