JORGE NAVARRO LÓPEZ • ALBERTO TURÓN LANUZA • JUAN AGUARÓN JOVEN

# ESTADÍSTICA Y CIENCIA DE DATOS CON R

DIAZ DE SANTOS

**JORGE NAVARRO LÓPEZ**
**ALBERTO TURÓN LANUZA**
**JUAN AGUARÓN JOVEN**

# ESTADÍSTICA Y CIENCIA DE DATOS CON R

Madrid • Buenos Aires • México • Bogotá

© Jorge Navarro López, Alberto Turón Lanuza, Juan Aguarón Joven, 2025

(edición ebook)

Reservados todos los derechos.

“No está permitida la reproducción total o parcial de este libro, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier
medio, ya sea electrónico, mecánico por fotocopia, por registro u otros métodos, sin el permiso previo y por escrito de los titulares del Copyright.”

Ediciones Díaz de Santos
Internet: http//www.editdiazdesantos.com
E-mail: ediciones@editdiazdesantos.com

e-ISBN: 978-84-9052-555-5 (edición digital)

ISBN: 978-84-9052-554-8 (edición papel)

Fotocomposición y diseño de cubiertas: P55 Servicios Culturales

# Índice general

**Notas de los autores** vii

**Introducción** ix

**I Trabajando con R y R Commander** 1

**1. Introducción a R** 3
1.1. El lenguaje R . . . 3
1.2. Instalación de R . . . 4
1.2.1. Usuarios de Windows . . . 5
1.2.2. Usuarios de Mac OS . . . 8
1.3. Manejo básico de `R` . . . 9
1.4. Instalación de `R Commander` . . . 10
1.4.1. Instalación de la librería `TeachStat` . . . 12
1.5. Trabajar con `R Commander` . . . 12

**2. Trabajando con datos** 15
2.1. Importar datos externos . . . 15
2.1.1. Importar datos desde una URL . . . 17
2.2. Guardar datos . . . 17
2.3. Estructuras de datos . . . 19
2.3.1. Vectores . . . 19
2.3.2. Matrices . . . 20
2.3.3. Arrays . . . 22
2.3.4. Factores . . . 23
2.3.5. Listas . . . 24
2.3.6. Data Frames . . . 26
2.4. Transformación de datos . . . 27

2.4.1. Filtrar filas . . . . . 28
2.4.2. Seleccionar columnas . . . . . 28
2.4.3. Crear nuevas columnas . . . . . 28
2.4.4. Ordenar por filas . . . . . 29
2.4.5. Recodificar variables . . . . . 29
2.5. Ejemplos resueltos . . . . . 30

**II Estadística Descriptiva 35**

**3. Estadística Descriptiva Univariante 37**
3.1. Conceptos básicos . . . . . 37
3.1.1. Tabulación . . . . . 37
3.1.1.1. Frecuencias . . . . . 38
3.1.1.2. Distribución de frecuencias . . . . . 39
3.1.2. Representación gráfica . . . . . 41
3.1.3. Descripción numérica . . . . . 41
3.2. Estadística Univariante con `R` . . . . . 45
3.2.1. Tabulación y representación gráfica con `R` . . . . . 46
3.2.2. Descripción numérica con `R` . . . . . 54
3.3. Estadística Univariante con `R Commander` . . . . . 58
3.4. Estadística Univariante con `TeachStat` . . . . . 65

**4. Estadística Descriptiva Bivariante 75**
4.1. Conceptos básicos . . . . . 75
4.1.1. Tabulación de datos bivariantes . . . . . 76
4.1.1.1. Frecuencias conjuntas . . . . . 76
4.1.1.2. Frecuencias marginales . . . . . 77
4.1.1.3. Frecuencias condicionadas . . . . . 79
4.1.1.4. Independencia . . . . . 80
4.1.2. Teoría de Correlación . . . . . 81
4.1.3. Teoría de Regresión . . . . . 84
4.1.3.1. Bondad del Ajuste . . . . . 85
4.1.3.2. Predicción y Fiabilidad . . . . . 87
4.2. Estadística Bivariante con `R` . . . . . 88
4.2.1. Tabulación y representación gráfica . . . . . 88
4.2.2. Correlación . . . . . 92
4.2.3. Regresión lineal . . . . . 93
4.3. Estadística Bivariante con `R Commander` . . . . . 97
4.3.1. Tabulación y representación gráfica . . . . . 97

4.3.2. Correlación . . . . . 101
4.3.3. Regresión lineal . . . . . 103

**III Cálculo de Probabilidades 107**

**5. Probabilidad y Variables Aleatorias 109**
5.1. Conceptos básicos . . . . . 109
5.1.1. Características notables . . . . . 110
5.1.1.1. Esperanza matemática . . . . . 110
5.1.1.2. Varianza de una variable aleatoria . . . . . 111
5.1.1.3. Desviación típica de una variable aleatoria . 111
5.1.2. Distribuciones discretas notables . . . . . 112
5.1.2.1. Distribución uniforme discreta, $U_D(1, N)$ . . 112
5.1.2.2. Distribución de Bernoulli, $Be(p)$ . . . . . 112
5.1.2.3. Distribución binomial, $Bi(n, p)$ . . . . . 113
5.1.2.4. Distribución hipergeométrica, $H(N, n, D)$ . . 114
5.1.2.5. Distribución geométrica, $G(p)$ . . . . . 114
5.1.2.6. Distribución de Poisson, $\wp(\lambda)$ . . . . . 114
5.1.3. Distribuciones continuas notables . . . . . 115
5.1.3.1. Distribución uniforme, $U(a, b)$ . . . . . 115
5.1.3.2. Distribución exponencial, $\text{Exp}(\lambda)$ . . . . . 116
5.1.3.3. Distribución gamma, $\text{Gamma}(a, \lambda)$ . . . . . 116
5.1.3.4. Distribución normal, $N(\mu, \sigma)$ . . . . . 117
5.2. Cálculo de probabilidades con `R` y `R Commander` . . . . . 118
5.3. Ejemplos resueltos . . . . . 124
5.4. Ejercicios propuestos . . . . . 139
5.5. Preguntas teórico–prácticas . . . . . 144
5.6. Soluciones . . . . . 150
5.6.1. Soluciones a los ejercicios propuestos . . . . . 150
5.6.2. Soluciones a las preguntas teórico–prácticas . . . . . 155

**IV Inferencia estadística 157**

**6. Muestreo 159**
6.1. Conceptos básicos . . . . . 159
6.1.1. Muestreo aleatorio simple . . . . . 159
6.1.1.1. Muestreo aleatorio simple con reposición . . 160
6.1.1.2. Muestreo aleatorio simple sin reposición . . . 160

6.1.2. Estadísticos muestrales . . . . . 161
6.1.2.1. Media muestral . . . . . 162
6.1.2.2. Proporción muestral . . . . . 162
6.1.2.3. Varianza y cuasivarianza muestrales . . . . . 162
6.1.2.4. Suma muestral . . . . . 163
6.1.3. Distribución de los estadísticos muestrales . . . . . 163
6.1.3.1. Distribuciones exactas . . . . . 164
6.1.3.2. Distribuciones aproximadas . . . . . 164
6.2. Ejemplos resueltos . . . . . 166
6.3. Ejercicios propuestos . . . . . 174
6.4. Preguntas teórico–prácticas . . . . . 177
6.5. Soluciones . . . . . 182
6.5.1. Soluciones a los ejercicios propuestos . . . . . 182
6.5.2. Soluciones a las preguntas teórico–prácticas . . . . . 186

**7. Estimación puntual y por Intervalos de Confianza 189**
7.1. Conceptos básicos . . . . . 189
7.1.1. Estimación puntual . . . . . 189
7.1.1.1. Métodos de construcción de estimadores . . . 189
7.1.1.2. Propiedades de los estimadores . . . . . 190
7.1.2. Estimación por intervalos . . . . . 191
7.1.3. Distribuciones relacionadas con los intervalos de confianza . . . . . 191
7.1.3.1. Distribución chi–cuadrado de Pearson . . . . 191
7.1.3.2. Distribución $t$ de Student . . . . . 191
7.1.4. Intervalos de confianza notables para una muestra . . 193
7.1.4.1. Intervalo de confianza para la media de una normal con varianza conocida . . . . . 193
7.1.4.2. Intervalo de confianza para la media de una normal con varianza desconocida . . . . . 193
7.1.4.3. Intervalo de confianza para la proporción $p$ de una distribución Bernoulli con tamaño muestral elevado . . . . . 194
7.1.4.4. Intervalo de confianza para la varianza de una distribución normal . . . . . 194
7.1.5. Determinación del tamaño muestral . . . . . 194
7.1.5.1. Estimación de la media de una distribución normal . . . . . 195

7.1.5.2. Estimación del parámetro $p$ de una distribución Bernoulli con tamaño muestral elevado . 195
7.1.5.3. Corrección por población finita . . . . . 195
7.2. Ejemplos resueltos . . . . . 196
7.3. Ejercicios propuestos . . . . . 219
7.4. Preguntas teórico–prácticas . . . . . 224
7.5. Soluciones . . . . . 233
7.5.1. Soluciones a los ejercicios propuestos . . . . . 233
7.5.2. Soluciones a las preguntas teórico–prácticas . . . . . 238

**8. Contrastes de hipótesis paramétricas 241**
8.1. Conceptos básicos . . . . . 241
8.1.1. Contrastes de hipótesis . . . . . 241
8.1.2. Tipos de errores . . . . . 243
8.1.3. Metodología del contraste de hipótesis . . . . . 244
8.1.4. Contrastes de hipótesis paramétricas . . . . . 244
8.1.4.1. Ejemplos de contrastes paramétricos . . . . . 245
8.1.5. Aplicaciones . . . . . 245
8.1.6. Contrastes más comunes para un parámetro de una variable . . . . . 246
8.1.6.1. Contraste para la media de una población con varianza desconocida . . . . . 246
8.1.6.2. Contraste para la proporción de una variable Bernoulli . . . . . 247
8.1.6.3. Contraste para la varianza de una variable normal . . . . . 247
8.2. Ejemplos resueltos . . . . . 249
8.3. Ejercicios propuestos . . . . . 272
8.4. Preguntas teórico–prácticas . . . . . 278
8.5. Soluciones . . . . . 287
8.5.1. Soluciones a los ejercicios propuestos . . . . . 287
8.5.2. Soluciones a las preguntas teórico–prácticas . . . . . 293

**9. Análisis paramétrico de dos poblaciones 297**
9.1. Conceptos básicos . . . . . 297
9.1.1. Tipos de contrastes para dos poblaciones . . . . . 297
9.1.2. Aplicaciones . . . . . 298
9.1.3. Distribución de probabilidad $F$ de Snedecor . . . . . 299
9.2. Análisis paramétrico de dos muestras independientes . . . . . 300

9.2.1. Diferencia entre las medias de dos poblaciones normales con varianzas desconocidas que pueden suponerse iguales . . . 300
9.2.2. Diferencia entre las medias de dos poblaciones normales con varianzas desconocidas que no pueden suponerse iguales . . . 300
9.2.3. Diferencia entre las proporciones de dos poblaciones cuando se tienen muestras grandes . . . 301
9.2.4. Cociente de las varianzas de dos poblaciones normales 302
9.3. Análisis paramétrico de dos muestras pareadas . . . 302
9.3.1. Diferencia entre las medias de dos poblaciones . . . 303
9.3.2. Diferencia entre las proporciones de dos poblaciones . 303
9.4. Ejemplos resueltos . . . 304
9.4.1. Muestras independientes . . . 304
9.4.2. Muestras pareadas . . . 324
9.5. Ejercicios propuestos . . . 330
9.6. Preguntas teórico–prácticas . . . 335
9.7. Soluciones . . . 349
9.7.1. Soluciones a los ejercicios propuestos . . . 349
9.7.2. Soluciones a las preguntas teórico–prácticas . . . 353

**10. Contrastes de hipótesis no paramétricas 357**
10.1. Conceptos básicos . . . 357
10.1.1. Las pruebas de la chi–cuadrado . . . 358
10.1.2. Pruebas de bondad de ajuste . . . 359
10.1.2.1. Test de la chi–cuadrado de bondad de ajuste 359
10.1.2.2. Test de bondad de ajuste a una distribución Normal . . . 360
10.1.3. Test de la chi–cuadrado de independencia . . . 360
10.1.4. Aplicaciones . . . 362
10.2. Ejemplos resueltos . . . 362
10.2.1. Test de bondad de ajuste a una distribución normal . 362
10.2.2. Test de la chi–cuadrado de independencia . . . 366
10.3. Ejercicios propuestos . . . 375
10.4. Preguntas teórico–prácticas . . . 377
10.5. Soluciones . . . 380
10.5.1. Soluciones a los ejercicios propuestos . . . 380
10.5.2. Soluciones a las preguntas teórico–prácticas . . . 381

# Notas de los autores

**Estructura y organización del contenido**
Este libro se enfoca principalmente en el cálculo de probabilidades y la inferencia estadística. Por ello, los capítulos dedicados a estos temas siguen una estructura coherente y didáctica, que incluye una exposición teórica, ejemplos resueltos, ejercicios propuestos, ejercicios teórico–prácticos y, finalmente, las soluciones correspondientes. Sin embargo, con el propósito de ofrecer una introducción accesible y ampliar el alcance del texto, se han añadido dos capítulos iniciales centrados en la estadística descriptiva univariante y bivariante utilizando `R`. En estos capítulos se adopta un enfoque distinto, más directo y práctico, presentando conceptos teórico-prácticos acompañados de ejemplos ilustrativos, para facilitar desde el inicio una aproximación exploratoria a los datos.

**Notación numérica y criterios de consistencia**
A lo largo del libro se ha decidido emplear el punto como separador decimal en lugar de la coma, basándose en dos fundamentos principales. En primer lugar, esta elección se alinea con la recomendación de la Real Academia Española (RAE), que avala este uso en ámbitos científicos y técnicos, además de concordar con las normas del Sistema Internacional de Unidades (SI). Este criterio persigue fomentar la estandarización de la notación numérica en contextos académicos y especializados. En segundo lugar, se busca mantener la coherencia con el entorno de trabajo en `R`, donde el punto es el separador decimal predeterminado. Esta decisión contribuye a evitar confusiones en la interpretación de los resultados y garantiza una presentación uniforme a lo largo del texto.

De igual modo, se ha descartado el uso del punto como separador de miles, siguiendo las recomendaciones de la RAE y las directrices del SI. En su lugar, se emplea un espacio para agrupar las cifras en miles, lo que favorece la legibilidad y evita posibles ambigüedades con el separador decimal.

**Materiales complementarios y agradecimientos**

Los archivos de Excel que contienen los conjuntos de datos utilizados en el libro están disponibles para consulta y descarga en la siguiente dirección:

`https://www.editdiazdesantos.com/libros/9788490525548`

Estos recursos permiten a los lectores reproducir los análisis presentados y experimentar de manera autónoma con los datos, aplicando las técnicas estadísticas y herramientas descritas en el texto.

Finalmente, los autores expresan su más sincero agradecimiento a los profesores Tomás R. Cotos Yañez, Manuel A. Mosquera Rodríguez, Ana Pérez González y Benigno Reguengo Lareo, del Departamento de Estadística e Investigación Operativa de la Universidad de Vigo, por el desarrollo del paquete `RcmdrPlugin.TeachStat`. Esta herramienta, fruto de su labor académica e investigadora, representa una valiosa contribución a la enseñanza de la estadística, al facilitar la aplicación práctica de conceptos teóricos en un entorno accesible e integrado en `R Commander`.

# Introducción

¿Qué probabilidades tengo de que uno de mis clientes se vaya a otra compañía? ¿Debo vender mis acciones? ¿Por cuánto dinero podría vender mi casa? ¿Este email es *spam*? ¿Cuánto debería cobrar a este cliente por su póliza de seguros? ¿Qué productos debería poner juntos en las estanterías de mi supermercado? ¿Debería conceder un crédito a este cliente? Preguntas como estas se formulan en todo el mundo miles de veces cada día. Todas ellas tienen una cosa en común: la respuesta no está escrita en ningún manual. Habitualmente, estas preguntas preceden a una decisión que, en el mejor de los casos, no sabremos si ha sido correcta o equivocada hasta que sea demasiado tarde, e incluso puede que nunca lleguemos a saber si la decisión que tomamos fue buena o mala.

Para garantizar la mejor decisión posible tratamos de recoger y analizar toda la información disponible con el mayor rigor posible. La Estadística es la ciencia que nos ayuda a procesar esta información y a utilizarla en la búsqueda de soluciones. Si bien nada de lo dicho hasta aquí es nuevo, la realidad que nos trae el siglo XXI es que los estudiantes de la era *Big Data* tienen a su alcance métodos estadísticos y computacionales que hasta ahora no habíamos podido ni soñar. La *Ciencia de Datos*, como actualmente nos gusta llamarla, combina la Estadística clásica con novedosas técnicas desarrolladas en el ámbito de las Ciencias de la Computación, ofreciendo a nuestros graduados un esperanzador futuro en un mundo en el que Google interpreta las consultas realizadas por los usuarios a través de varias interfaces (navegador, *Hey Google*...), las entidades de crédito analizan el riesgo potencial de sus clientes en función de perfiles minuciosamente elaborados a partir de información de lo más variada que incluye hasta su actividad en redes sociales, las compañías de entretenimiento a través de *streaming* utilizan técnicas predictivas para recomendar películas y series a sus suscriptores, los gobiernos utilizan modelos biométricos para identificar a sospechosos de terrorismo, agencias de verificación analizan procesos de transmisión de mensajes en redes para

detectar *fake news*, se utilizan técnicas de análisis de *outliers* para detectar comportamientos sospechosos en el uso de tarjetas de crédito, intrusiones en los sistemas informáticos, anomalías en los datos medidos por sensores remotos, y un largo etcétera. Estos son unos pocos ejemplos en los que los métodos estadísticos, combinados con la capacidad casi infinita de procesar grandes volúmenes de datos a gran velocidad, proporcionan soluciones a problemas que hasta hace poco ni nos hubiéramos planteado.

La explosión de datos generada en las últimas décadas —impulsada por internet, los dispositivos móviles, los sensores, las redes sociales, entre otros factores— ha favorecido el desarrollo de sistemas de inteligencia artificial que hoy en día afectan prácticamente todos los ámbitos de nuestra vida. Esta transformación ha generado la necesidad de crear nuevas técnicas estadísticas y herramientas computacionales, propias de la Ciencia de Datos, capaces de almacenar, procesar y analizar grandes volúmenes de información. Todo ello ha acelerado el aprendizaje de las inteligencias artificiales y ha facilitado su aplicación en campos como la medicina, el marketing o las finanzas, entre otros.

Entre los estudiantes de dobles grados, actualmente el 40 % eligen uno con contenidos en Economía y Ciencia de Datos. Los Científicos de Datos aplican la teoría económica a situaciones económicas del mundo real y explotan las grandes bases de datos para guiar las decisiones económicas en finanzas, gobierno, ocio o industria. Basta un simple vistazo a las ofertas de empleo para observar que numerosas empresas están reemplazando a los tradicionales CEO por CDO: *Chief Digital Officer*. La vertiginosa evolución que está experimentando la Estadística, y que no lleva visos de frenarse al menos en los próximos diez años, ofrece a los estudiantes de estas especialidades un extensísimo campo en el que desarrollar exitosas carreras profesionales.

Aquí van algunos hechos que apoyan estas afirmaciones[1]:

- Ya hace algunos años que la demanda de analistas y científicos de datos supera con creces la oferta. Los salarios de un recién graduado en Ciencia de Datos están muy por encima de la media.

- Empresas de todos los sectores y tamaños han acumulado ingentes cantidades de datos desde la implantación, hace veinte o treinta años, de los sistemas de gestión de bases de datos. Ahora buscan profesionales

[1] Fuente: Oficina de Estadísticas Laborales, Departamento de Trabajo de EE. UU., Manual de Perspectivas Ocupacionales, Matemáticos y Estadísticos (visitado el 19 de marzo de 2025). [https://www.bls.gov/ooh/math/mathematicians-and-statisticians.htm]

con estos perfiles para saber interpretar sus datos y extraer de ellos información de la que puedan beneficiarse.

- Se espera que la tasa de empleo de matemáticos y estadísticos crezca un 11 por ciento entre 2023 y 2033, mucho más rápido que el promedio de todas las profesiones. Para esta década se prevé la creación de 3 900 puestos de trabajo cada año para matemáticos y estadísticos.

Alcanzar esta tierra prometida exige de los estudiantes una sólida formación matemática y un buen conocimiento de los métodos estadísticos, pero también cierta habilidad para utilizar técnicas computacionales que les permitan gestionar grandes cantidades de datos a los que aplicar dichos métodos. Por eso pretendemos, con esta colección de problemas, enseñarles los métodos estadísticos a la vez que se introducen en el uso de `R`, un lenguaje de programación orientado a la Estadística.

`R` es un entorno de programación de software libre que incorpora prácticamente todas las herramientas estadísticas creadas hasta ahora, al mismo tiempo que ofrece una gran flexibilidad para crear herramientas nuevas o adaptar las existentes a un problema específico. Al mismo tiempo, `R` está orientado al manejo de datos, proporcionando medios tanto para la adquisición de estos desde fuentes externas (bases de datos relacionales, fuentes de *open data*, API de aplicaciones web, *web scraping*...) como desde datos almacenados localmente en una gran variedad de formatos. Por último, `R` implementa numerosas herramientas gráficas que permiten realizar potentes análisis gráficos de datos.

Introducir al estudiante de Estadística en el manejo de este lenguaje `R` es un paso necesario para complementar su formación estadística con vistas a una futura salida profesional como las arriba descritas. Por eso, el reto que le proponemos en este libro no es sólo que aprenda Estadística: es que la aprenda a través de una herramienta computacional que le permita abrirse paso en el esperanzador mundo de la Ciencia de Datos. Al alcanzar su graduación, el estudiante debería estar preparado para seguir alguno de los cada vez más numerosos másteres y cursos de especialización en Ciencia de Datos que le abrirán las puertas de un brillante futuro profesional.

# Parte I

# Trabajando con R y R Commander

# Capítulo 1

# Introducción a R

## 1.1. El lenguaje R

`R` es un lenguaje de programación orientado al análisis estadístico de datos. En la actualidad es, junto con `Python`, el lenguaje más utilizado por la comunidad científica aunque, a diferencia de éste, `R` está especialmente orientado al ámbito de la estadística y las matemáticas financieras.

Al ser un lenguaje de código abierto, `R` permite ampliar sus funcionalidades mediante paquetes o bibliotecas creados por terceros. Así, `R` ofrece numerosas librerías de visualización gráfica que facilitan el análisis, no sólo numérico sino también gráfico, de los datos.

Su facilidad y versatilidad para acceder a fuentes de datos externas permiten que `R` pueda leer y manejar grandes volúmenes de datos. Además, es un lenguaje interactivo, por lo que resulta fácil de aprender.

En esta primera sección veremos cómo instalar `R` en nuestro ordenador. También veremos cómo instalar `R Commander`, un interfaz gráfico que nos permitirá aplicar las técnicas estadísticas que vamos a ver en este libro sin necesidad de conocer en detalle el lenguaje `R`. Los ejemplos del libro están resueltos por ambos procedimientos: utilizando los comandos de `R` y mediante `R Commander`. Este último facilita mucho el trabajo para quienes no tengan conocimientos de programación y resulta suficiente para todo lo que necesitamos hacer. Sin embargo, recomendamos a los alumnos que aprendan los fundamentos del lenguaje `R` porque les puede resultar muy útil para sus futuros estudios o a lo largo de su carrera profesional.

## 1.2. Instalación de R

El software `R` se descarga gratuitamente de la web oficial del proyecto R, `http://www.r-project.org` (Figura 1.1). En ella se pueden encontrar versiones para plataformas Linux, MacOS y Windows, tanto binarios auto-ejecutables como el código fuente original, además de abundante información y manuales. En el momento de escribir estas líneas, la versión estable de `R` es la R 4.4.3, *Trophy Case*, publicada el 28 de febrero de 2025.

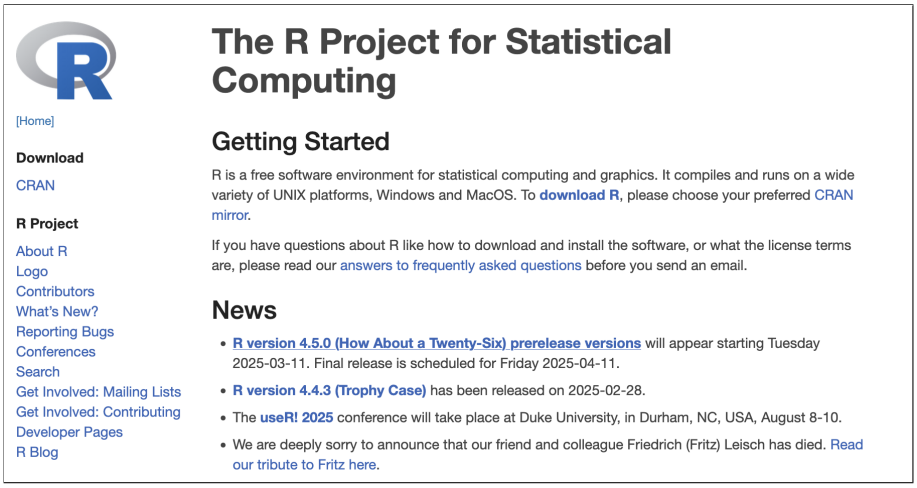
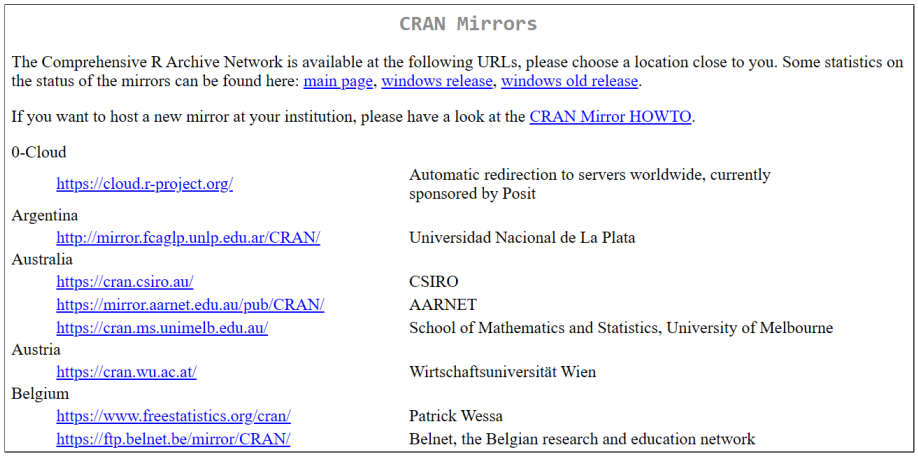



**Figura 1.1:** Página principal del proyecto R.

Seleccionando el enlace **download R** accedemos a una página en la que debemos elegir el mirror (repositorio) desde el que queremos descargar el software (Figura 1.2). La lista es bastante extensa, encontrándose ubicados en diferentes zonas geográficas. Podemos seleccionar, por ejemplo, `0-Cloud`, que nos redirige automáticamente a alguno de los repositorios disponibles.

Independientemente del repositorio seleccionado, accedemos a una página que nos permite elegir entre la descarga de los binarios (instaladores) para diferentes sistemas operativos, o del código fuente (Figura 1.3). Hay que seleccionar el enlace correspondiente al sistema operativo del ordenador donde se quiere instalar. A continuación, veremos cómo es la instalación en los sistemas Windows y MacOS.

Suelen publicarse cuatro o cinco versiones nuevas cada año; cada una presenta pequeñas mejoras con respecto a la anterior, pero no es necesario instalar todas. Recomendamos instalar la más reciente una vez al año.



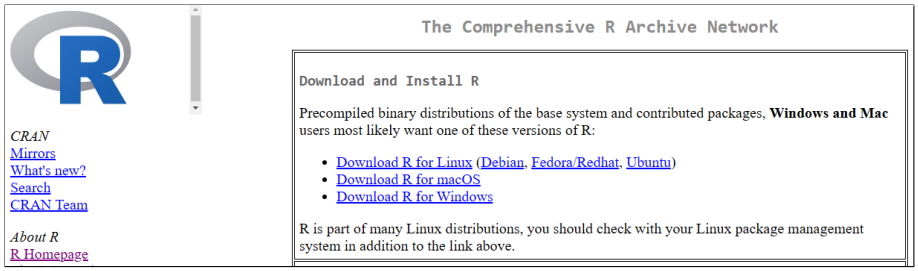
CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: main page, windows release, windows old release.

If you want to host a new mirror at your institution, please have a look at the CRAN Mirror HOWTO.

| | |
|---|---|
| 0-Cloud | |
| https://cloud.r-project.org/ | Automatic redirection to servers worldwide, currently sponsored by Posit |
| Argentina | |
| http://mirror.fcaglp.unlp.edu.ar/CRAN/ | Universidad Nacional de La Plata |
| Australia | |
| https://cran.csiro.au/ | CSIRO |
| https://mirror.aarnet.edu.au/pub/CRAN/ | AARNET |
| https://cran.ms.unimelb.edu.au/ | School of Mathematics and Statistics, University of Melbourne |
| Austria | |
| https://cran.wu.ac.at/ | Wirtschaftsuniversität Wien |
| Belgium | |
| https://www.freestatistics.org/cran/ | Patrick Wessa |
| https://ftp.belnet.be/mirror/CRAN/ | Belnet, the Belgian research and education network |

**Figura 1.2:** Repositorios de R.



CRAN
Mirrors
What's new?
Search
CRAN Team

About R
R Homepage

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux (Debian, Fedora/Redhat, Ubuntu)
- Download R for macOS
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

**Figura 1.3:** Página principal de descargas.

### 1.2.1. Usuarios de Windows

La selección de la opción **Download R for Windows** conduce a otra página en la que hay que elegir (Figura 1.4) el enlace **install R for the first time**. Dicho enlace lleva a una nueva página (Figura 1.5), en la que se encuentra el enlace **Download R-4.4.3 for Windows** que descargará el programa de instalación que debemos ejecutar.

Aparece el cuadro del asistente de instalación que nos pide seleccionar el idioma. A continuación, nos muestra el tipo de licencia y presionamos en **Siguiente**. Nos indica la carpeta donde va a instalar los archivos, proporcionando una por defecto. La recomendación es no modificarla y pulsar **Siguiente**.

A continuación, hay que seleccionar las componentes que queremos instalar. Se recomienda mantener la selección por defecto (todos los elementos) y presionar en **Siguiente**. Esto conduce a un diálogo que permite indicar si queremos utilizar las opciones de configuración (Figura 1.6).

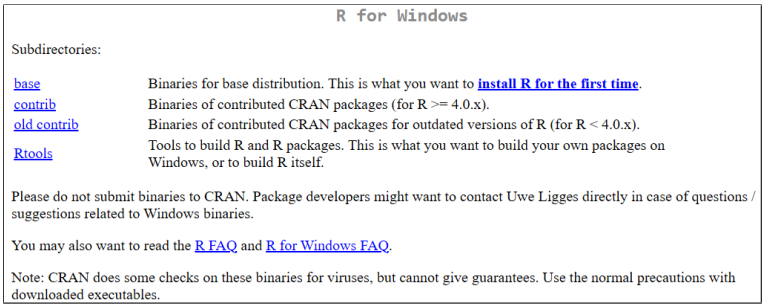



**Figura 1.4:** Página de descargas de Windows.

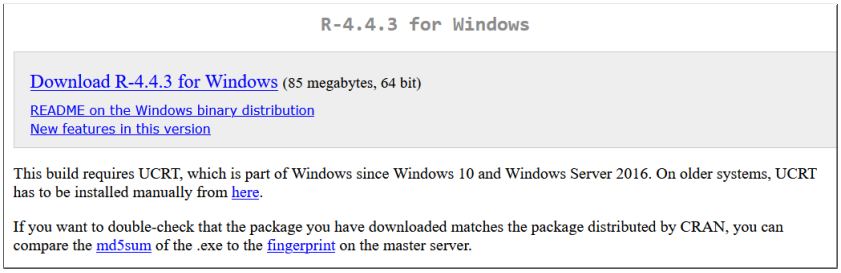



**Figura 1.5:** Página de descarga de la versión 4.4.3.

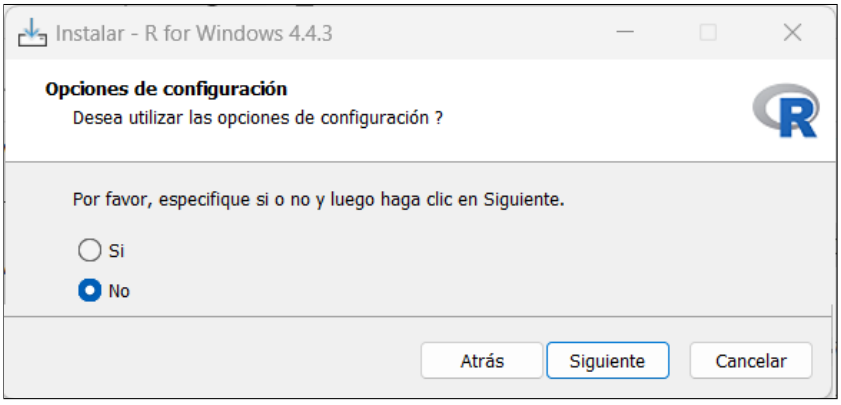



**Figura 1.6:** Usar opciones de configuración.

Podemos obviar dichas opciones y continuar con la instalación por defecto, en cuyo caso sólo nos quedan un par de diálogos. El primero nos pregunta por el nombre de la carpeta del menú Inicio (por defecto, `R`) y en el segundo diálogo podemos indicar si deseamos crear un acceso directo en el escritorio y asociar ficheros `.RData` con `R`. Se recomienda mantener dichas opciones. Tras ello, habremos terminado el proceso de instalación.

En caso de que decidamos utilizar las opciones de configuración mostradas en la Figura 1.6, vamos a tener que pasar por dos diálogos adicionales. En el primero debemos seleccionar el modo de display (Figura 1.7), pudiendo seleccionar entre MDI o SDI. Cuando se trabaja con `R` suele ser más cómodo elegir MDI. Por el contrario, es conveniente escoger SDI si vamos a trabajar habitualmente con `R Commander`[1].

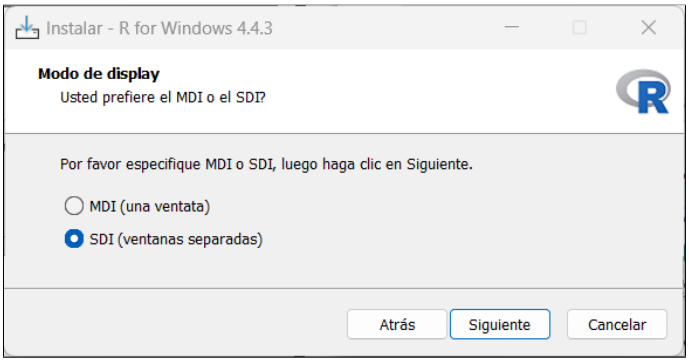



**Figura 1.7:** Opciones del modo de display.

El siguiente diálogo nos muestra la segunda opción de configuración, en la que se puede seleccionar el estilo de la ayuda, pudiendo elegir entre formato HTML o de texto simple (véase Figura 1.8). La recomendación es seleccionar el formato HTML. Con esto terminan las opciones de configuración, en caso de haberlas elegido, y pasamos a los últimos diálogos ya comentados anteriormente (nombre de la carpeta del menú Inicio, iconos en el escritorio, etc.).

Una vez finalizado el proceso de instalación, aparecerá en el escritorio el icono de la aplicación.

[1]Si se selecciona MDI, los gráficos se visualizarán dentro de la ventana general de `R`. Si trabajamos con `R` esto es cómodo, pero en caso de hacerlo con `R Commander` nos obligará a estar alternando entre la ventana de `R Commander` y la de `R` continuamente. Seleccionando SDI los gráficos se mostrarán en una ventana independiente de la de `R`.

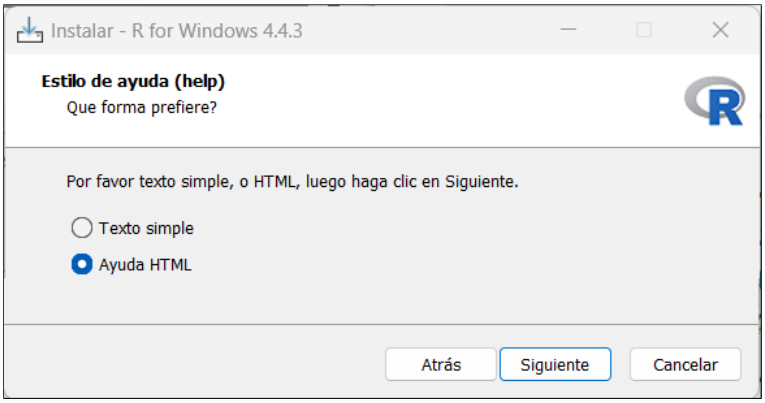



**Figura 1.8:** Opciones de la ayuda de `R`.

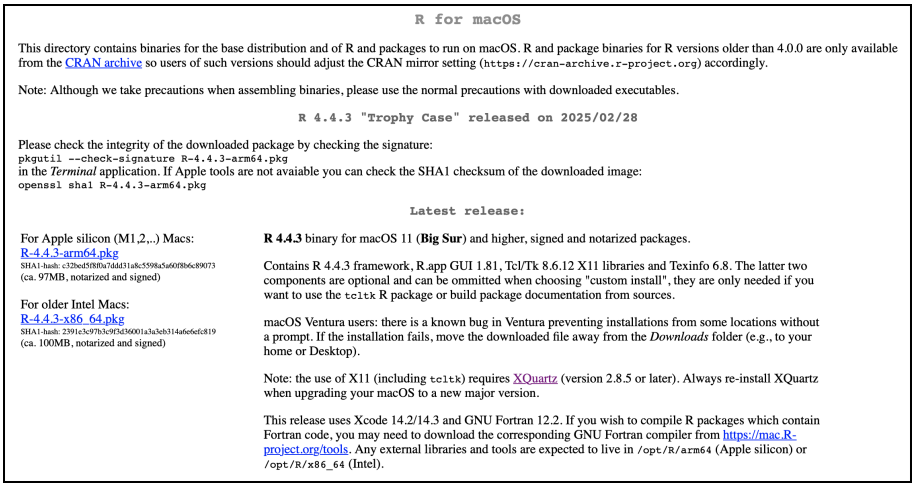
### 1.2.2. Usuarios de Mac OS

La versión 4.4.3 funciona con cualquier versión de OS X a partir de la 11, conocida como *Big Sur*. En la página principal de descargas (Figura 1.3) pulsamos el enlace **Download R for macOS**, y en la página que aparecerá veremos dos enlaces correspondientes a dos versiones de instalación distintas (Figura 1.9): la primera, *R-4.4.3-arm64.pkg*, corresponde a los ordenadores que llevan procesadores M1, M2 o M3; la segunda, *R-4.4.3-x86_64.pkg*, corresponde a los ordenadores más antiguos equipados con procesadores Intel. Elegimos el correspondiente a nuestro equipo y comenzará la descarga.

Una vez descargado el paquete de instalación, debemos localizarlo donde se haya guardado al descargarlo. Lo ejecutamos y comenzará el proceso de instalación. En este caso, el proceso es muy sencillo, pues basta con aceptar todas las opciones que propone por defecto. Al concluir el proceso, aparecerá el icono del programa en la carpeta de aplicaciones.

Sin embargo, el interfaz `R Commander` requiere el uso de las librerías gráficas `X11`, que no vienen instaladas de forma nativa en Mac. Por lo tanto, debemos instalar el paquete `XQuartz`, cuya versión más reciente descargaremos de la dirección `https://www.xquartz.org` (Figura 1.10).

Pulsando en el enlace de descarga obtendremos el instalador de la versión más reciente de `XQuartz`. El paquete descargado es una versión comprimida; al abrirla veremos el instalador, un fichero llamado *XQuartz.pkg*. A continuación, basta con ejecutar este instalador y aceptar todas las opciones propuestas. Al finalizar el proceso, las librerías quedan instaladas y listas para ser usadas por `R Commander`.



R for macOS

This directory contains binaries for the base distribution and of R and packages to run on macOS. R and package binaries for R versions older than 4.0.0 are only available from the CRAN archive so users of such versions should adjust the CRAN mirror setting (`https://cran-archive.r-project.org`) accordingly.

Note: Although we take precautions when assembling binaries, please use the normal precautions with downloaded executables.

R 4.4.3 "Trophy Case" released on 2025/02/28

Please check the integrity of the downloaded package by checking the signature:
`pkgutil --check-signature R-4.4.3-arm64.pkg`
in the *Terminal* application. If Apple tools are not avaiable you can check the SHA1 checksum of the downloaded image:
`openssl sha1 R-4.4.3-arm64.pkg`

Latest release:

For Apple silicon (M1,2,..) Macs:
R-4.4.3-arm64.pkg
SHA1-hash: c32bed5f8f0a7ddd31a8c5598a5a60f8b6c89073
(ca. 97MB, notarized and signed)

For older Intel Macs:
R-4.4.3-x86_64.pkg
SHA1-hash: 2391e3c97b3c9f3d36001a3a3eb314a6e6efc819
(ca. 100MB, notarized and signed)

**R 4.4.3** binary for macOS 11 (**Big Sur**) and higher, signed and notarized packages.

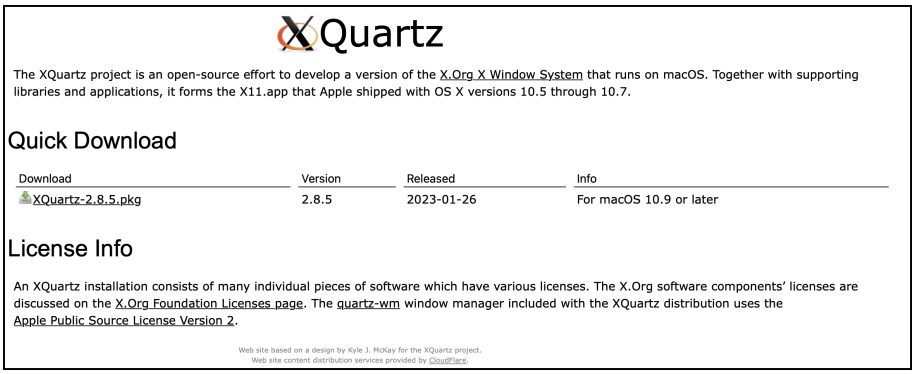
Contains R 4.4.3 framework, R.app GUI 1.81, Tcl/Tk 8.6.12 X11 libraries and Texinfo 6.8. The latter two components are optional and can be ommitted when choosing "custom install", they are only needed if you want to use the `tcltk` R package or build package documentation from sources.

macOS Ventura users: there is a known bug in Ventura preventing installations from some locations without a prompt. If the installation fails, move the downloaded file away from the *Downloads* folder (e.g., to your home or Desktop).

Note: the use of X11 (including `tcltk`) requires XQuartz (version 2.8.5 or later). Always re-install XQuartz when upgrading your macOS to a new major version.

This release uses Xcode 14.2/14.3 and GNU Fortran 12.2. If you wish to compile R packages which contain Fortran code, you may need to download the corresponding GNU Fortran compiler from https://mac.R-project.org/tools. Any external libraries and tools are expected to live in `/opt/R/arm64` (Apple silicon) or `/opt/R/x86_64` (Intel).

**Figura 1.9:** Enlaces de descarga de R para Mac OS.



XQuartz

The XQuartz project is an open-source effort to develop a version of the X.Org X Window System that runs on macOS. Together with supporting libraries and applications, it forms the X11.app that Apple shipped with OS X versions 10.5 through 10.7.

Quick Download

| Download | Version | Released | Info |
|---|---|---|---|
| XQuartz-2.8.5.pkg | 2.8.5 | 2023-01-26 | For macOS 10.9 or later |

License Info

An XQuartz installation consists of many individual pieces of software which have various licenses. The X.Org software components' licenses are discussed on the X.Org Foundation Licenses page. The quartz-wm window manager included with the XQuartz distribution uses the Apple Public Source License Version 2.

Web site based on a design by Kyle J. McKay for the XQuartz project.
Web site content distribution services provided by CloudFlare.

**Figura 1.10:** Página de descarga de la librería gráfica `XQuartz`.

## 1.3. Manejo básico de `R`

`R` es un lenguaje de programación interactivo, lo que significa que se pueden escribir comandos directamente en la consola y al presionar la tecla `Enter` los ejecuta, mostrando a continuación el resultado (si lo hay). Los comandos se escriben en la última línea de la ventana de `R`, justo a continuación del símbolo `>` (llamado *prompt*).

Por ejemplo, si escribimos la instrucción `print("Hola, mundo!")` y pul-

samos `Enter` el resultado será el siguiente:

```
> print("Hola, mundo!")
[1] "Hola, mundo!"
```

La orden `print()` es una **función**; indica que `R` debe escribir en la línea o líneas siguientes lo que haya entre paréntesis, que en este caso es un texto.

Se pueden efectuar operaciones aritméticas directamente en la consola:

```
> 3 + 5    # Suma
[1] 8

> 10 / 2   # División
[1] 5

> sqrt(16) # Raíz cuadrada
[1] 4

> 2^3      # Potencia
[1] 8
```

o se pueden asignar valores a variables, utilizando los símbolos `<-` o el símbolo `=`, para ser usados posteriormente:

```
> a <- 10                       # O también a = 10
> b <- 20                       # O también b = 20
> suma <- a + b                 # O también suma = a + b
> suma
[1] 30
```

Nótese que el carácter `#` se emplea en `R` para insertar comentarios.

## 1.4. Instalación de `R Commander`

`R` admite la instalación de scripts creados por terceros con el fin de realizar diversas tareas (cálculos financieros, cálculos matemáticos, visualización de datos, acceso a API de aplicaciones externas y una gran cantidad de funcionalidades diversas), en forma de paquetes, que una vez se han integrado en la aplicación reciben el nombre de *librerías*. Para facilitar el análisis estadístico de datos a personas no expertas en el lenguaje de programación `R`, se ha creado la librería `R Commander`, que consiste en una interfaz gráfica en la que las principales funciones estadísticas pueden seleccionarse mediante

opciones de un menú, en lugar de tener que recordar los comandos de `R` o tener que conocer su sintaxis en detalle.

Antes de utilizar `R Commander` por primera vez, es necesario instalar el paquete, que recibe el nombre de `Rcmdr`. Para instalar un paquete en `R` se debe acceder a la opción `Paquetes` y dentro de ella seleccionar `Instalar paquetes`. La primera vez que instalemos un paquete, nos pedirá que elijamos el mirror desde el que queremos realizar la instalación. Al igual que ocurría con la descarga del instalador de `R`, podemos seleccionar diferentes ubicaciones, bastando normalmente con seleccionar la primera (`0-Cloud`).

A continuación, se abre una nueva ventana con los paquetes disponibles en este servidor; deberemos seleccionar el paquete `Rcmdr` y automáticamente se instalará sobre `R`.

Una vez instalado el paquete `Rcmdr` lo tendremos disponible para su uso siempre que queramos utilizarlo. Para abrir `R Commander` existe la opción del menú **Cargar paquete**, en la cual podemos seleccionar `Rcmdr`. Es posible que la primera vez que se cargue solicite la instalación de alguna librería auxiliar, lo que autorizaremos pulsando en el botón Aceptar. Finalmente, se abrirá la ventana gráfica de `R Commander` (Figura 1.11).

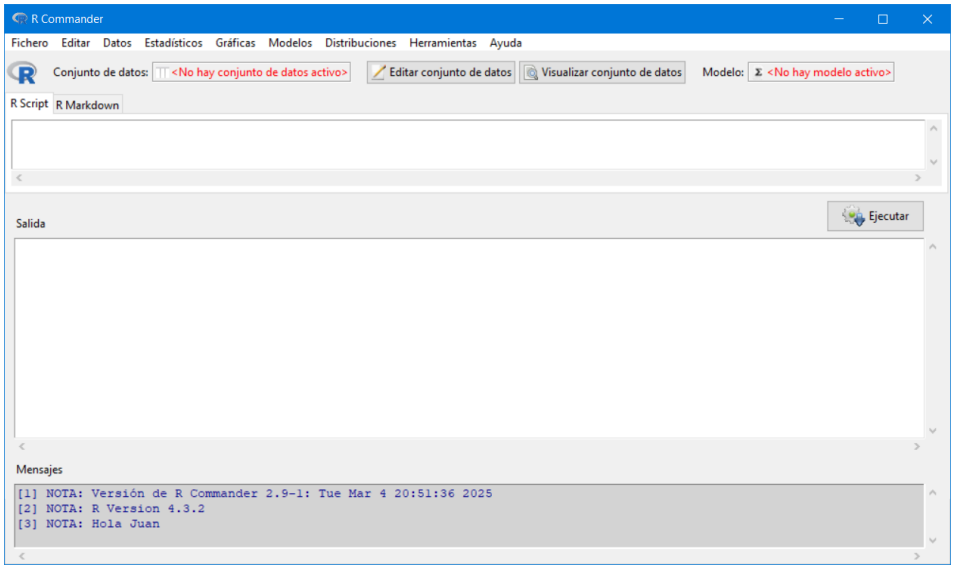



**Figura 1.11:** Interfaz gráfico de `R Commander`.

También tenemos la opción de abrir `R Commander` ejecutando el comando

```
> library(Rcmdr)
```

### 1.4.1. Instalación de la librería `TeachStat`

Algunas de las técnicas que utilizaremos en este libro vienen implementadas en un paquete auxiliar de `R Commander` llamado `TeachStat`. Para poder utilizarlo con los ejemplos del libro deberemos instalarlo previamente, una vez tengamos instalado `R Commander`. Para ello, podemos proceder de la misma manera que para la instalación de `R Commander`: desde `R` accedemos al menú `Paquetes`, escogemos la opción `Instalar paquetes`, elegimos un mirror y, finalmente, seleccionamos el paquete `RcmdrPlugin.TeachStat` de la lista de todos los disponibles.

Alternativamente lo podemos hacer desde la consola de `R` ejecutando el siguiente comando:

```
> install.packages("RcmdrPlugin.TeachStat")
```

Si la instalación es correcta, el complemento estará listo para su uso. Para ello tenemos varias opciones:

- Desde `R` podemos acceder al menú **Cargar paquete**, en la cual podemos seleccionar `RcmdrPlugin.TeachStat`.
- Desde la consola de `R` podemos escribir:

  ```
  > library(RcmdrPlugin.TeachStat)
  ```

Estas dos opciones cargan automáticamente tanto `R Commander` como el complemento `RcmdrPlugin.TeachStat`.

- Si hemos cargado previamente `R Commander`, el paquete está accesible desde el menú **Herramientas** → **Cargar plugin(s) de Rcmdr...** (véase Figura 1.12). Si lo seleccionamos de la lista de complementos que aparecerá, podremos trabajar con él durante toda la sesión de `R Commander`. Previamente deberemos reiniciar `R Commander`.

## 1.5. Trabajar con `R Commander`

Al abrir `R Commander` nos encontramos una ventana principal (Figura 1.11) dividida en varias secciones:

1. **Barra de Menús**. En la parte superior de la ventana se encuentran los menús desplegables que nos permiten acceder a las diferentes funcionalidades de `R Commander`.

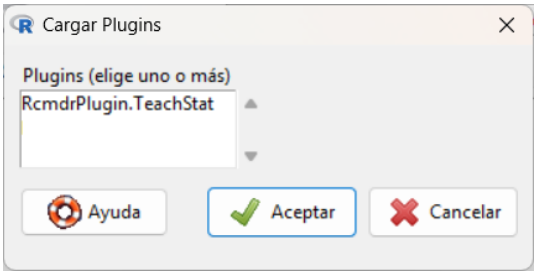



**Figura 1.12:** Activación del complemento `TeachStat`.

2. **Pestaña R Script**. En esta zona se pueden ver los comandos de `R` que se ejecutan automáticamente cuando usamos los menús y cuadros de diálogo. También podemos escribir aquí comandos de `R` directamente y ejecutarlos mediante el botón `Ejecutar`. Podemos editar scripts, que son secuencias de comandos que se pueden guardar y ejecutar posteriormente.

3. **Pestaña R Markdown**. Sirve para generar informes dinámicos, combinando el código R, los resultados (tablas, gráficos) y los comentarios explicativos en un único documento (HTML, PDF, Word...).

4. **Ventana de Salida**. Debajo de la consola y la ventana de scripts, hay una ventana de salida donde se muestran los resultados de los comandos que ejecutamos.

Las principales opciones accesibles desde el menú de `R Commander` son:

1. **Archivo:** crear un nuevo script de `R`, abrir un script existente o guardar el script actual o los resultados.

2. **Editar:** abrir un editor de datos para modificar el conjunto de datos activo o vaciar la consola de `R`.

3. **Datos:** gestionar conjuntos de datos (datasets), crear, modificar o eliminar variables en el conjunto de datos, transformar conjuntos de datos o importar datos desde Excel o diversos formatos.

4. **Estadísticos:** proporcionar resúmenes estadísticos básicos (media, mediana, desviación estándar, etc.), aplicar pruebas estadísticas inferenciales paramétricas y no paramétricas (test de la `t`, `chi-cuadrado`,

ANOVA...), análisis dimensional, o ajustar diversos modelos estadísticos (lineales, modelos lineales generalizados, etc.).

5. **Gráficos:** crear gráficos de barras, diagramas de dispersión, diagramas de caja, Q–Q plots, etc.

6. **Herramientas:** configurar opciones de `R` como el directorio de trabajo o el editor de scripts, o cargar paquetes adicionales de `R`.

7. **Ayuda:** acceder a documentación *on–line*, tutoriales y ayuda sobre `R` y `R Commander`.

Además, si instalamos el complemento `TeachStat`, veremos una nueva opción del menú llamada **Estadística Básica** desde la que podremos ejecutar la mayoría de los comandos que utilizaremos en este libro.

Hay que tener en cuenta que `R Commander` facilita el uso de `R`, ya que no necesitamos escribir código para realizar análisis básicos, y además es interactivo, de manera que podemos ver los comandos de `R` que se generan al utilizar los menús y de esta manera aprendemos `R` con ellos. Pero `R Commander` no incluye todas las funcionalidades de `R`, especialmente las más avanzadas. Para análisis complejos, es posible que necesitemos escribir código directamente en `R`. Si lo necesitamos, podemos recurrir a diversas fuentes de ayuda. Por ejemplo, la documentación oficial de `R` incluye varios manuales que cubren desde los conceptos básicos hasta temas avanzados. Se puede acceder a ellos desde la consola de `R` usando el comando:

```
> help.start()
```

También encontraremos información en la página web del proyecto CRAN `https://cran.r-project.org/manuals.html`.

Para información sobre comandos específicos se puede acceder a la documentación de cualquier función de `R` usando el comando `?` seguido del nombre de la función. Por ejemplo, el comando:

```
> ?mean
```

nos dará información sobre la función `mean` para obtener la media de una serie de valores.

Desde la opción `Ayuda` de `R Commander` se puede acceder a diversos sitios web con información sobre `R` y `R Commander`.

# Capítulo 2

# Trabajando con datos

R tiene un formato propio para crear, guardar y recuperar ficheros de datos, pero también tiene diversas opciones para importar datos de aplicaciones externas y en diferentes formatos: CSV, Excel, JSON, texto... Nosotros vamos a trabajar con ficheros de Excel en los que los datos estarán almacenados en una única hoja, donde cada observación estará almacenada en una fila distinta y cada columna será una de las variables observadas. La primera fila de la tabla contendrá los nombres de las variables.

## 2.1. Importar datos externos

Para importar un fichero de Excel deberemos tener instalada y cargada la librería `readxl`. Los comandos necesarios para ello son:

```
> install.packages("readxl")
> library(readxl)
```

y para leer el archivo `archivo.xlsx` ejecutaremos el comando:

```
> datos <- read_excel("[trayectoria/]archivo.xlsx", sheet =
    "Hoja1")
```

La expresión `[trayectoria/]` hace referencia a la carpeta en la que se encuentra ubicado el archivo dentro del sistema de carpetas del equipo. La ruta al archivo puede ser absoluta (con relación al directorio raíz) o relativa (con relación al directorio en el que estemos trabajando).

También podemos cargar el archivo desde la librería `RcmdrMisc`, de la siguiente manera:

```
> datos <- readXL("[trayectoria/]archivo.xlsx", sheet = "Hoja1",
    stringsAsFactors = TRUE)
```

La librería `RcmdrMisc` la tendremos disponible si previamente hemos cargado `R Commander`. El parámetro `sheet` no es necesario si el fichero contiene una única hoja con datos.

En `R Commander`, para importar datos desde un fichero Excel se selecciona la opción **Importar Datos** en la pestaña **Datos**, y a continuación la opción **Desde un fichero Excel** (Figura 2.1). Al hacerlo, aparece una ventana en la que podemos escribir el nombre con el que queremos identificar el conjunto de datos en **R**, así como detalles sobre cómo están almacenados los datos (Figura 2.2). Las opciones que muestra por defecto corresponden al formato de nuestros ficheros Excel, por lo que no será necesario cambiar nada en esta ventana, salvo, como mucho, poner un nombre al conjunto de datos. A continuación, una ventana de búsqueda nos permitirá buscar el archivo dentro del equipo.

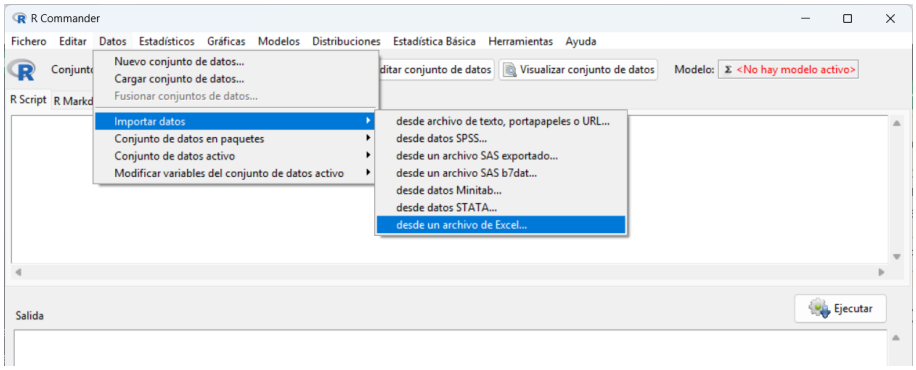



**Figura 2.1:** Importar datos de un fichero Excel.

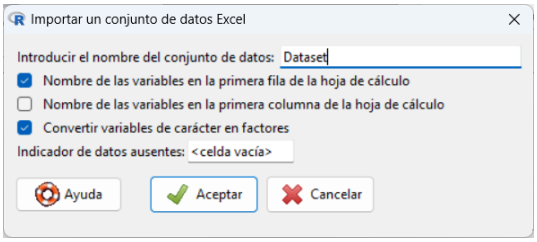



**Figura 2.2:** Formato de importación de ficheros Excel.

### 2.1.1. Importar datos desde una URL

Las librerías comentadas anteriormente no permiten la importación directa de datos de Excel desde una ubicación web: es preciso descargarse previamente el archivo y luego importarlo siguiendo las indicaciones previas.

Sin embargo, otras librerías (`openxlsx` o `openxlsx2`) sí que permiten acceder directamente a ficheros Excel a través de su URL[1]:

```
> library(openxlsx)
> cot <- read.xlsx("https://www.editdiazdesantos.com/wwwdat/
material/9788490525548/cotizaciones.xlsx")
```

```
> library(openxlsx2)
> cot <- read_xlsx("https://www.editdiazdesantos.com/wwwdat/
material/9788490525548/cotizaciones.xlsx")
```

Si se desea acceder a un fichero de datos CSV remoto, no es necesario el uso de ninguna librería adicional. La función `read.csv`, perteneciente a la distribución base de `R`, también permite acceder a ficheros CSV a través de su URL:

```
> cot<-read.csv("https://www.editdiazdesantos.com/wwwdat/
material/9788490525548/cotizaciones.csv",sep=";",
    dec=",",header=TRUE)
```

## 2.2. Guardar datos

Una vez que hemos procesado los datos, es común guardarlos en un archivo para almacenarlos de manera permanente, compartirlos o usarlos en otro programa.

Se pueden exportar en el formato de datos propio de `R` (formato .Rdata) o en diversos tipos de formato: CSV, Excel, JSON, texto...

Para exportar los datos a un fichero Excel necesitaremos tener instalada la librería `writexl`. En tal caso, bastará con ejecutar el comando:

```
> write_xlsx(datos, "[trayectoria/]archivo.xlsx")
```

También se pueden exportar a formato CSV utilizando el comando:

[1] Evidentemente, primero hay que instalar dichas librerías o paquetes siguiendo un proceso similar al descrito en el Capítulo 1 para la instalación de `R Commander`.

```
> write.csv(datos, "[trayectoria/]archivo.csv", row.names = FALSE)
```

Cuando utilicemos comandos de `R` sin referencia a ninguna librería, como es el caso de este último, entenderemos que se trata de comandos de la librería `base R`, que es el conjunto fundamental de funciones y paquetes que se instalan y cargan automáticamente al iniciar `R` y, por lo tanto, no necesitamos cargarla nosotros para usar dichos comandos.

En `R Commander` se puede exportar un conjunto de datos en formato CSV, que luego podrá ser abierto con Excel. Para ello hay que seleccionar la opción **Conjunto de datos activo** en la pestaña **Datos**, y a continuación la opción **Exportar conjunto de datos activo** (Figura 2.3). Se abrirá entonces una ventana en la que podremos elegir diversas opciones (Figura 2.4). Para exportar a CSV seleccionaremos las opciones **Escribir los nombres de las variables** y **Entrecomillar los valores de caracteres**, desmarcando la opción **Escribir los nombres de las filas**. Borraremos el contenido del campo **Valores faltantes**, dejándolo en blanco, y como **Separador de campos** elegiremos el Punto y coma [;]. Al presionar el botón OK, nos aparecerá un cuadro de diálogo en el que podremos elegir el lugar donde guardar el fichero y el nombre que le daremos; al nombre le pondremos extensión `.csv`, con lo cual podremos abrir el fichero directamente con Excel y, si lo consideramos necesario, guardarlo de nuevo en formato Excel, es decir, con extensión `.xlsx`.

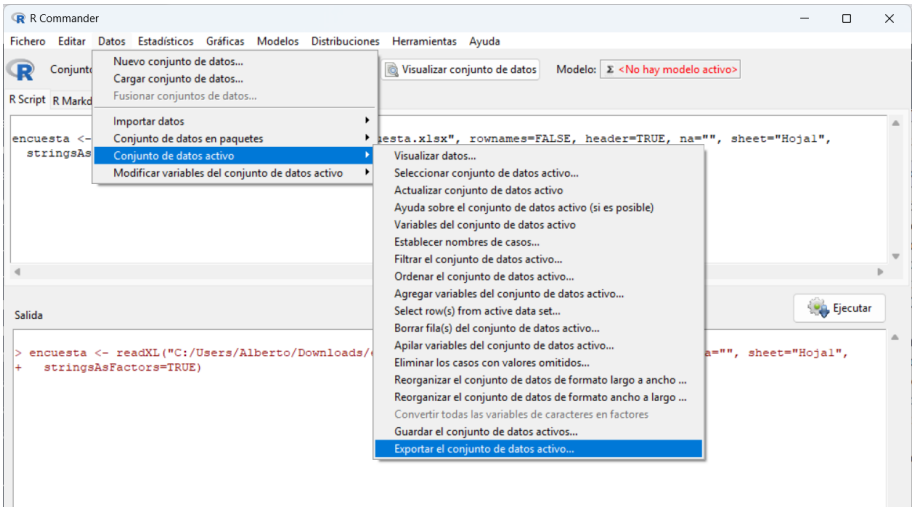



**Figura 2.3:** Exportar el conjunto de datos activo.

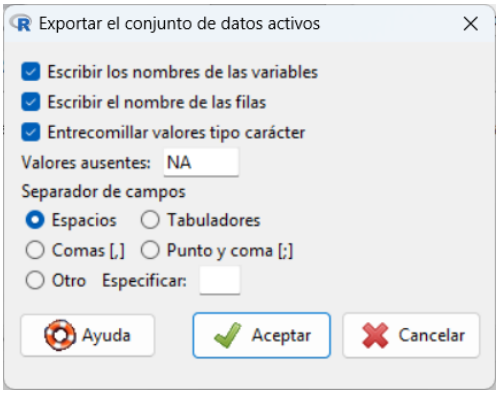



**Figura 2.4:** Exportar una tabla de datos en formato CSV.

## 2.3. Estructuras de datos

En R, los datos se pueden organizar usando distintas estructuras dependiendo del tipo de datos de que se trate y de cómo se vaya a acceder a ellos. Veamos a continuación las principales estructuras de datos.

### 2.3.1. Vectores

Un vector es una secuencia unidimensional de elementos del mismo tipo (números, caracteres, lógicos...). Los vectores tienen un nombre.

```
# Creación de un vector numérico
> edades <- c(35, 31, 19, 58, 42)

# Vector de caracteres
> nombres <- c("Ana", "Pedro", "María")

# Vector lógico (valores booleanos)
> compra_online <- c(TRUE, FALSE, TRUE, TRUE)
```

Para hacer uso de un elemento del vector, se utiliza el nombre del vector y la posición del elemento o los elementos a los que se quiere acceder:

```
> vector <- c("A", "B", "C", "D")

> vector[1]    # Primer elemento
[1] "A"
```

```
> vector[2:4]  # Elementos 2 al 4
[1] "B" "C" "D"

> vector[c(1, 3)] # Elementos 1 y 3
[1] "A" "C"
```

Se pueden efectuar operaciones matemáticas con los vectores. Las operaciones se efectúan elemento a elemento. También existen funciones que operan con vectores elemento a elemento:

```
> a <- c(1, 2, 3)
> b <- c(4, 5, 6)

> a + b
[1] 5 7 9

> a * 2
[1] 2 4 6

> sqrt(a) # Raíz cuadrada
[1] 1.000000 1.414214 1.732051
```

### 2.3.2. Matrices

Las matrices son estructuras de datos bidimensionales donde todos los elementos son del mismo tipo (numéricos, caracteres, lógicos...).

Para crear una matriz se puede usar la función `matrix()`, especificando el número de filas `(nrow)`, el número de columnas `(ncol)`, y si se llena por filas `(byrow = TRUE)` o por columnas `(byrow = FALSE)`:

```
# Crear una matriz de 2x3 (2 filas, 3 columnas)
> matriz <- matrix(
  data = c(1, 2, 3, 4, 5, 6),
  nrow = 2,
  ncol = 3,
  byrow = TRUE # la opción por defecto es FALSE
)

> matriz
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

Para acceder a un elemento de una matriz se usa su posición en la forma `[fila, columna]`:

```
> matriz <- matrix(1:6, nrow = 2, byrow = TRUE)

# Elemento en fila 1, columna 2:
> matriz[1, 2]
[1] 2

# Toda la fila 2:
> matriz[2, ]
[1] 4 5 6

# Toda la columna 3:
> matriz[, 3]
[1] 3 6

# Submatriz (filas 1-2, columnas 2-3):
> matriz[1:2, 2:3]
     [,1] [,2]
[1,]    2    3
[2,]    5    6
```

Con las matrices se pueden hacer operaciones aritméticas, de la misma forma que con los vectores:

```
> m1 <- matrix(1:4, nrow = 2)
> m2 <- matrix(5:8, nrow = 2)

> m1
     [,1] [,2]
[1,]    1    3
[2,]    2    4

> m2
     [,1] [,2]
[1,]    5    7
[2,]    6    8

> m1 + m2
     [,1] [,2]
[1,]    6   10
[2,]    8   12
```

```
> m1 * 2
     [,1] [,2]
[1,]    2    6
[2,]    4    8
```

También puede efectuarse el producto matricial de dos matrices, con el operador `%*%`:

```
> m1 %*% m2
     [,1] [,2]
[1,]   23   31
[2,]   34   46
```

### 2.3.3. Arrays

Los arrays son estructuras de datos multidimensionales que generalizan las matrices a más de dos dimensiones.

Todos los elementos de un array deben ser del mismo tipo (numéricos, caracteres, lógicos...). Se usan generalmente para representar datos en 3D o más (imágenes, datos espaciales, series temporales multidimensionales...).

Se puede crear un array mediante la función `array()`, especificando los datos y las dimensiones:

```
> # Crear un array de 2 filas, 3 columnas y 2 capas
> mi_array <- array(
    data = 1:12,  # Datos del array
    dim = c(2, 3, 2)  # Dimensiones: 2 filas, 3 columnas, 2 capas
  )

> mi_array
, , 1

     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

, , 2

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

Se puede acceder a los elementos de un array de la misma manera que se hace con los elementos de una matriz, indicando entre corchetes el índice de cada una de las dimensiones del elemento o elementos a los que se quiere acceder.

También se pueden realizar operaciones matemáticas, elemento a elemento, entre dos o más arrays de las mismas dimensiones.

### 2.3.4. Factores

Los **factores** son una estructura de datos especializada para manejar variables categóricas, es decir, variables que toman un conjunto limitado de valores (llamados **niveles**).

Se usan para variables cualitativas: color, género, nivel educativo, categorías de productos, etcétera.

Se crean con la función `factor()`:

```
> respuesta <- factor(c("Sí", "Sí", "No", "Sí"))

> respuesta
[1] Sí Sí No Sí
Levels: No Sí
```

Por defecto, los niveles se ordenan alfabéticamente, pero se puede especificar un orden arbitrario de manera manual:

```
> respuesta <- factor(
  c("Sí", "Sí", "No", "Sí"),
  levels = c("Sí", "No"),  # Orden manual
  ordered = TRUE  # Factor ordenado
)

> respuesta
[1] Sí Sí No Sí
Levels: Sí < No
```

También se pueden recodificar los niveles de un factor:

```
> respuesta_recod <- factor(respuesta, levels = c("Sí", "No"),
    labels = c("S", "N"))

> respuesta_recod
[1] S S N S
Levels: S < N
```

### 2.3.5. Listas

Las **listas** son una de las estructuras de datos más flexibles y potentes de R. A diferencia de los vectores o matrices, las listas pueden almacenar elementos de diferentes tipos (números, caracteres, vectores, matrices, otras listas...). Se usan para almacenar conjuntos de datos complejos.

Se crean con la función `list()`:

```
> ficha <- list(
  nombre = "Jazmín",
  apellidos = c("Díaz", "López"),
  edad = 25,
  es_mujer = TRUE,
  expediente = list(asignaturas = c("Matemáticas", "Estadística",
    "Economía"), notas = c(8.5, 9.0, 7.8))
)
> ficha
$nombre
[1] "Jazmín"

$apellidos
[1] "Díaz"  "López"

$edad
[1] 25

$es_mujer
[1] TRUE

$expediente
$expediente$asignaturas
[1] "Matemáticas" "Estadística" "Economía"

$expediente$notas
[1] 8.5 9.0 7.8
```

Se puede acceder a los elementos de una lista de tres maneras diferentes:

- Usando corchetes simples [ ]: el resultado es una sublista con los elementos indicados en los corchetes.
- Usando doble corchete [[ ]]: el resultado es el elemento directamente.
- Usando el nombre de la lista y el del elemento al que se quiere acceder, separados por el símbolo `$`: el resultado es el contenido de ese elemento.