

Índice general

1	El programa R: Generalidades	11
1	Entorno de trabajo y carga de módulos de R	13
1.1	Introducción	13
1.2	R-Commander	13
1.3	Ayuda y comentarios	15
1.4	Paquetes	17
1.5	Documentación	18
1.6	Usar directorios	19
2	El R como calculadora	19
3	Estructuras de datos en R: vectores, factores, matrices y arrays, listas y data frames	21
3.1	Objetos en R	21
3.2	Vectores	26
3.3	Matrices	35
4	Lectura y escritura de datos	37
4.1	Lectura de datos	37
4.1.1	De un fichero	37
4.1.2	La función scan()	38
4.1.3	Leer datos de librerías	38
4.1.4	Acceder a datos de una biblioteca	39
4.2	Exportar datos	40

4.3	Guardar y leer datos	41
2	Organización y manipulación de datos	43
1	Generalidades	45
1.1	El análisis de datos con ordenador: proceso . . .	45
2	La matriz de datos	46
2.1	Introducir o importar los datos	47
2.1.1	Introducir los datos	50
2.1.2	Renombrar las variables	51
2.1.3	Convertir variable numérica en factor .	52
2.1.4	Importación de datos	54
2.2	Filtrado y depuración de la matriz de datos . . .	58
2.2.1	Casos aislados	58
2.2.2	Casos perdidos	58
2.2.3	Datos atípicos	59
2.2.4	Detección de valores atípicos. Diagrama de caja	60
3	Selección de un procedimiento. Análisis descriptivo de datos	63
3	Representación de la información	69
1	Generalidades	71
2	Tipos de datos	72
3	Gráficos para datos cualitativos o discretos	72
3.1	Diagrama de barras	72
3.2	Diagrama de sectores	76
3.3	Pictogramas	80
4	Gráficos para datos continuos	81
4.1	Diagrama diferencial o histograma	81
4.2	Estimar la densidad de la distribución de una va- riable	86

4.3	Gráfico de tallos y hojas	89
4.4	Gráfico de caja (Box Plot)	92
4.5	Gráfica de comparación de cuantiles	98
5	Superposición y salvado de gráficas	101
5.1	Superposición de gráficas	101
5.2	Salvado de gráficas	104
4	Relación entre variables	107
1	Medidas de proximidad	110
2	Gráficas de relación	110
2.1	Gráficos de dispersión simple	110
2.2	Gráfico de dispersión con curvas de ajuste	111
2.3	Diagrama de dispersión matricial	113
2.4	Gráfica de dispersión cuando hay muchos datos .	116
2.5	Gráficos de dispersión en 3D	119
3	Medidas de similaridad	122
3.1	Coeeficientes de correlación	122
3.1.1	Coeeficiente de correlación de Pearson .	124
3.1.2	El coeeficiente de determinación	127
3.2	Asociación para variables ordinales	128
3.2.1	El coeeficiente de correlación de rangos de Spearman: r_s	128
3.2.2	El coeeficiente de correlación de rango de Kendall: τ	130
3.2.3	El coeeficiente de concordancia de Kendall:W	133
3.2.4	Índice de concordancia de kappa	136
3.3	Medidas de asociación en tablas de contingencia	140
3.3.1	El coeeficiente de contingencia C y la V de Cramer	140

	3.3.2	El coeficiente $\phi(\phi)$	142
4		Coeficientes de disimilaridades (distancias)	144
	4.1	Para datos en intervalo	145
		4.1.1 La distancia euclídea	145
		4.1.2 La distancia de Mahalanobis	145
		4.1.3 Otras distancias para medidas en intervalo	145
	4.2	Para datos categóricos no dicotómicos	149
		4.2.1 Distancia ji-cuadrado	149
		4.2.2 Distancia phi-cuadrado	149
	4.3	Para datos binarios	150
		4.3.1 Distancia euclídea binaria	150
		4.3.2 Distancia euclídea binaria al cuadrado .	151
		4.3.3 Diferencia de tamaño	151
		4.3.4 Diferencia de configuración	151
		4.3.5 Varianza	152
		4.3.6 Forma	152
		4.3.7 Lance y Williams	152
5		Análisis de contenido textual	155
	1	Introducción	157
	2	Tipos de documentos	158
	3	Procedimiento a seguir en el análisis de contenido	158
	4	Análisis de contenido con ayuda de ordenador	160
		4.1 Análisis de contenido con R-Commander	160
	5	Ejemplo de análisis de contenido con R-Commander . . .	162
		5.1 Carga de los datos y filtrado	162
		5.2 Diccionario de términos	166
		5.3 Análisis descriptivo	166

5.4	Análisis de correspondencias	173
5.5	Análisis de agrupamiento o cluster	177
Bibliografía		183

Capítulo 1

El programa R: Generalidades

1. Entorno de trabajo y carga de módulos de R
2. El R como calculadora
3. Estructuras de datos en R: vectores, factores, matrices y arrays, listas y data frames
4. Lectura y escritura de datos

Objetivos

- Conocer un entorno de recogida y análisis de datos
- Saber introducir e importar datos
- Diseñar un proceso de adaptación de la fuente de información a datos idóneos para análisis estadístico

1. Entorno de trabajo y carga de módulos de R

1.1. Introducción

R es un software estadístico-matemático orientado al procesamiento y análisis de datos, con un entorno gráfico de trabajo. Se pueden destacar las siguientes componentes:

- Un sistema de manejo y almacenamiento de datos eficiente.
- Funciones matemáticas para el proceso de vectores y matrices.
- Una gran cantidad de funciones estadísticas integradas en el sistema.
- Una amplia variedad de librerías especializadas para estadística y análisis de datos.
- Funciones de procesamiento y análisis gráfico orientadas al análisis de datos.
- Un lenguaje de programación completo basado en el lenguaje S.

El lenguaje de programación R es un dialecto del lenguaje S, desarrollado en AT&T por Becker, Chambers y Wilks. En general, cualquier manual de aprendizaje de S o SPlus puede servir para aprender a programar en R.

R se obtiene de forma gratuita en <http://www.r-project.org/>. En dicha dirección se pueden encontrar muy distintos materiales de apoyo.

1.2. R-Commander

R-Commander es una Interfaz Gráfica de Usuario (GUI en inglés), creada por John Fox, que mediante menús facilita la introducción, manipulación y tratamiento de datos.

Para poder ejecutar R-Commander se necesita instalar R y la librería **Rcmdr**.

-Para descargar e instalar R y el paquete **Rcmdr**:

1. Descargar R de la dirección <http://cran.es.r-project.org> e instalar R
2. Dar Instalar paquete(s) y escribir **Rcmdr**
3. Dar Cargar paquete y seleccionar **Rcmdr**
4. Escribir en la línea de comandos

```
>library(Rcmdr)
```

- En sucesivas ocasiones para entrar en el Rcmdr (R Commander) se puede optar por:

1. Escribir en la línea de comandos

```
>library(Rcmdr)
```

2. Modificar el archivo Rprofile.site situado en c:/Archivos de programas/R/etc/Rprofile.site y añadir `library(Rcmdr)` salvar el fichero y la próxima vez ya se entra directamente en R-Commander

El aspecto de la pantalla de R-Commander se puede observar en la figura.

1.1

Mediante los menús de R-Commander se pueden realizar las operaciones básicas más frecuentes, en otros casos se puede utilizar directamente la consola de R u otros editores, entre los que destaca por sus facilidades y desarrollo de herramientas auxiliares del mismo, RStudio <https://www.rstudio.com/>.

Como verá el lector, todas las operaciones que se realicen en el menú de RCommander se reflejarán en comandos de R que se mostrarán en la *ventana de instrucciones*, de tal manera que al final de los cálculos se podrá salvar el fichero de instrucciones, que permitirá realizar de nuevo el proceso para los mismos u otros datos similares. Para ejecutar un comando de R se seleccionará éste con el ratón y se pulsará el botón de ejecutar. El resultado de la ejecución del comando se reflejará en la *ventana de resultados*. Además, si se produce algún error en la ejecución de un comando se mostrará en la *ventana de Mensajes*.

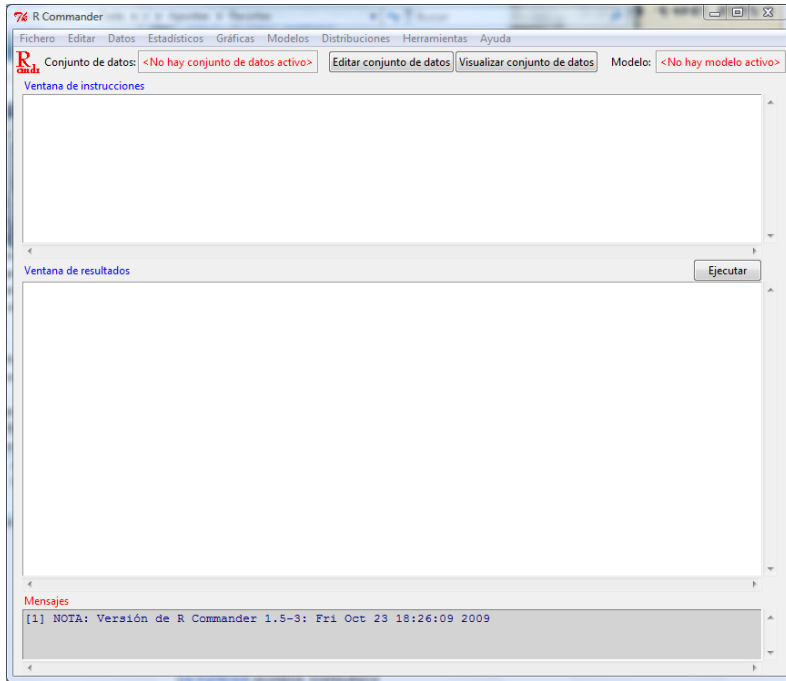


Figura 1.1: Pantalla de R-Commander

Junto con R-Commander se pueden cargar paquetes adicionales (*plugins*) que facilitan la realización de algunos análisis suplementarios, entre los cuales uno de los más utilizados en ciencias sociales es el relativo al análisis de contenido. Para cargar un plugins se debe elegir en el menú de R-Commander: **Herramientas -> Cargar plugins de R Commander** y seleccionar los paquetes. Después se pedirá reiniciar R-Commander, tras lo cual estarán disponibles los paquetes seleccionados.

1.3. Ayuda y comentarios

Existen diversos procedimientos de ayuda:

Para encontrar todos los comandos que contengan un determinado término:

```
apropos ("normal")
```

Para saber más de la sintaxis de un comando:

```
help ("if")
```

Dos páginas con manuales y referencias:

- Manuales de R: <https://cran.r-project.org/manuals.html>
- Quick-R: <https://www.statmethods.net/>

```
?mean
help(package = 'dplyr') # de un paquete
```

Para poner un comentario aclaratorio en alguna instrucción o comando se utiliza el símbolo `#` que hace invisible al programa el texto colocado a la derecha del símbolo.

```
data(cars, package = "datasets") # maneja la data: cars
cars # presenta en consola los datos de la data cars
```

```
##      speed dist
## 1         4    2
## 2         4   10
## 3         7    4
## 4         7   22
## 5         8   16
## 6         9   10
## 7        10   18
## 8        10   26
## 9        10   34
## 10       11   17
## 11       11   28
## 12       12   14
## 13       12   20
## 14       12   24
## 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
## 31	17	50
## 32	18	42
## 33	18	56
## 34	18	76
## 35	18	84
## 36	19	36
## 37	19	46
## 38	19	68
## 39	20	32
## 40	20	48
## 41	20	52
## 42	20	56
## 43	20	64
## 44	22	66
## 45	23	54
## 46	24	70
## 47	24	92
## 48	24	93
## 49	24	120
## 50	25	85

1.4. Paquetes

R consta de un sistema base y un conjunto de paquetes adicionales que permiten una mayor funcionalidad.

Tipos de paquetes: Los que forman parte del sistema base (ej. ctest).

Los que no son parte del sistema base, pero son *recommended* (ej., *survival*, *nlme*). En GNU/Linux y Windows ya (desde 1.6.0) forman parte de la distribución estándar.

Otros paquetes; ej., *rpart*, *rweka*, etc. han de instalarse individualmente. Más adelante se verá cómo hacerlo.

Se pueden instalar paquetes adicionales desde el propio programa escribiendo:

```
install.packages(dplyr) # para instalar un paquete (dplyr)
```

donde entre los paréntesis se pondrá el nombre del paquete requerido. También se puede realizar la carga de un paquete con la opción *Paquetes* del menú del programa.

Para usar un determinado módulo se escribirá:

```
library(dplyr) # para usar una librería
dplyr::select #para usar una función (select) que está
              #en el paquete dplyr
data(iris) # usar el fichero de datos iris
```

1.5. Documentación

Los manuales de R, incluidos en todas las instalaciones en (*C:/Program Files/R/R-3.5.1/doc/manual*) para la versión 3.5.1 del programa en este caso, son:

- An introduction to R. (De lectura obligada)
- Writing R Extensions.
- R Data Import/Export.
- R: A Language and Environment for Statistical Computing.
- R Installation and Administration.
- R Internals.

- R Language Definition.

Además de los múltiples documentos en inglés, recomendamos consultar: <http://cran.r-project.org/other-docs.html#nenglish> donde se pueden encontrar muy distintos materiales de apoyo.

1.6. Usar directorios

Para localizar el directorio de trabajo o situarnos en un determinado directorio:

```
getwd () #Para encontrar el directorio de trabajo actual
setwd ('C:..../ruta') #Fija un directorio de trabajo
```

2. El R como calculadora

R puede ser utilizado como una calculadora de modo interactivo. Por ser los comienzos, en este apartado se muestra algunos ejemplos sencillos.

Ejemplo de comando completo en R :

```
2+2
```

```
## [1] 4
```

Escribimos 2+2 y damos enter

[1] 4 Sale el resultado. [1] indica que es el primer (y único resultado) de nuestra orden. Al terminar el comando, si estamos en la consola de R, el sistema vuelve a presentar su indicador >.

Otros ejemplos:

```
2*5 #Multiplicación de dos números
```

```
## [1] 10
```

```
5/2 #División real de dos números
```

```
## [1] 2.5
```

```
5%/%2 #División entera: se devuelve la parte entera solamente
```

```
## [1] 2
```

```
5%%2 #Módulo: resto de dividir un número por otro
```

```
## [1] 1
```

```
5^2 #Potenciación
```

```
## [1] 25
```

```
5^2.3 #Potenciación, exponente real
```

```
## [1] 40.51641
```

```
exp(1) #El número e
```

```
## [1] 2.718282
```

```
sqrt(2) #La raíz cuadrada de 2
```

```
## [1] 1.414214
```

```
log(3) #El logaritmo neperiano de 3
```

```
## [1] 1.098612
```

```
log(3,10) #El logaritmo de 3 en base 10
```

```
## [1] 0.4771213
```

```
abs(-3.4) #Valor absoluto de un número
```

```
## [1] 3.4
```

```
pi #El número pi
```

```
## [1] 3.141593
```

Para averiguar algo más sobre la organización de las funciones matemáticas en R, podemos probar las siguientes órdenes:

```
help("Math")
```

```
help("Special")
```

3. Estructuras de datos en R: vectores, factores, matrices y arrays, listas y data frames

3.1. Objetos en R

Como R es un lenguaje que maneja objetos, casi todo en R es un objeto, esto significa que las variables, datos, funciones, resultados, etc., se guardan en la memoria activa del ordenador en forma de *objetos* con un *nombre* específico. El usuario puede modificar o manipular estos objetos con *operadores* (aritméticos, lógicos, y comparativos) y *funciones* (que a su vez son objetos).

Tipos de objetos:

- Objetos del lenguaje: llamadas, expresiones, nombres
- Funciones: constan de lista de argumentos, código, entorno
- Sin objeto: NULL

Manipulación de los objetos de la memoria:

`ls()` lista los objetos de ejemplo:

```
name<- "Juan"; n1<- 5; n2<- 50; p <- 0.5
ls()
```

```
## [1] "cars" "n1"   "n2"   "name" "p"
```

pat si se quiere listar los objetos que contengan un carácter en particular.

Por ejemplo:

```
ls(pat="n")
```

```
## [1] "n1"   "n2"   "name"
```

La función `ls.str()` muestra algunos detalles de los objetos en memoria:

```
ls.str()
```

```
## cars : 'data.frame': 50 obs. of  2 variables:
## $ speed: num  4 4 7 7 8 9 10 10 10 11 ...
## $ dist : num  2 10 4 22 16 10 18 26 34 17 ...
## n1 :   num 5
## n2 :   num 50
## name :  chr "Juan"
## p :   num 0.5
```

Para borrar objetos en memoria, utilizamos la función `rm()`:

```
rm(n1)
```

Objetos para los datos:

vector: colección ordenada de elementos del mismo tipo.

```
x <- c(1, 2, 3); y <- c("a", "b", "c")
z<- c(TRUE, TRUE, FALSE)
```

array: generalización multidimensional del vector. Elementos del mismo tipo.

```
x <- array(1:3,c(2,4))
```

data frame: como el array, pero con columnas de diferentes tipos. Es el objeto más habitual para los datos experimentales.

```
resul<-data.frame(ID=c("alum1","alum2","alum3","alum4"),
peso=c(54,53,48,58),
altura=c(1.62,1.65,1.68,1.7),
CI=c(98,86,89,90))
```

Como resultado de salida es:

```
resul
```

```
##      ID peso altura CI
## 1 alum1   54   1.62  98
## 2 alum2   53   1.65  86
## 3 alum3   48   1.68  89
## 4 alum4   58   1.70  90
```

Si interesa referirse exclusivamente a los datos de peso se expresará de la siguiente forma:

```
resul$peso
```

```
## [1] 54 53 48 58
```

factor: tipo de vector para datos categóricos.

```
x<- factor(c(1, 2, 2, 1, 1, 2, 1, 2, 1))
```

Se puede discretizar datos usando las funciones *split* y *cut* . Veamos:

```
n<-10;n1<-10
g<- factor(letters[1:5],label="a")
x<- rnorm(n * n1) + sqrt(as.numeric(g))
xg<- split(x, g)
print(xg,digits=2)
```

```
## $a1
## [1] 1.03 1.49 1.90 -0.66 0.93 0.70 3.61
## [8] 1.52 1.33 1.43 0.04 1.80 1.40 1.27
## [15] 1.45 1.00 1.74 0.82 1.17 -0.98
##
## $a2
## [1] 2.33 2.61 0.51 2.91 0.95 3.32 0.95
## [8] 1.57 3.31 1.49 2.54 1.40 1.98 -0.57
## [15] 0.38 1.44 1.32 2.48 1.38 2.80
##
## $a3
## [1] 1.33 0.35 2.60 2.33 3.22 1.11 0.52 1.94 2.21
## [10] 1.81 2.06 1.28 3.27 2.47 1.66 0.92 0.19 1.59
## [19] 4.01 0.37
##
## $a4
## [1] 1.74 2.99 3.31 0.53 3.06 3.58 1.82 1.80 1.82
## [10] 2.67 3.62 2.89 3.54 0.55 3.62 0.80 0.62 3.12
## [19] 2.86 2.46
##
## $a5
```

```
## [1] 2.1 2.7 1.8 1.8 2.8 2.7 1.5 3.8 3.3
## [10] -1.3 1.7 2.9 1.8 3.5 1.4 1.9 1.9 1.3
## [19] 2.6 3.3
```

Otro caso:

```
v1<-c(7,8,10,1,2,5)
cut1<-cut(v1,3)
summary(cut1)
```

```
## (0.991,4]      (4,7]      (7,10]
##           2           2           2
```

list: vector generalizado. Cada lista está formada por componentes, que pueden ser otras listas, y cada componente puede ser de distinto tipo. Son contenedores generales de datos. Muy flexibles, pero sin estructura. Muchas funciones devuelven una lista o conjunto de resultados de distinta longitud y distinto tipo.

```
una.lista<-c(un.vector=1:5,una.palabra="gracias",
  una.matriz=matrix(round(rnorm(20),3),ncol=5),
  lista2=c(a=3,b=factor(c("a","b"))))
```

Como resultado de salida es:

```
una.lista
```

```
## un.vector1 un.vector2 un.vector3
##          "1"          "2"          "3"
## un.vector4 un.vector5 una.palabra
##          "4"          "5"      "gracias"
## una.matriz1 una.matriz2 una.matriz3
##      "-1.558"      "-0.325"      "0.555"
## una.matriz4 una.matriz5 una.matriz6
##      "0.22"      "-1.648"      "1.016"
## una.matriz7 una.matriz8 una.matriz9
##      "0.866"      "-0.813"      "-0.123"
```