

Análisis de datos en Lenguaje R

Carmen Ximénez
Javier Revuelta

Análisis de Datos en Lenguaje R

Carmen Ximénez y Javier Revuelta

EDICIONES DE LA UNIVERSIDAD AUTÓNOMA DE MADRID

28049 Madrid

Teléfono 91 497 42 33

Fax 91 497 51 69

servicio.publicaciones@uam.es

www.uam.es/publicaciones

© UAM Ediciones, 2022.

© Carmen Ximénez y Javier Revuelta, 2022.

Reservados todos los derechos. Está prohibido, bajo las sanciones penales y resarcimiento civil previsto en las leyes, reproducir, registrar o transmitir esta publicación, íntegra o parcialmente (salvo, en este último caso, para su cita expresa en un texto diferente, mencionando su procedencia), por cualquier sistema de recuperación y por cualquier medio, sea mecánico, electrónico, magnético, electroóptico, por fotocopia o cualquier otro, sin la autorización previa por escrito de Ediciones de la Universidad Autónoma de Madrid.

eISBN: 978-84-8344-830-4

DOI: <http://doi.org/10.15366/9788483448304.dt.107>

Diseño de la colección: Producción Gráfica UAM.

Prólogo

Este libro tiene como objetivo introducir al manejo del *Lenguaje de programación R* para aquellas personas que deseen tener un primer contacto sencillo con la herramienta.

El *Lenguaje R* es un software libre y gratuito. Quizá este sea su principal atractivo. Pero tiene un atractivo mayor y es que es utilizado y bien conocido por la comunidad académica y profesional a nivel mundial. Vayamos donde vayamos, las personas que saben analizar datos conocen y usan *R*. Asimismo, *R* es muy versátil y su variedad de *librerías* asociadas hace que se utilice en multitud de disciplinas (psicología, economía, ingeniería y también en otras como geografía o musicología). Estos aspectos confieren una utilidad importante a esta herramienta. De ahí que sea necesario tener un conocimiento de la misma.

El libro está redactado para los y las estudiantes del grado en Psicología que están cursando las asignaturas *Análisis de Datos I y II* en la UAM pero puede usarlo cualquier persona que quiera iniciarse de forma sencilla y rápida al manejo del *Lenguaje R*. El punto de partida del libro asume que se ha cursado las clases prácticas iniciales de las asignaturas *Análisis de Datos I y II* en las que se explica el manejo del software *SPSS* (Ximénez y Revuelta, 2011), que es uno de los más conocidos y usados en la práctica. Pero lo que se pretende introducir aquí es un software que funciona de forma diferente, y que llega al mismo resultado al que llegaba el *SPSS*. Por eso haremos paralelismos y referencia a ambas herramientas.

Es importante saber que *R* no trabaja con menús desplegables de Windows o haciendo clic en ventanas. *R* nos va a exigir, además de pensar en los análisis que queramos hacer, escribir los *comandos* que nos permitirán llevarlos a cabo. Esto requiere un cambio de mentalidad importante. En este caso la idea no es memorizar aquellos menús que nos permitan obtener resultados, lo que vamos a hacer es pensar en la sintaxis que tenemos que escribir para llegar a ese resultado concreto. Esto en principio puede resultar una tarea compleja, pero enseguida te darás cuenta de que no es así; y que ello te dará un control importante sobre lo que estás haciendo. Sin embargo, vas a tener que armarte de paciencia. Muchos comandos no te funcionarán a la primera y tendrás que repasar muchas veces lo que has escrito hasta que des con el error que hayas tenido.

Existen infinidad de manuales sobre el manejo de *R* y sobre todo tutoriales y videos donde se explican los diferentes *comandos* y *librerías* que utiliza *R*. Hay incluso foros donde puedes preguntar en línea las dudas que tengas. Debes usar todos estos recursos, pero una vez te hayas iniciado en el manejo básico de *R*, no lo hagas antes. Entonces ¿qué vas a encontrar en este libro? Pues algo parecido a lo que ya viste en nuestro libro de *SPSS* (Ximénez y Revuelta, 2011). Un material muy práctico en el que te vamos a indicar todos los pasos a seguir para aprender a usar *R* y que te va a servir como material de apoyo para las clases prácticas de las asignaturas *Análisis de Datos I y II*. Son clases que se imparten en el aula de informática y en las que resulta difícil tomar nota de todo. Por tanto, la idea es que cuando trabajes tú con *R* por tu cuenta, uses este libro, y leas despacio las explicaciones que te ofrecemos; y sobre todo, que te sientes en tu ordenador y vayas repitiendo tú los ejercicios que te proponemos.

Queremos avisarte que el libro está escrito de forma sintética y pedagógica pero, en algunos momentos, puede que encuentres algunos apartados difíciles de seguir. En los inicios, puedes obviar algunos apartados más técnicos (por ejemplo, en relación al *capítulo 1*) y volver a ellos cuando tengas más soltura con *R*, y entonces los entenderás mejor.

Queremos animarte a que te enfrentes a la tarea de aprender *R* con ilusión pero también con paciencia. Hay gran parte “autodidacta” en el aprendizaje de *R*. De hecho, las clases para introducirse a *R*, idealmente, deben impartirse mediante el procedimiento de *Flipped classroom* o aula invertida (Bergmann y Sams, 2012), de manera que dispongas de materiales para que prepares por tu cuenta cada tema (además de este libro puedes ver nuestros videos en nuestros canales de YouTube y otros materiales interactivos) antes de asistir a cada una de las sesiones presenciales, que más bien te servirán para resolver dudas y poner en práctica los comandos que vayas aprendiendo. La idea por tanto es que pases tiempo escribiendo tus propios comandos y resolviendo ejercicios prácticos. Puede que encuentres comandos diferentes a los que te proponemos aquí y que te funcionen, lo cual sería estupendo y señal de que estás aprendiendo.

Los contenidos de este libro han sido elaborados con *R* mediante el programa *Markdown*. Esto es, no han sido escritos usando un editor de textos clásico, como Word, sino usando un lenguaje de programación. Esto nos ha permitido mostrarte tanto los comandos como los resultados de algunos de los ejemplos que hemos incluido con el mismo aspecto y color en que aparecerán en tu pantalla del ordenador, lo que sin duda facilitará tu aprendizaje, sobre todo en los inicios, y también tu familiarización con la sintaxis empleada.

No podemos terminar este prólogo sin agradecer a la UAM por habernos concedido el *Proyecto de Innovación Docente PS_001.19_INN* de la convocatoria del curso 2019-20, y el *Proyecto Implanta PS_004_20_IMP* de la convocatoria del curso 2020-21, gracias a los cuales hemos podido formar un equipo de trabajo que nos ha permitido elaborar este libro. El equipo está formado por profesores de *Análisis de Datos* del área de Metodología de la Facultad de Psicología de la UAM, así como por un grupo de estudiantes que se unió a nuestro equipo. Gracias a los profesores Juan Botella, Manuel Suero, Eduardo García-Garzón y Ricardo Olmos, este libro incluye contenidos que nosotros no hubiéramos pensado y también hemos evitado varias erratas. Queremos agradecer muy especialmente también a nuestras estudiantes del equipo, Gádor Rubias, Claudia Rodríguez, Alba Claudio y Liz Mendoza por todo el feedback que nos han dado durante este tiempo y por su paciencia a la hora de revisar todos los materiales que se presentan aquí, así como por “darnos el punto de vista del estudiante” y hacernos ver la mejor manera de explicar las cosas. Y por supuesto a nuestros/as estudiantes de grado de primer y segundo curso de las promociones 2017-18 hasta 2020-21 que han sufrido nuestras clases, y con sus comentarios y preguntas, han contribuido mucho más de lo que puedan imaginar en la redacción final del libro. Esperamos poder recibir mucho más feedback de futuros estudiantes que nos vayan marcando el camino y hagan mejorar este libro, cuyos contenidos, inevitablemente, quedarán obsoletos pronto y habrán de actualizarse.

Los autores

Índice de Contenidos

Introducción	1
1. Familiarizarse con R	3
1.1. Qué es el <i>Lenguaje R</i>	3
1.2. Instalación de <i>R</i> y <i>RStudio</i>	6
1.3. Estructura de <i>RStudio</i>	10
1.4. Cómo trabaja <i>R</i>	12
1.4.1. Los comandos y el argumento	12
1.4.2. Crear objetos	13
1.4.3. Operaciones con objetos	15
1.4.4. Vectores	16
1.4.5. Funciones	17
1.5. El Script	18
1.5.1. Vincular carpeta	19
1.5.2. El concepto de <i>Librería de R</i>	20
1.6. Manejo de datos	21
1.6.1. Tipos de datos básicos	21
1.6.2. Estructuras de datos	23
1.6.2.1. Datos de una dimensión	23
1.6.2.2. Datos de dos o más dimensiones	28
1.7. Leer datos de archivos externos	31
1.7.1. Datos de texto	31
1.7.2. Datos en SPSS	32
1.7.3. Datos CSV	33
1.7.4. Datos en Excel	33
1.8. Los gráficos en <i>R</i>	34
1.9. Ejercicios propuestos	37
2. Análisis descriptivos	39
2.1. Iniciar la sesión con <i>RStudio</i> y preparar Script	39
2.2. Análisis descriptivos univariantes	40
2.2.1. Tablas de frecuencias	42
2.2.2. Gráficos	43
2.2.3. Estadísticos	45
2.2.4. Transformación de puntuaciones	50
2.2.5. Seleccionar casos	52
2.3. Análisis descriptivos bivariantes	56
2.3.1. Covarianza y correlación	56
2.3.2. Gráficos de dispersión	56
2.3.3. Matriz de varianzas-covarianzas y matriz de correlaciones	57
2.3.4. Puntuaciones combinadas	60

2.3.5. Regresión lineal simple	61
2.3.6. Tablas de contingencia	64
2.4. Ejercicios propuestos	66
3. Probabilidad: introducción a los modelos de distribución	69
3.1. Conceptos previos	69
Función de probabilidad y función de densidad de probabilidad	69
Función de distribución	70
Función de distribución inversa	70
3.2. Distribuciones incluidas en R	70
3.3. Modelo Binomial	71
3.4. Modelo Normal	74
3.5. Modelo Chi-cuadrado de Pearson	77
3.6. Modelo t de Student	79
3.7. Modelo F de Snedecor	80
3.8. Simular datos	83
3.9. Ejercicios propuestos	85
4. Contrastes de hipótesis sobre uno y dos parámetros	87
4.1. Lectura y preparación de datos del archivo <i>terapia.dat</i>	87
4.2. Prueba Z para una media	88
4.3. Prueba T sobre una media	89
4.4. Contraste sobre dos medias y dos varianzas independientes	90
4.5. Contraste sobre dos medias relacionadas	92
4.6. Contraste sobre una correlación	94
4.7. Ejercicios propuestos	96
5. Contrastes sobre proporciones	97
5.1. Contraste sobre una proporción	97
5.1.1. Prueba binomial	97
5.1.2. Contraste mediante la aproximación normal	98
5.2. Contraste sobre dos proporciones independientes	99
5.3. Contraste de bondad de ajuste	100
5.4. Contraste de independencia en tablas de contingencia	101
5.4.1. Ejemplo con dos proporciones independientes	101
5.4.2. Caso general	102
5.4.3. Prueba exacta de Fisher	103
5.4.4. Índices de asociación entre variables categóricas	103
5.4.4.1. Índice de riesgo	104
5.4.4.2. Razón de ventajas	105
5.4.4.3. Medidas de asociación	107
5.5. Prueba de homogeneidad marginal o de McNemar	108
5.6. Ejercicios propuestos	109

6. Análisis de correlación y regresión lineal	111
6.1. Análisis de correlación	111
6.2. Análisis de regresión	114
6.2.1. Regresión lineal simple	114
6.2.2. Regresión lineal múltiple	117
6.2.2.1. Modelo básico	117
6.2.2.2. Regresión por pasos	118
6.2.3. Regresión lineal múltiple multivariante	120
6.2.4. Regresión no lineal	120
6.3. Ejercicios propuestos	124
 7. Análisis de varianza de un factor y comparaciones múltiples	 125
7.1. Análisis de varianza de un factor	125
7.1.1. Análisis descriptivos	126
7.1.2. Tabla de ANOVA	127
7.1.3. Medidas del tamaño del efecto	128
7.2. Comparaciones múltiples	130
7.2.1. Comparaciones a-posteriori	130
7.2.2. Comparaciones planeadas o a-priori	131
Comparaciones de tendencia	132
Comparaciones planeadas ortogonales	134
7.3. Análisis de varianza con medidas repetidas	136
7.3.1. Organización de los datos	136
7.3.2. Análisis descriptivos	138
7.3.3. Tabla de ANOVA	138
7.4. Ejercicios propuestos	141
 8. Análisis de varianza de dos factores	 143
8.1. Diseño inter-sujetos	143
8.1.1. Preparación de los datos	144
8.1.2. Análisis descriptivos	144
8.1.3. Tabla de ANOVA	144
8.1.4. Medidas del tamaño del efecto	146
8.1.5. Representación gráfica de la interacción	148
8.1.6. Comparaciones múltiples	148
8.2. Diseño intra-sujetos o de medidas repetidas	150
8.2.1. Preparación de los datos	150
8.2.2. Tabla de ANOVA	152
8.3. Diseño mixto	154
8.3.1. Preparación de los datos	154
8.3.2. Tabla de ANOVA	155
8.3.3. Representación gráfica de la interacción	156
8.4. Análisis de covarianza	158
8.5. Ejercicios propuestos	162

9. Contrastes no paramétricos	163
9.1. Prueba de Kolmogorov-Smirnov	163
9.2. Prueba de los signos (binomial)	164
9.2.1. Contraste sobre una mediana	165
9.2.2. Comparación de la mediana de dos variables	165
9.3. Prueba de las rachas	166
9.4. Prueba de Mann-Whitney	168
9.5. Prueba de Kruskal-Wallis	169
9.6. Prueba de Friedman	169
9.7. Ejercicios propuestos	171
 Referencias	 173
 Anexos	 175
Anexo 1. Archivo de datos <i>practica.sav</i>	177
Anexo 2. Tabla resumen comandos <i>R</i> (estadística univariada)	178
Anexo 3. Tabla resumen comandos <i>R</i> (estadística bivariada)	179
Anexo 4. Tabla resumen comandos <i>R</i> (probabilidad)	180
Anexo 5. Archivo de datos <i>terapia.dat</i>	181
Anexo 6. Tabla resumen comandos <i>R</i> (inferencia estadística)	183
Anexo 7. Librerías de <i>R</i> más usadas en el Análisis de datos en Psicología	185

Introducción

Este libro contiene una introducción al lenguaje de programación *R* destinada a los y las estudiantes de las asignaturas *Análisis de Datos I* y *Análisis de Datos II* del Grado en Psicología en la Universidad Autónoma de Madrid. A día de hoy, el lenguaje *R* es el sistema informático universal para realizar todo tipo de análisis estadísticos. Agotar todas sus posibilidades es virtualmente imposible debido a la existencia de miles de librerías para procesar y analizar datos en infinitud de ámbitos. Esto se debe a que *R* está diseñado para que cualquier profesional pueda desarrollar librerías y compartirlas, lo que lo ha convertido en el estándar en el que se reflejan los nuevos avances en análisis de datos y psicometría. Además, la emergencia de las tecnologías de la información y su confluencia con la estadística, en los campos conocidos como *Big Data* y *Machine Learning*, no ha hecho sino aumentar el interés en el lenguaje *R*, que constituye una parte indispensable del bagaje de conocimientos necesarios para trabajar en estas áreas.

Al ser un lenguaje de programación, o más exactamente un lenguaje de scripts, *R* tiene una curva de aprendizaje más lenta para el/la estudiante de las ciencias sociales que otros sistemas informáticos, como *SPSS*, basados en ventanas. Para manejar *R* es necesario tener un conocimiento básico de las estructuras de datos en que se almacenan los datos en la memoria del ordenador, el concepto de función informática y su codificación. En este libro se ha tratado de realizar una introducción a dichos aspectos lo más sencilla posible. Al mismo tiempo se han obviado todos los conceptos relativos a algoritmos informáticos, como los bucles, que son indispensables para la utilización de *R* en estudios de simulación pero que no aparecen sino en cursos más avanzados. El propósito ha sido ceñirse fielmente a los contenidos de análisis de datos que son propios de las asignaturas del grado en Psicología, así como a los aspectos de estructuras de datos y programación imprescindibles para realizar estos análisis.

Con ello, este manual pretende servir de introducción al lenguaje *R* partiendo desde cero, de modo que permita al estudiante realizar de forma autónoma todos los análisis aquí contenidos al tiempo que sirva de primer paso para iniciarse en este lenguaje para quienes quieran orientarse hacia el ámbito de la estadística aplicada, cursar un posgrado en temas de Metodología y Psicometría, o simplemente llevar a cabo los análisis de datos propios de una investigación empírica.

1 Familiarizarse con R

En este capítulo se pretende dar al lector una explicación básica para facilitar su primer contacto con el *Lenguaje R*.

El *Lenguaje R* funciona de forma muy diferente a otros softwares estadísticos muy conocidos, como el programa *SPSS*. La mayor diferencia estriba en que, en lugar de utilizar menús de Windows que se manejan de forma sencilla, *R* utiliza códigos de sintaxis que se necesita escribir y ejecutar para poder llegar a unos resultados. Esto es, *R* requiere *escribir comandos* y funciona como **lenguaje de programación**. Esto puede hacer difícil el primer contacto con *R*. Sin embargo, una vez te familiarices con la tarea, enseguida descubrirás que es muy sencillo escribir comandos y ejecutarlos para llegar a un resultado concreto, ya sea un análisis de datos o la elaboración de un gráfico.

El programa *R* utiliza un interfaz muy diferente al que estamos acostumbrados a ver en otros programas de Windows (véase Figura 1.11). Para trabajar con *R* nos tenemos que acostumbrar a escribir y ejecutar una serie de comandos, lo que nos permitirá trabajar con mucha autonomía y control sobre los análisis estadísticos concretos que queramos llevar a cabo.

1.1. Qué es el Lenguaje R

R es un software matemático que fue desarrollado inicialmente por Robert Gentleman y Ross Ihaka del Departamento de Estadística de la Universidad de Auckland (Nueva Zelanda) en 1990. Sin embargo, el proyecto se inició en los Bell Laboratories de AT&T en Nueva Jersey con el **Lenguaje S** en el año 1970.

Su desarrollo actual es responsabilidad del *R Development Core Team* (2019), cuyo Website se encuentra en: www.r-project.org

Principales características del lenguaje R

- *R* es un lenguaje y entorno de programación para análisis estadístico y gráfico.
- Se trata de un proyecto de software libre, resultado de la implementación del *Lenguaje S*. Probablemente, *R* es el lenguaje más utilizado en investigación por la comunidad estadística. A esto contribuye la posibilidad de cargar diferentes **Librerías** o **Paquetes** con finalidades específicas de cálculo o gráfico.

1 Familiarizarse con R

- *R* se distribuye bajo la licencia GNU GPL y está disponible para los sistemas operativos Windows, Macintosh, Unix y GNU/Linux.
- *R* se trata de un lenguaje de programación que es muy potente y con una sintaxis muy sencilla e intuitiva de aprender. No se necesitan conocimientos de otros lenguajes de programación para poder usar *R*.
- *R* proporciona un abanico de herramientas estadísticas (e.g., modelos lineales y no lineales, test estadísticos, algoritmos de clasificación y agrupamiento, etc.) y gráficas.
- *R* puede integrarse con distintas bases de datos y existen *librerías* que facilitan su utilización (ver Anexo 7).
- *R* permite generar gráficos con alta calidad (véase a modo de ejemplo los gráficos mostrados en las Figuras 1.1 y 1.2).
- *R* también puede usarse como herramienta de cálculo numérico y puede ser tan eficaz como otras herramientas tales como MATLAB.
- *R* forma parte de un proyecto colaborativo y abierto. Sus usuarios pueden publicar **Librerías** (también denominadas **paquetes**, del inglés **package**) que extienden su configuración básica. Las librerías están organizadas por temas.

En resumen, *las ventajas del lenguaje R* son las siguientes:

- *R* es un lenguaje de programación eficaz que permite hacer todo tipo de análisis estadísticos.
- Aunque en un principio *R* puede resultar poco amigable, ya que requiere redactar códigos, su sintaxis es relativamente fácil de aprender.
- *R* permite combinar fácilmente librerías de *código R* y de otros programas y ofrece gráficos de alta calidad.
- *R* es gratuito. Por tanto, se utiliza sin ningún coste. Esto hace que su alcance sea mucho mayor que el de cualquier otro programa comercial.
- *R* se está actualizando permanentemente, con las aportaciones de la comunidad científica de expertos en Estadística. Esto obliga al usuario a actualizar el programa de forma permanente también en su equipo.
- *R* tiene un **Menú de Ayuda** muy completo. Además, en la web pueden encontrarse múltiples tutoriales y videos donde se explica el manejo de muchas de las funcionalidades y librerías del programa.
- *R* no dispone de un soporte comercial pero se actualiza permanentemente por miles de usuarios en todo el mundo, lo que permite contactar online y consultar directamente con el autor o autora de la **función** o **librería** en cuestión, aquellas dudas que tengamos.

```
x = seq(-10,10,length=50) # Generamos una malla
                             de puntos (x,y)
y = x
f = function(x,y){ x^2 - y^2 } # Definimos la función
                                que dibujaremos
z = outer(x,y,f)               # La función outer
                                evalua la función f en
                                cada punto(xi,yj)
persp(x,y,z,theta=30,phi=30) # Un gráfico en
                                perspectiva
```

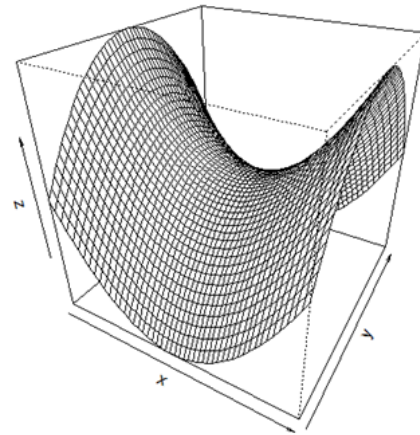


Figura 1.1: Ejemplo de gráfico hecho con R

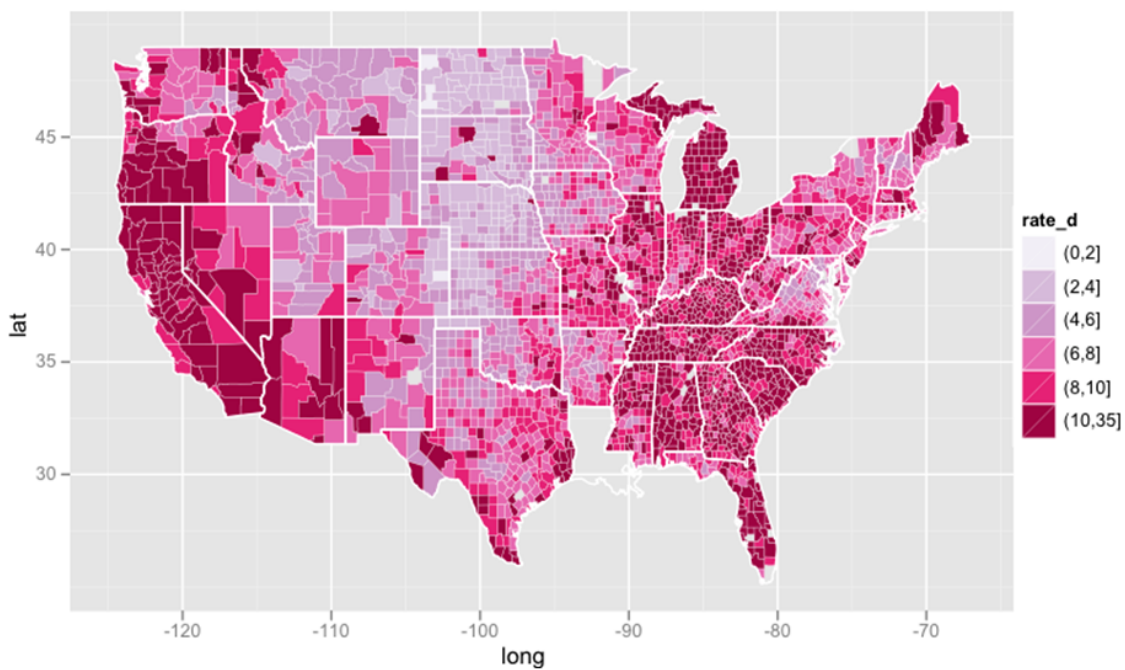


Figura 1.2: Ejemplo de gráfico hecho con R

1.2. Instalación de R y RStudio

Para trabajar con *R* lo primero que se necesita es instalar el **Programa R base** desde CRAN. Después se instala el **Programa Rstudio**, que consiste en un editor de textos para *R* que hace mucho más sencillo su manejo. A continuación se detallan los pasos para poder completar la instalación de ambos programas.

Para instalar el **Programa R base** vamos al enlace www.r-project.org cuya ventana se muestra en la Figura 1.3. A continuación, pulsamos **Download R** y esto nos lleva a la ventana de la Figura 1.4. A continuación pulsamos la opción **Spain**, lo que nos conduce a la ventana de la Figura 1.5. El programa funciona en varios sistemas operativos (aquí veremos cómo hacerlo para Windows pero puede instalarse también en MAC y bajo Linux). Elegimos la opción **Windows** y el enlace nos lleva a la ventana de la Figura 1.6, que nos permite iniciar la instalación.

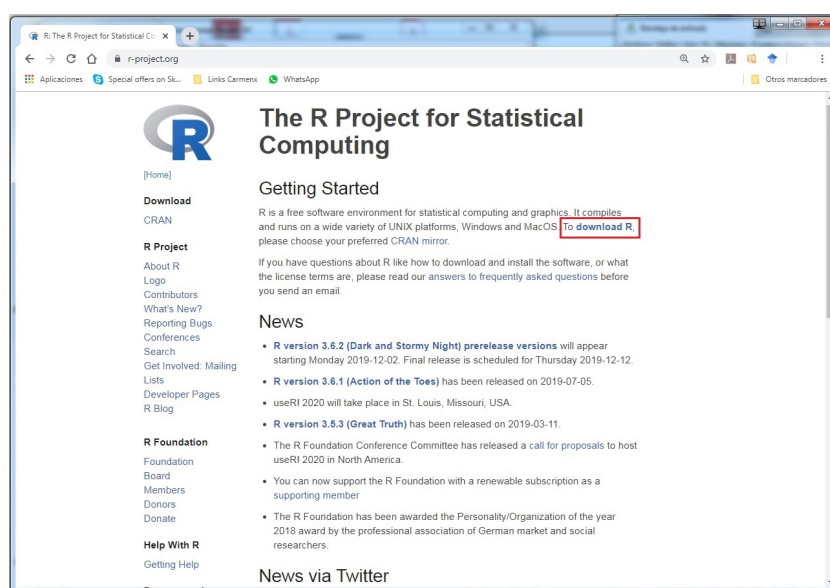


Figura 1.3: Página web para instalar el Programa R base

Por último, como muestra la Figura 1.7, instalaremos la última versión disponible para *R*. En el momento de redactar estos apuntes era la **versión 4.1.2 para Windows**, pero lógicamente se irá renovando con el paso del tiempo.

Una vez instalado el *Programa R*, a continuación instalaremos el **Programa RStudio**. Para ello, iremos al enlace www.rstudio.com tal y como se muestra en la ventana de la Figura 1.8.

De las opciones ofrecidas en la Figura 1.8, elegiremos la opción **Download Rstudio**. Después se abre la ventana que aparece en la Figura 1.9. Desde aquí se selecciona la opción **Free Download** de la Figura 1.9. y se abre la ventana de la Figura 1.10., desde donde se puede iniciar ya la instalación de la versión de RStudio para Windows.

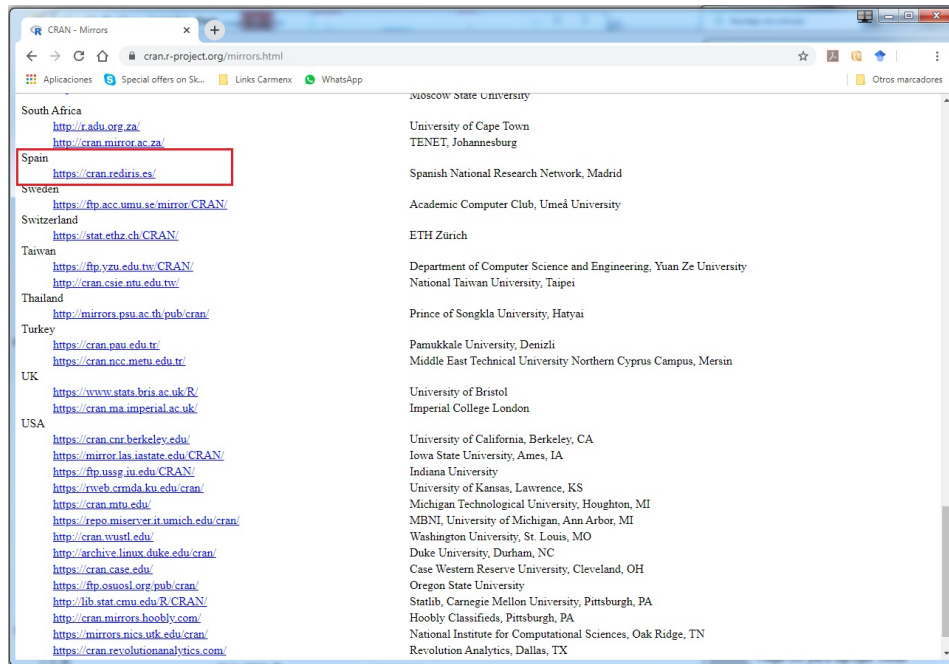


Figura 1.4: Paso 1 para instalar el Programa R base

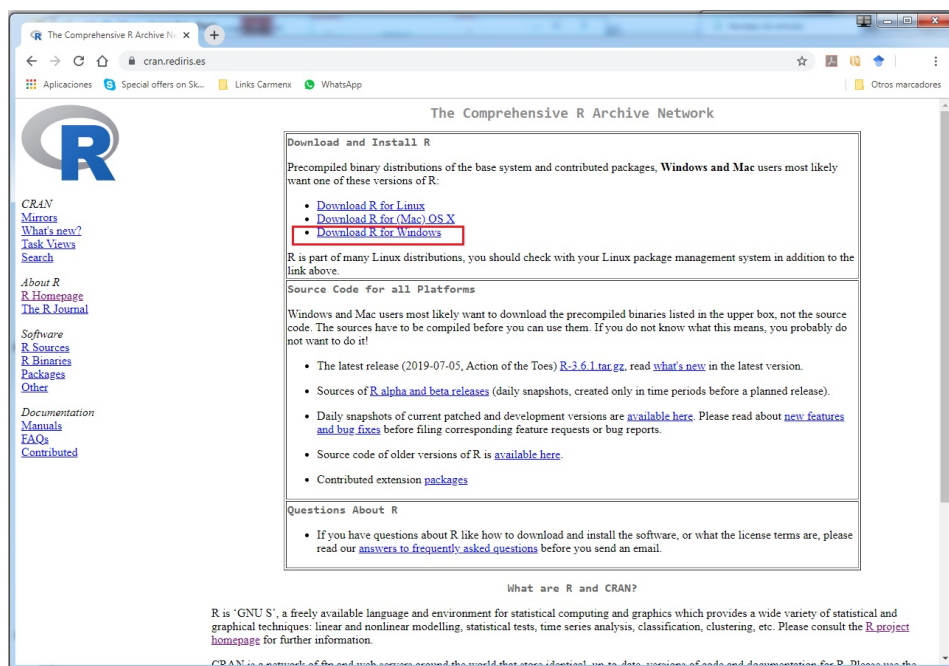


Figura 1.5: Paso 2 para instalar el Programa R base

1 Familiarizarse con R

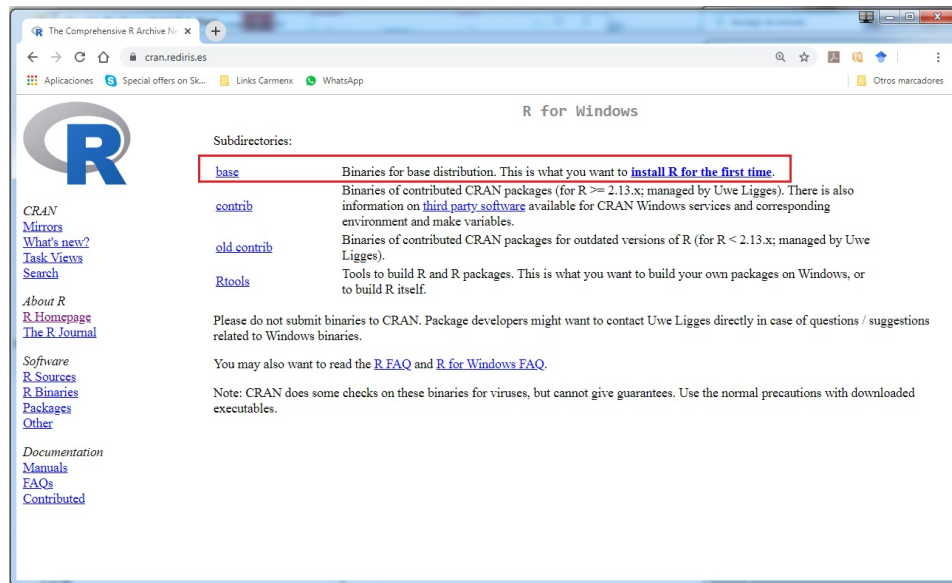


Figura 1.6: Paso 3 para instalar el Programa R base

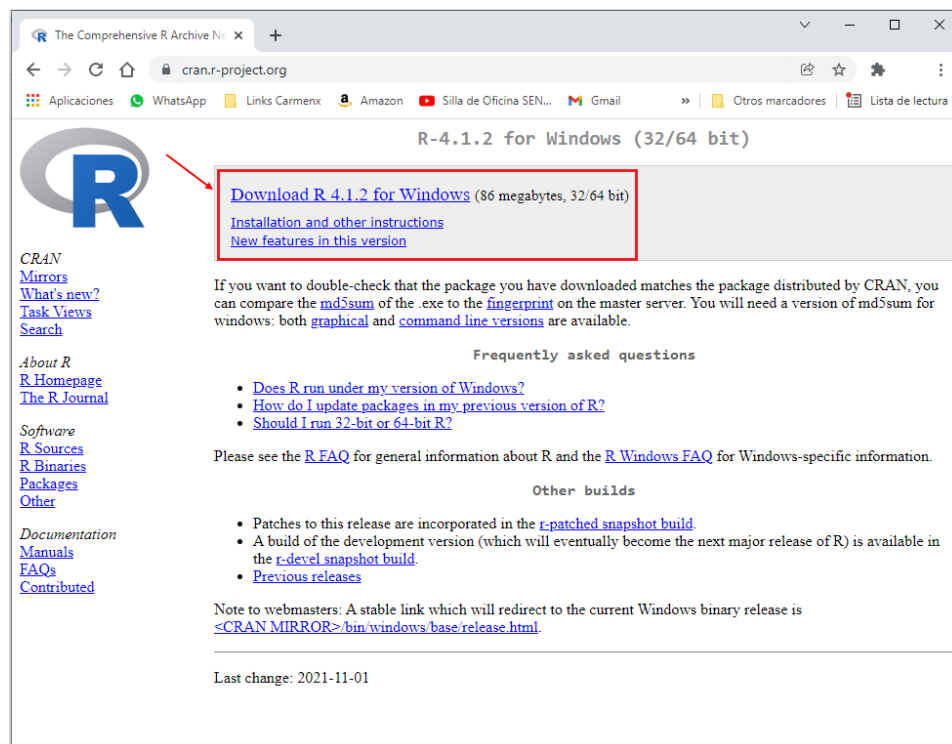


Figura 1.7: Paso 4 para instalar el Programa R base

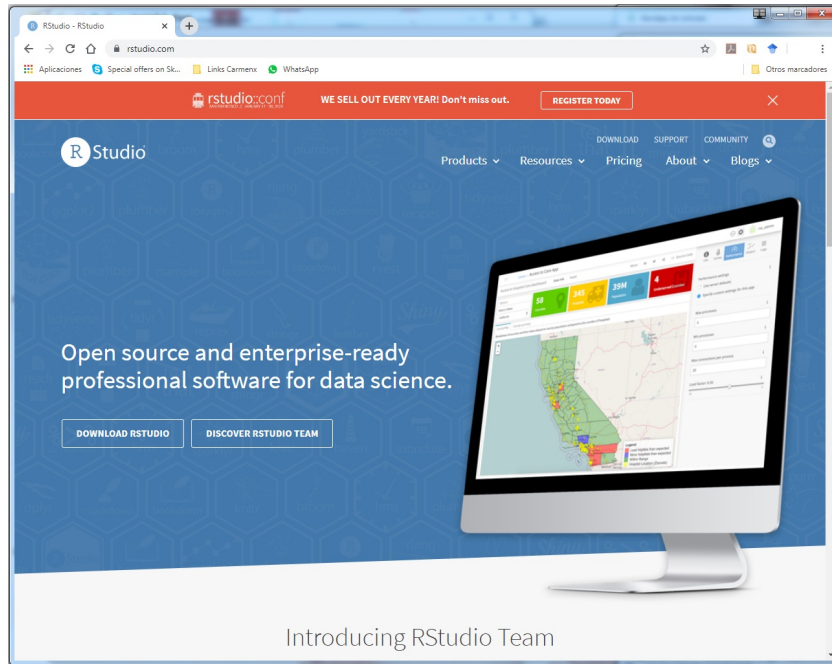


Figura 1.8: Página web para instalar el Programa RStudio

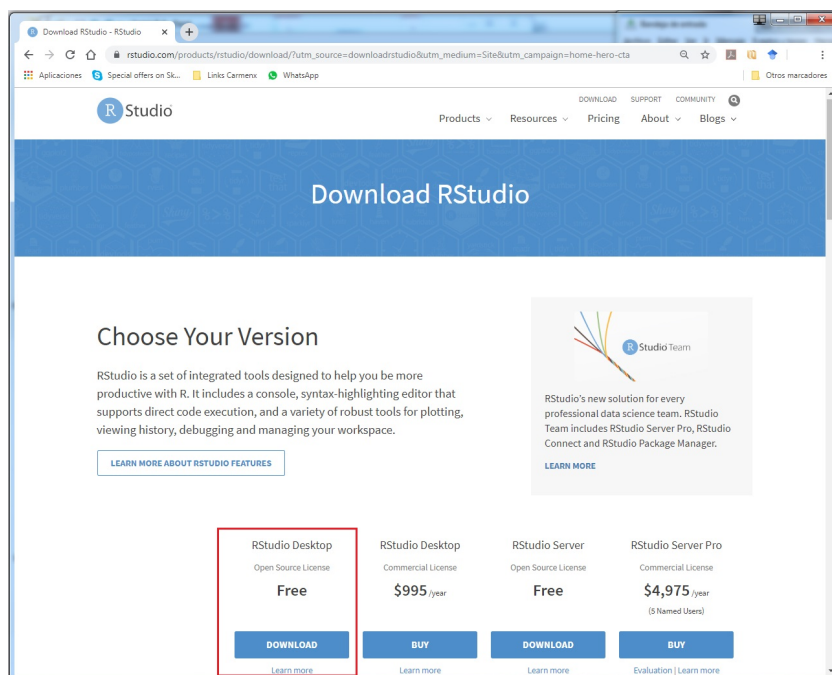


Figura 1.9: Paso 1 para instalar el Programa RStudio

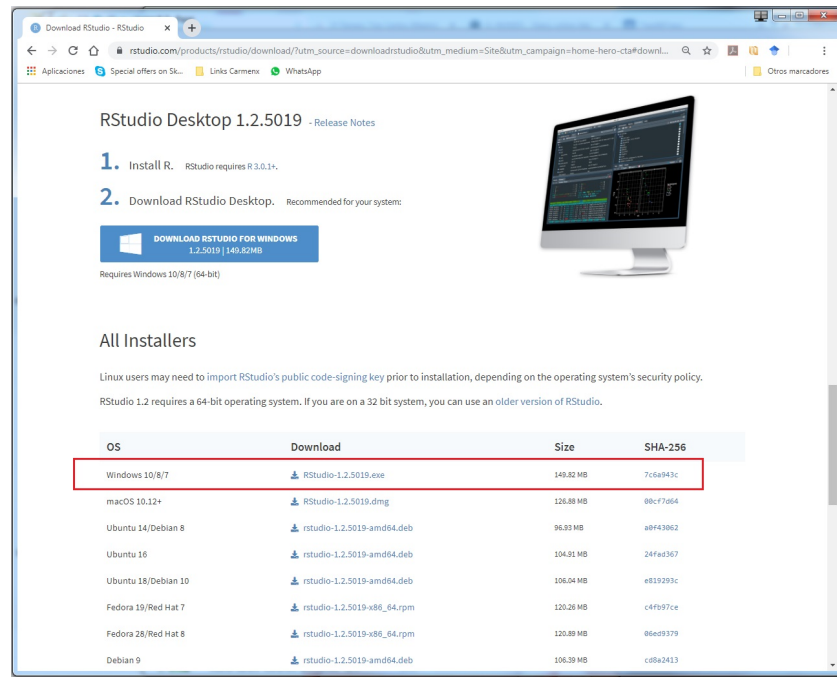



Figura 1.10: Paso 2 para instalar el Programa RStudio

Ya podemos empezar a trabajar con *RStudio*. Automáticamente se habrá creado un icono en tu escritorio desde el que podrás acceder al programa *RStudio*. En el siguiente apartado se explica el entorno y estructura de *RStudio*.

1.3. Estructura de RStudio

Para entrar en *RStudio* hacemos doble clic en el icono  que debes tener en tu escritorio.

Una vez entramos en el programa *RStudio* aparece la ventana que se muestra en la Figura 1.11. Como puede verse, el programa tiene cuatro ventanas:

1. **La Ventana 1** de la Figura 1.11 se denomina **Área de Trabajo** o *Source*. En esta ventana es donde vamos a escribir los comandos que nos permitirán llevar a cabo (o *correr*, que viene de *Run*) análisis estadísticos y solicitar gráficos. Desde esta ventana generaremos el denominado *SCRIPT* (véase apartado 1.5), que guardaremos en un archivo con la extensión *.R.
2. **La Ventana 2** de la Figura 1.11 se denomina **Consola** o *área de resultados*. La principal utilidad de esta ventana es la de mostrar los resultados de los análisis aunque también puede utilizarse para escribir y ejecutar comandos. La ventana 2 es equivalente al *Programa R base*.

3. **La Ventana 3** de la Figura 1.11 se denomina **Memoria de Trabajo**. Desde aquí se van grabando los comandos que vamos escribiendo (**History**) para realizar operaciones (por ejemplo: variables generadas, datos leídos, etc.) de forma que pueden ejecutarse de nuevo las acciones realizadas. Esto puede grabarse para abrirlo y recuperar la información que se tenía en una sesión previa. Desde la pestaña **History** se pueden enviar estos comandos a ‘To Source’ (a la ventana de comandos) o a ‘To Console’ (a la ventana de resultados).
4. **La Ventana 4** de la Figura 1.11 se denomina **Ventana de ayuda y resultados gráficos** e incluye varias solapas. La solapa **Files** muestra los archivos con los que hemos ido trabajando, por orden. La solapa **Plots** muestra los gráficos que hemos solicitado desde el SCRIPT en el Área de trabajo. La solapa **Packages** muestra las librerías de *R* que tenemos instaladas en nuestro ordenador y permite también su actualización. La solapa **Help** permite solicitar ayuda sobre los diferentes comandos del programa. La solapa **Viewer** nos permite ver contenido web local. Por ejemplo, gráficos web generados usando librerías específicas de *R* (e.g., googleVis, htmlwidgets y rCharts).

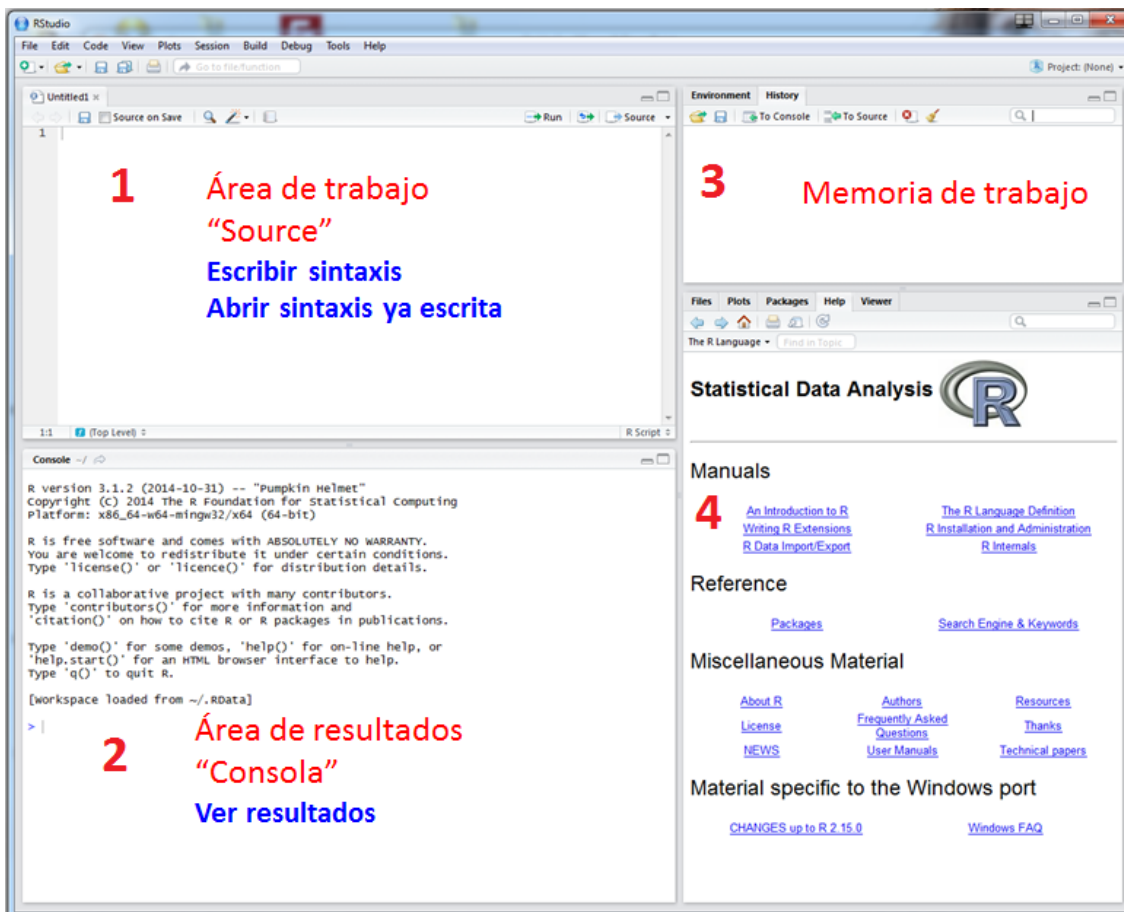




Figura 1.11: Estructura de RStudio

1.4. Cómo trabaja R

1.4.1. Los comandos y el argumento

R es un lenguaje de programación y difiere de otros softwares estadísticos comerciales en que R trabaja mediante **comandos**. Para ello, utiliza el **objeto** como *entidad básica*. Esto es, cualquier expresión evaluada por R tiene como resultado un **objeto**.

Los comandos se escriben en el editor de archivos (*Ventana 1* de la Figura 1.11). Como resultado, se genera un programa informático (denominado SCRIPT), que puede tener cualquier longitud y se ejecuta al pulsar el botón  **Run**. Para la ejecución de los comandos, se necesita tener posicionado el cursor en la línea exacta donde esté escrito el comando. Si queremos ejecutar el programa completo, se puede pulsar  **Source**.

Vamos a escribir nuestro primer comando. Por ejemplo, vamos escribir en el terminal:

```
> print("Hola UAM")
```

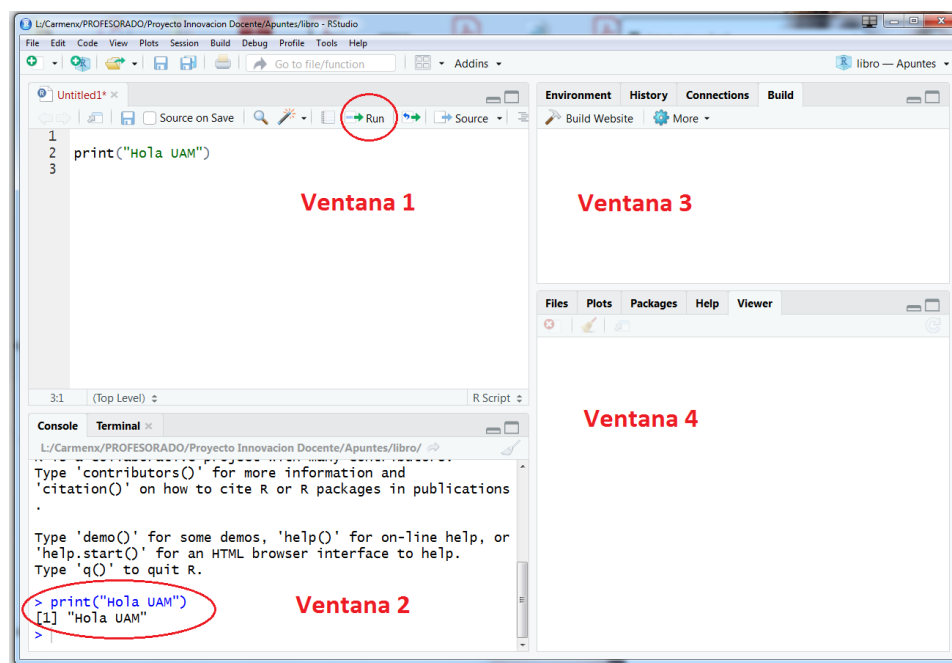


Figura 1.12: Ejemplo de comando Print

Como puede verse en la Figura 1.12, hemos escrito el comando en la línea 2 de la *Ventana 1*. El comando que hemos escrito es `print()` y lo que lleva dentro del paréntesis se denomina **argumento** (en este caso “Hola UAM”). Para ejecutar el comando, es importante dejar el cursor donde hayamos escrito el comando (en este caso en la línea

2), y a continuación pulsamos el botón **Run** (está en la parte superior derecha de la *Ventana 1*, en la Figura 1.12, rodeado con un círculo rojo). Al ejecutar este comando el resultado se muestra en la *Consola* o *Ventana 2* tal y como puede verse en la Figura 1.12 (aparece rodeado con un círculo abajo a la izquierda).

En el ejemplo anterior hemos escrito un **comando** (`print()`) que incluía en el **argumento** un texto (entre comillas). Pero habitualmente trabajaremos con números. Uno de los usos más frecuentes de *R* es como calculadora. Veamos un ejemplo sencillo donde se propone la suma de los valores 2 y 3:

```
> 2+3
[1] 5
```

El cuadro anterior muestra el comando empleado (`> 2+3`) y el resultado obtenido en la *Consola* al correr este comando (`[1] 5`). En lo sucesivo, mostraremos ejemplos similares a este para que puedas comprobar el resultado que debe salir en tu *Consola* al correr ciertos comandos.

Lo que hemos hecho en los ejemplos anteriores al escribir comandos se denomina **Crear objetos**. En el siguiente apartado veremos la definición de **Objeto** en más detalle.

1.4.2. Crear objetos

Como ya se ha señalado, *R* es un lenguaje de programación y, como tal, se necesita definir todas las operaciones que queramos llevar a cabo. Para ello, necesitaremos manejar variables denominadas **Objetos**.

Los **objetos** pueden tener el contenido que queramos. Entre otros:

- Un dato (por ejemplo, el número 2).
- Múltiples datos (por ejemplo: 1, 2, 3, 4, etc.)
- Nombres y cadenas ("Hola UAM", como en el ejemplo de la Figura 1.12).
- Bases de datos (e.g., un fichero *SPSS*).
- Conjuntos de bases de datos. etc.

Cada **objeto** recibe un nombre y es guardado en la memoria de *R* (*Ventana 3* de la Figura 1.11) para ser reutilizado posteriormente. Por tanto, un paso importante a realizar implica “*dar contenido al objeto*”. Por ejemplo, si creamos el objeto `x` y queremos definir que `x = 10`, esto sería dar contenido a nuestro objeto (en este caso, un valor numérico).

En *R* se utiliza una flecha `<-` o el signo `=` para crear objetos y darles contenido.

Veamos un nuevo ejemplo.

Crearemos los objetos **a**, **b** y **suma**:

```
a <- 1  
b = 2  
suma = a + b
```

a y **b** incluyen números. Para definir **a**, que es el número 1, se ha utilizado la expresión `<-`; mientras que para definir **b** se ha utilizado el signo `=`. Ambos operadores son equivalentes. Teniendo definidos los objetos **a** y **b**, puede definirse el objeto **suma** que es la operación **a** + **b**. Si más adelante queremos utilizar el objeto **suma**, podremos llamarlo y que forme parte de otro objeto diferente. Podría, como en el ejemplo inferior referido al objeto **c**, ser el numerador de un cociente:

```
c <- suma / b
```

Tenemos que tener en cuenta que en *R* lo que no se guarde en un **objeto**, se pierde. Por tanto, es necesario mentalizarse de que el trabajo con *R* requiere estar permanentemente definiendo objetos. Esto es, en lugar de trabajar con menús y ventanas, como se haría con el programa *SPSS*, definiremos objetos y haremos operaciones con ellos.

También es importante tener en cuenta que *R* distingue entre letras mayúsculas y minúsculas. En el siguiente ejemplo, obsérvese que los objetos **d** y **D**, pese a tener la misma letra, son distintos:

```
d <- 8  
D <- 9
```

Si usamos el mismo nombre para un objeto que ya existe, *se sobrescribe*. En el ejemplo, al ejecutar los siguientes comandos:

```
f <- 5  
f <- 6
```

el objeto **f** termina tomando el valor 6.

Iremos familiarizándonos con este lenguaje a medida que vayamos escribiendo comandos y veamos cómo se muestran los resultados en las ventanas 3 y 4 de la Figura 1.11. Sobra decir que si cometemos algún error al escribir comandos, se mostrará una indicación del error en la línea en que éste se encuentre. Y en la *Ventana 3* de resultados aparecerá un **Warning message** con la descripción del error. Normalmente este tipo de mensajes se muestran en color rojo, pero este color no siempre indica que haya un error. Esto es, si aparece un mensaje en color rojo en la *Ventana 3* de la Figura 1.11, no hay que alarmarse, ya que no necesariamente significa que sea un error.

1.4.3. Operaciones con objetos

Las operaciones que pueden realizarse con objetos en *R* son muy diversas. Entre las que nos interesan se encuentran las siguientes:

Operación	Notación
Suma	+
Resta	-
Producto	*
División	/
Exponente	^
Raíz cuadrada	sqrt
Producto matrices	%*%

A continuación veremos ejemplos de sintaxis acerca de las operaciones de la tabla anterior y de sus resultados, que se muestran con el nombre del objeto seguido de la expresión `## [1]`, que es el resultado obtenido al ejecutar cada comando con **Run**:

```
a <- 1
b <- 3
r <- a + b
s <- b - a
p <- s*r
q <- sqrt(p)
r
```

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

```
s
```

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

```
p
```

```
## [1] 8
```

```
q
```

```
## [1] 2.828427
```

1.4.4. Vectores

Los **vectores** son objetos que representan un conjunto de elementos del mismo tipo. Lo más habitual es que los elementos sean un conjunto de números, aunque también se pueden incluir letras u otros tipos de elementos.

Para crear un objeto o vector que contenga series de números, puede usarse el comando `c()`, que significa **conjunto** o **concatenar**. Los elementos incluidos en la serie numérica han de ir separados por comas.

En el ejemplo siguiente se crean dos tipos de series de objetos. Uno con **nombres** (denominado: **nom**) y otro con **números** (**x**):

```
nom <- c("Manuel","Eduardo")
x <- c(1,4,3,8)
```

También es posible crear una secuencia de números utilizando el comando `1:N`. En el siguiente ejemplo, referido a **num1** y **num2**, con cinco datos, podríamos definirlo de dos maneras que, como se observa, llevan al mismo resultado:

```
num1 <- c(1, 2, 3, 4, 5)
num2 <- 1:5
```

```
num1
```

```
## [1] 1 2 3 4 5
```

```
num2
```

```
## [1] 1 2 3 4 5
```

Los objetos también se pueden concatenar. Por ejemplo, podemos definir **num3** a partir de **num1** y **num2** del siguiente modo:

```
num3 <- c(num1,num2)
```

```
num3
```

```
## [1] 1 2 3 4 5 1 2 3 4 5
```

Para comprobar el tipo de datos de un vector puede usarse el comando **str**. Por ejemplo:

```
str (num1)
```

```
## num [1:5] 1 2 3 4 5
```

Como se observa en el resultado, al comprobar el tipo de datos en el ejemplo anterior, lo que hemos hecho es aplicar una **función** (en este caso **str**).

En el siguiente apartado introduciremos el concepto de **función**.

Te aconsejamos que dediques unos minutos a leer ese apartado ya que este es un aspecto crucial para entender cómo funciona R.

1.4.5. Funciones

Una **función** aplica un conjunto de operaciones sobre unos datos de entrada y devuelve un resultado.

La lógica es:

- Tengo el objeto **a** que contiene unos datos.
- Aplico la función **b** sobre los datos que hay en **a**.
- Guardo el resultado en **c**.

Esto se parece un poco a cuando hacemos una receta de cocina:

- Tengo una serie de ingredientes (por ejemplo, leche, azúcar, harina, huevos y chocolate) guardados en el objeto **ingredientes** mediante el comando **c()**.
- A continuación, realizo la función **cocinar** sobre el objeto **ingredientes** definido en el paso anterior.
- Y por último, guardo el resultado en **tarta**.

¿Cómo indicaríamos esta operación en R?

```
ingredientes <- c("leche", "azúcar", "harina", "huevos", "chocolate")
tarta <- cocinar(ingredientes)
```

Las funciones se indican con un nombre y un paréntesis. Dentro del paréntesis van los denominados **argumentos**, que son los datos que la función recibe como entrada. La función devuelve un resultado, que en este ejemplo se almacena en el objeto **tarta**.

Si ejecutamos el código anterior para cocinar la tarta, *R* dará un error porque la función `cocinar()` es solo un ejemplo y no existe en *R*. A continuación veremos ejemplos con funciones que sí existen y funcionan en *R*.

Algunas funciones tienen argumentos opcionales, que pueden especificarse o no. Por ejemplo, la función `mean()` calcula la media. Entre paréntesis () se incluye el o los argumento/s (si hubiera más de uno, se separa mediante comas). En el ejemplo inferior se solicita a *R* que calcule la media para la variable **datos**. Antes de solicitar la media, hay que crear el objeto **datos**. Como puede verse, **datos** incluye 50 números y el comando *NA* que se refiere a *los valores perdidos*:

```
datos <- c (1:50, NA)
media1 <- mean (datos)
media2 <- mean (datos, na.rm = TRUE)
```

```
media1
```

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

```
media2
```

```
## [1] 25.5
```

Si usamos únicamente como argumento el objeto **datos**, como aparece en el objeto **media1**, no obtendremos ningún resultado (por eso dice *NA*). Para obtener la media de **datos** necesitamos añadir el argumento `na.rm`, como se ha hecho en el objeto **media2**, que indica si los valores perdidos (denotados mediante *NA*) deben ignorarse (*FALSE*) o no (*TRUE*). En este ejemplo, la media de **datos** es 25,5.

1.5. El Script

Ya hemos hablado del *SCRIPT*. Es el fichero con extensión **.R* donde se escriben los comandos que nos permitirán ejecutar los análisis y/o solicitar gráficos. Los *Script* siempre se graban desde la *Ventana 1* de la Figura 1.11 y contienen *el programa* o listado de comandos que ejecutaremos para llevar a cabo nuestros análisis.

Es recomendable que este fichero no sea demasiado largo. Y también que anotemos algún comentario que nos permita recordar la operación que hayamos definido. Para añadir comentarios junto a los comandos se usa el símbolo *#*. A modo de ejemplo, escribiremos el ejemplo anterior con el comentario de lo realizado de la siguiente manera:

```
datos <- c(1:50, NA) # generación de 50 datos, NA son los valores perdidos
```

Una vez hayas elaborado el **Script**, debes guardarlo en una carpeta concreta. Es importante tener en cuenta que todos los archivos relacionados con nuestros análisis (e.g., datos, imágenes, etc.) deben encontrarse grabados en la misma carpeta donde esté nuestro **Script**. Solamente de esta forma funcionarán nuestros comandos. Esta operación es crucial y se denomina **Vincular carpeta**, y la explicaremos a continuación.

1.5.1. Vincular carpeta

Para trabajar en *R* necesitamos ser muy minuciosos y aprender a prestar atención a ciertos detalles. Uno de los aspectos más importantes que hay que aprender es **cómo iniciar nuestra sesión con R**. Con el programa *SPSS*, lo único que hacía falta era abrir nuestro archivo de datos y manejar las ventanas de los menús. Con el *Lenguaje R* hace falta algo más. Trabajaremos con el **SCRIPT**, que es el archivo donde se encuentran grabados los comandos, y tendremos que decirle a *R* en qué carpeta se encuentra tanto el Script como los otros ficheros que necesitaremos para que el Script entre en funcionamiento (por ejemplo, datos, imágenes, etc.). Si olvidamos *Vincular la carpeta al SCRIPT* antes de comenzar los análisis, y esto es algo que suele pasar cuando somos usuarios noveles, nos aparecerá un mensaje de error. Por tanto, es necesario concienciarse de que hay que hacer esto antes de empezar con el trabajo de análisis estadístico en *R*.

Para vincular la carpeta, tenemos que abrir nuestro Script en la *Ventana 1* de la Figura 1.11 y pulsar en el menú **Session -> Set Working directory (To Source File Location)**, tal y como se muestra en la Figura 1.13.

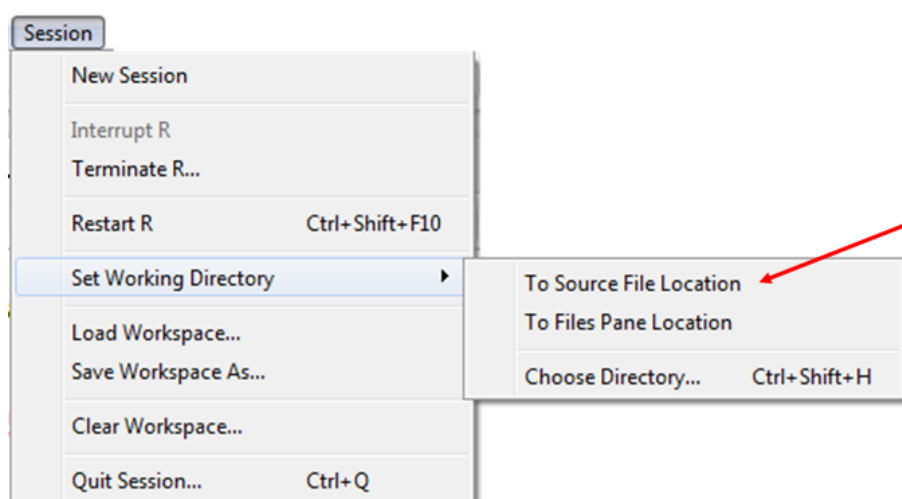



Figura 1.13: Vincular carpeta a Script

1.5.2. El concepto de Librería de R

Con la instalación de *R* tenemos muchas posibilidades de análisis estadístico que vienen por defecto. No obstante, *R* tiene la peculiaridad de que ofrece multitud de módulos opcionales, a los que se denomina **paquetes** (del inglés *packages*), aunque aquí nos referiremos a ellos como **librerías**.

Las **librerías** o **paquetes** son colecciones de funciones y datos y no siempre están todas activas, ya que esto enlentecería el funcionamiento del programa. Por tanto, habrá que llamarlas cada vez que necesitemos realizar ciertos análisis que no se encuentren disponibles por defecto con la instalación del programa *R base*. Hemos elaborado el anexo 7 para resumir las librerías que más usaremos aquí.

Lo que debes saber sobre las librerías es lo siguiente:

- El directorio de tu PC donde se almacenan los packages es **/library**.
- Para ver qué librerías tienes instaladas debes emplear la función `library()`. Al ejecutar este comando en la *Ventana 1* de la Figura 1.11, podrás ver qué **librerías** tienes instaladas y si dispones de todas las que necesitas. Esto también puede verse a través de la solapa **Packages** de la *Ventana 4* de la Figura 1.11.
- En caso de que no dispongas de la librería o **paquete** que necesites, deberás ir a la solapa *Packages* de la Ventana 4 de la Figura 1.11 y pulsar el icono  para buscar el librería que necesites e *Instalarla*.
- No basta con **instalar la librería**, también se necesita **Cargar la librería** para que esté disponible para su uso en *R*, cuando se necesite. Para cargar una **librería** que ya tengamos instalada se utiliza el comando `library("nombre_librería")`. Es importante que recuerdes cargar la **librería** que quieras usar en cada sesión de *R*. Pese a estar instalada, si no cargas la **librería**, no funcionará. Y esto ha de hacerse en cada sesión que iniciemos de *RStudio*.
- No es lo mismo “instalar” que “cargar” una **librería**. La instrucción `search()` nos dice qué librerías están instaladas y cargadas en el sistema, listas para usarse.
- Sólo instalaremos una vez cada **librería**. Para cargarla ejecutaremos el comando `library("nombre_librería")` en cada sesión en que queramos emplearla. Si no hacemos esta operación de *cargar la librería*, pese a tener instalada la **librería** ésta no funcionará y *R* nos dará error.
- Las **librerías** disponibles están en <http://cran.r-project.org/> (ver anexo 7)
- Normalmente son necesarias más **librerías** que las que vienen por defecto, así que es habitual instalar **librerías** nuevas.
- Si deseamos obtener información sobre una **librería** concreta emplearemos la función `help("nombre_de_la_librería")`.

1.6. Manejo de datos

En este apartado veremos cómo se definen los datos en *R*. Más adelante veremos cómo leer archivos externos (por ejemplo, de *Excel* o de *SPSS*) pero antes, es importante que nos familiaricemos con la forma en que *R* entiende nuestros datos.

1.6.1. Tipos de datos básicos

Los tipos básicos de datos en *R* son los siguientes:

Valores numéricos y enteros

Los números reales en *R* se denominan **numéricos**, y son el tipo de datos por defecto. Como ya hemos visto, para asignarles un valor se puede usar el comando `=` o bien `<-`

Por ejemplo, `x = 10.0`

los valores enteros se crean con la función `as.integer`, y para verificar si una variable contiene valores enteros podemos llamar a las funciones `class` o `is.integer`. Continuando con el ejemplo anterior:

```
x = as.integer(10)
class(x)
is.integer(x)
print(as.integer(10.5))
```

Valores lógicos

Las variables lógicas toman los valores `TRUE` y `FALSE`, que pueden abreviarse por `T` y `F`, respectivamente, y suelen crearse mediante la comparación de variables. Veamos un ejemplo:

```
x = 1 ; y = 2;
z = x > y
z
class(z)
```

Las variables lógicas se combinan mediante los *operadores* o *conectivas lógicas*:

- Conjunción (*y*): `&`
- Disyunción (*o*): `|`
- Negación (*no*): `!`

Por ejemplo:

```
x = 1; y = 2;
v = (x>1)&(y>1)
w = (x>1)|(y>1)
z = ((x>0)&!(y>0))|(!(x>0)&(y>0))
print(v)
print(w)
print(z)
print(v)
```

Cadenas de caracteres

Un objeto de tipo `character` es una variable de texto. Por ejemplo, el objeto `s`, que es un texto que tiene 44 caracteres (`nchar`):

```
s = "El hombre es una piedra arrojada en el vacío"
nchar(s)
```

```
## [1] 44
```

Las funciones `as.character` y `as.numeric` permiten convertir números en caracteres y viceversa. Por ejemplo:

```
x = 3.14
s = as.character(x)
y = as.numeric(s)
print(y+1)
print(s+1) # Esto va a dar un error
```

Por último, la función `sprintf` resulta de mucha utilidad para mostrar mensajes por pantalla. Es el equivalente a la función `printf` del *lenguaje C*, y combina variables de todos los tipos y las convierte en una cadena de texto en un formato legible.

Para ver un ejemplo puede correrse el siguiente comando y ver el resultado más abajo en `##[1]`:

```
n = "Pitágoras"
x = 3.14
y = as.integer(x)
sprintf("Según%s, pi vale%4.2f y su parte entera es%1d", n, x, y)
```

```
## [1] "Según Pitágoras, pi vale 3.14 y su parte entera es 3"
```


Factores

Los *factores* son variables categóricas que representan datos cualitativos. Resultan necesarias para realizar análisis estadísticos tales como el ANOVA, en los que los valores de la variable independiente se interpretan como etiquetas para formar grupos de observaciones y no tienen un sentido numérico (esto lo veremos en detalle en los capítulos 7 y 8). Para crear variables de este tipo se utiliza la función `factor`. Por ejemplo:

```
x = factor(1)
y = 1
is.factor(x)
is.factor(y)
x + y
```

También es posible convertir variables numéricas en factores del siguiente modo:

```
y = 1
z = factor(y)
is.factor(z)
```

1.6.2. Estructuras de datos

Las *estructuras de datos* se utilizan para representar las muestras del estudio estadístico u otros conjuntos de datos. Las **estructuras de datos** más comunes se definen en función del número de dimensiones en que se organizan los datos y del tipo de datos que admiten, y son las siguientes:

Dimensiones	Homogéneos	Heterogéneos
1	Vector atómico	Lista
2	Matriz	Marco de datos
N	Array	

1.6.2.1. Datos de una dimensión

La estructura de datos más sencilla incluye sólo una dimensión en el **vector**. Pueden ser de dos clases, **vectores atómicos** y **listas**.

Vectores atómicos

Todos los elementos de un vector atómico (**atomic vector**) son del mismo tipo, que debe ser alguno de los tipos básicos:

1 Familiarizarse con R

- Valores numéricos, números reales, también conocido como tipo *double*.
- Números enteros.
- Valores lógicos, TRUE o FALSE, que se puede abreviar por T o F.
- Caracteres y códigos de texto.

Además, el código NA indica un valor perdido.

Vamos a crear un vector de valores enteros:

```
int_dat <- 1:5
typeof(int_dat)
```

```
## [1] "integer"
```

```
print(int_dat)
```

```
## [1] 1 2 3 4 5
```

Como casi todo en R, la definición de vectores puede hacerse de diferentes formas. Las más comunes son mediante las funciones `c()`, `vector()`, `rep()` o `seq()`.

La función `c()`, como se ha visto anteriormente, toma una lista de argumentos de entrada y los reúne en un vector que devuelve como resultado.

Veamos un ejemplo que permite crear un vector que contiene valores numéricos y algún dato perdido:

```
dbl_dat <- c(23, 18, NA, 32, 27, 30, 25.5)
typeof(dbl_dat)
```

```
## [1] "double"
```

```
print(dbl_dat)
```

```
## [1] 23.0 18.0 NA 32.0 27.0 30.0 25.5
```

```
log_vec <- dbl_dat %% 2 == 0
typeof(log_vec)
```

```
## [1] "logical"
```

```
print(log_vec)

## [1] FALSE  TRUE    NA  TRUE FALSE  TRUE FALSE

sum(as.integer(log_vec), na.rm = TRUE)

## [1] 3
```

En el siguiente ejemplo utilizamos `c()` para crear un vector que contiene cadenas de caracteres:

```
str_dat <- c("esto", "es", "un", "curso", "de", "R")
typeof(str_dat)
print(str_dat)
paste(str_dat, collapse= " ")
```

Para acceder a los elementos de un vector se utilizan corchetes, `[]`. Por ejemplo:

```
numeros <- 0:10
numeros[1]
numeros[c(2,4,6)]
numeros[c(3:4)]
numeros[seq(1,10,2)]
numeros[numeros %% 2 == 0]
numeros[which(numeros > 5)]
```

Con la función `vector()` creamos un vector sin especificar cuáles son los elementos que contiene con el propósito de darles valor, posteriormente.

A `vector()` hay que pasarle el tipo de elementos que pueda admitir el vector y su longitud. Por ejemplo:

```
n <- vector("numeric", 3)
n[1] <- 3.14
n[2] <- 2.72
print(n)
```

```
## [1] 3.14 2.72 0.00
```

Con la función `rep()` creamos un vector con todos los elementos iguales, aunque estos pueden cambiarse posteriormente. Por ejemplo, para obtener un vector de unos de longitud cinco se utilizaría el siguiente código:

```
unos <- rep(1, 5)
print(unos)
```

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

La función `seq(x, y, c)` proporciona una secuencia de valores entre `x` e `y` con incrementos dados por `c`. Esto tiene gran utilidad para dibujar gráficos en un rango y con precisión especificados.

Por ejemplo, el siguiente código crea una secuencia de valores entre -3 y 3 con intervalos de cinco décimas y dibuja una curva normal como la que aparece en la Figura 1.14.

```
z <- seq(-3, 3, 0.5)
f <- exp(-0.5*z^2)
plot(z, f)
```

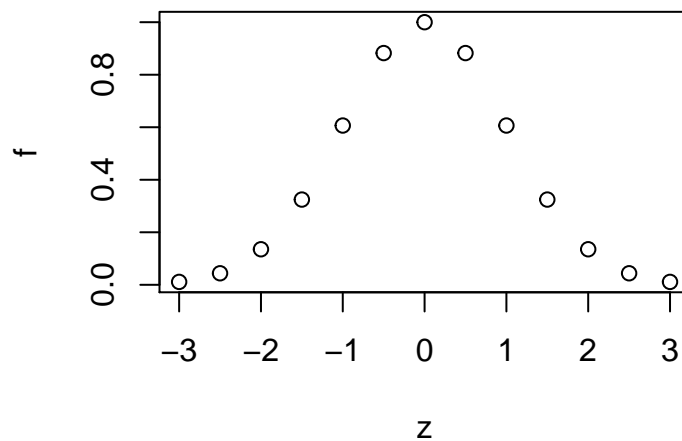


Figura 1.14: Curva normal con puntos

Podemos mejorar un poco el aspecto del gráfico cambiando algunas opciones. Por ejemplo, podemos modificar el tipo de gráfico (`type="l"` proporciona un gráfico de líneas) y los límites de los ejes, lo que permite obtener el resultado de la Figura 1.15:

```
z <- seq(-3, 3, 0.01)
f <- exp(-0.5*z^2)
plot(z, f, type="l", lwd=2, ylim=c(0,1))
```

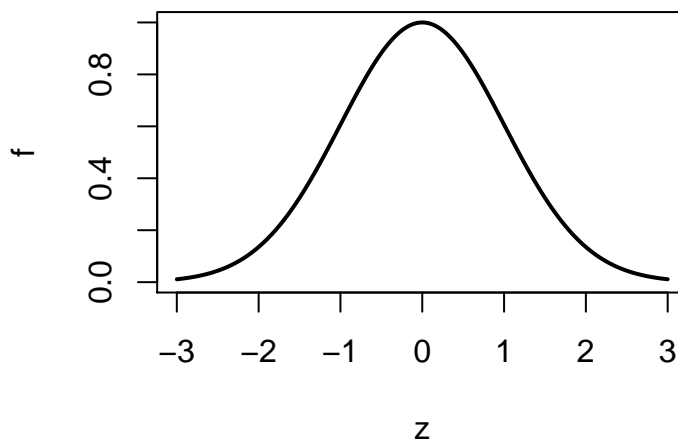


Figura 1.15: Curva normal con líneas

Listas

Una lista (`list`) es una colección ordenada de objetos conocidos como componentes. La lista se diferencia del `vector` en que los componentes pueden ser de diferentes tipos. Una lista puede estar compuesta por elementos de distintos tipos como valores numéricos, cadenas de caracteres, otra lista e incluso una matriz. Para acceder a los elementos de una lista se utilizan los corchetes `[]`, y para acceder a los elementos de una lista que está dentro de otra lista se utiliza el doble corchete `[[]]`.

En este ejemplo utilizamos la función `list` para crear una lista que contiene dos componentes, una cadena de texto y un vector numérico:

```
lista <- list("Valores de un dado", 1:6)
lista[1]
lista[2]
lista[[2]][3]
lista[[2]] > 3.5
```

Es posible unir varias listas en una mayor utilizando la función `c()`. Por ejemplo:

```
lista1 <- list("Valores de un dado", 1:6)
lista2 <- list("Valores de una moneda", c("cara", "cruz"))
lista <- c(lista1, lista2)
```

Atributos

Los atributos son etiquetas que indican las propiedades de la estructura de datos o de sus valores individuales. Para acceder a los atributos de un objeto se utilizan las funciones `attr()` y `attributes()`. La función `attr()` permite acceder a un atributo por su nombre, y `attributes()` indica cuales son los atributos de un objeto.

```
unos <- rep(1, 10)
attr(unos, "nombre") <- "Esto es un vector de unos"
attributes(unos)
attr(unos, "nombre")
```

Con la función `names()` asignamos nombres a cada uno de los elementos de un vector, lo que nos permite acceder al mismo por su nombre.

```
v <- c(2,4,6)
names(v) <- c("Primero", "Segundo", "Tercero")
v["Segundo"]
v[c("Primero", "Tercero")]
```

Uno de los atributos más importantes de un objeto es el que permite definir variables de tipo *factor*. Como hemos visto, los factores son variables categóricas que pueden ser utilizadas como variables independientes en un análisis de varianza o en otros análisis. De este modo es posible decirle a *R* que los valores de una variable no deben ser tratados como números sino como etiquetas.

Un factor se define aplicando la función `factor()` sobre una variable que contenga números enteros. Las funciones `class()` y `levels()` indican si un variable es o no un factor y cuáles son sus niveles.

```
vi <- 1:5
factor_vi <- factor(vi)
class(vi)
class(factor_vi)
levels(vi)
levels(factor_vi)
```

1.6.2.2. Datos de dos o más dimensiones

Matrices

Una matriz es una ordenación de elementos en dos o más dimensiones. Todos los elementos de una matriz deben ser del mismo tipo. Las matrices en *R* se utilizan para

representar muestras de datos, matrices de varianza-covarianza, etc. y para realizar operaciones matemáticas de álgebra matricial. Las matrices se definen mediante la función `matrix`, y las operaciones matemáticas que pueden aplicarse sobre ellas pueden consultarse en <http://www.statmethods.net/advstats/matrix.html>. Entre otras:

```
a <- matrix(1:6, nrow=2, ncol=3)
dim(a)
nrow(a)
ncol(a)
rownames(a) <- c("fila 1", "fila 2")
colnames(a) <- c("columna 1", "columna 2", "columna 3")
b <- matrix(7:12, nrow=3, ncol=2, byrow = TRUE)
c <- a%*%b
d <- solve(c)
print(a)
print(a[2,3])
print(b)
print(c)
print(d)
print(c%*%d)
```

Se puede acceder a los vectores que constituyen las filas o columnas de una matriz. Por ejemplo:

```
A <- matrix(seq(1,10), nrow=2, byrow=T)
A[1,]
A[,1]
```

También es posible crear una matriz combinando varios vectores o matrices mediante `cbind` (combinar columnas) o `rbind` (combinar filas).

```
A <- matrix(seq(1,4), nrow=2, byrow=T)
B <- matrix(seq(5,8), nrow=2, byrow=T)
cbind(A,B)
rbind(A,B)
```

Arrays

Los *arrays* son una generalización de las matrices para objetos de más de dos dimensiones. Por ejemplo, el siguiente código permite definir y acceder a un `array` de tres dimensiones.

```
a <- array(1:12, c(2,3,2))
dim(a)
print(a)
print(a[1,,])
print(a[2,,])
a[1,2,2] <- a[1,2,2]^2
```

Marcos de datos

Los marcos de datos (**data frame**) son un tipo especial de matriz que se caracteriza porque puede combinar elementos de diferentes tipos. Por ejemplo, puede contener elementos numéricos y cadenas de caracteres al mismo tiempo. Las columnas de los marcos de datos son variables y todas ellas deben tener la misma longitud, aunque el tipo de datos de una variable y otra puede ser distinto. Por este motivo se utilizan mucho para representar muestras de datos en las que las variables tienen distinto tipo. Más adelante las usaremos para elaborar una “Tabla de Frecuencias” (ver apartado 2.1.1).

Los marcos de datos se crean con el comando `data.frame()`, en el que se indican el nombre y los valores de sus variables. Para acceder a las variables que forman parte de un marco de datos podemos utilizar tanto el símbolo `$` como el doble corchete `[[]]`. Además, podemos utilizar el comando `str()` para obtener información sobre la estructura de un marco de datos. Por ejemplo, el siguiente código crea un marco de datos con dos variables, una que contiene los nombres de los sujetos y otra su edad.

```
df <- data.frame(nombres <- c("Claudia", "Gador", "Alba"),
                 edades <- c(8, 9, 12))
str(df)
df
df$nombres
df$edades
df[[1]]; df[[2]]
```

En ocasiones, la utilización del signo `$` puede ser engorrosa para referirse a las variables incluidas en un marco de datos. Para evitar utilizar `$` podemos llamar a la función `attach`, que hace que todas las variables incluidas en un marco de datos sean visibles. Continuando con el ejemplo anterior:

```
df <- data.frame(nombres <- c("Claudia", "Gador", "Alba"),
                 edades <- c(8, 9, 12))
df
attach(df)
nombres
edades
```


También es posible convertir las matrices en marcos de datos utilizando el comando `as.data.frame`, lo que tiene la ventaja de que permite acceder a las variables del marco de datos llamándolas por su nombre.

```
a <- matrix(1:6, nrow=2, ncol=3)
df <- as.data.frame(a)
print(a)
print(df)
df$V1
df$V2
df$V3
```

La función `as.data.frame()` asigna por defecto los nombres `V1`, `V2`, `V3` a nuestras variables. Estos nombres podemos cambiarlos fácilmente con la función `names()`.

```
names(df) <- c("col1", "col2", "col3")
df$col2
df[, "col2"]
```

1.7. Leer datos de archivos externos

Los marcos de datos pueden leerse y guardarse en archivos de datos en formato texto, SPSS y Excel utilizando las funciones que veremos a continuación.

1.7.1. Datos de texto

Para leer datos en formato de texto se utiliza la función `read.table()`. Por ejemplo, podemos leer el archivo de datos `ejemplo.dat` de la Figura 1.16 en *R* utilizando los siguientes comandos:

```
ejemplo <- read.table("ejemplo.dat", header=TRUE)
names(ejemplo)
```

Una matriz que hayamos creado en *R* podemos guardarla en un archivo de texto utilizando la función `write.table`. Por ejemplo, en las siguientes líneas de código definimos una matriz con tres filas y dos columnas y la guardamos en el archivo `matriz.dat`:

```
matriz <- matrix(c(1,2,3,4,5,6), nrow=3)
write.table(matriz, "matriz.dat", row.names=F, col.names=F)
```


Sujeto	Sexo	Inteligencia	Nivel cultural	Estrés
1	0	101	2	4
2	1	83	1	5
3	1	95	2	6
4	0	89	1	4
5	0	107	2	7

Figura 1.16: Datos para ejemplo.dat

1.7.2. Datos en SPSS

Para leer archivos de datos SPSS como el del ejemplo de la Figura 1.17 (fichero *practica.sav*) en R y convertirlos en un objeto puede usarse la función `read.spss()`, que está disponible en la librería **foreign**. La sintaxis que hay que emplear es la siguiente:¹.

```
library("foreign")
datos <- read.spss('practica.sav')
```



	Clave	genero	edad	peso	estatura	prov	idprov	rama
1	1	Mujer	34	46	1.63	MADRID	46	Ciencias Sociales y Juridicas
2	2	Mujer	34	51	1.72	BURGOS	22	Otros/Varios
3	3	Varon	25	45	1.53	VALLADOLID	28	Ciencias Sociales y Juridicas
4	4	Varon	26	64	1.75	PALENCIA	24	Ciencias Sociales y Juridicas
5	5	Varon	27	58	1.68	MADRID	46	Enseñanzas técnicas
6	6	Mujer	29	45	1.64	BURGOS	22	Humanidades
7	7	Varon	28	60	1.70	MADRID	46	Enseñanzas técnicas
8	8	Mujer	27	45	1.64	BURGOS	22	Ciencias Sociales y Juridicas
9	9	Varon	33	60	1.70	VALLADOLID	28	Humanidades
10	10	Varon	28	54	1.63	BURGOS	22	Ciencias Sociales y Juridicas
11	11	Varon	28	54	1.63	LEON	23	Ciencias Experimentales y de la Salud
12	12	Varon	32	54	1.63	BURGOS	22	Enseñanzas técnicas
13	13	Varon	35	61	1.71	PALENCIA	24	Ciencias Sociales y Juridicas
14	14	Varon	28	62	1.72	ZAMORA	29	Ciencias Experimentales y de la Salud
15	15	Varon	30	56	1.65	ASTURIAS	12	Humanidades
16	16	Mujer	35	45	1.64	ALAVA	50	Ciencias Sociales y Juridicas

Figura 1.17: Datos para practica.sav

Para guardar un marco de datos en un archivo que *SPSS* pueda leer utilizaremos la función `write.foreign()`, que también está en la librería **foreign**. Esta función no guarda un archivo *.sav* en formato SPSS, lo que hace es guardar un archivo de texto y un archivo de comandos para leer desde *SPSS* esos datos en formato texto.

¹En ocasiones esta sintaxis puede fallar. Alternativamente, puedes cargar la librería **haven** (si no la tienes, debes instalarla) y usar el código `practica <- read_sav("H:\carpeta\practica.sav")` y `View(practica)`. También es posible abrir un archivo SPSS desde la *Ventana 3* de la Figura 1.11 (Memoria de Trabajo) pulsando **Import Dataset** y seleccionando **From SPSS**. De este modo, R cargará el archivo como objeto en tu memoria y le dará el nombre *practica*.

```
matriz <- as.data.frame(matrix(c(1,2,3,4,5,6), nrow=3))
write.foreign(matriz, "matriz_spss.dat", "matriz_spss.sps",
              package=c("SPSS"))
```

1.7.3. Datos CSV

El formato *Comma Separated Values* (CSV) es un tipo de archivo de texto en el que las variables están separadas por comas y los decimales por punto. En el formato CSV2 las variables se separan por punto y coma y los decimales por coma. Para grabar un marco de datos en estos formatos utilizaremos la siguiente sintaxis:

```
a <- as.data.frame(matrix(seq(0.5,25,0.5), nrow=10, ncol=5))
write.csv(a, "matriz1.csv", row.names=FALSE)
write.csv2(a, "matriz2.csv", row.names=FALSE)
```

Los archivos CSV podemos leerlos con las funciones `read.csv()` o `read.csv2()`.

```
a1 <- read.csv("matriz1.csv")
a2 <- read.csv2("matriz2.csv")
```

1.7.4. Datos en Excel

Es posible guardar un marco de datos en un archivo *Excel* utilizando la función `write.xlsx()`, disponible en la librería `xlsx`.

```
library("xlsx")
a <- as.data.frame(matrix(1:6, nrow=2, ncol=3))
write.xlsx(a, "matriz.xls", col.names=FALSE, row.names=FALSE)
```

R es capaz de leer un archivo *Excel* de la versión 2003 o anteriores. Para ello utilizaremos la función `read.xls()`. Como los archivos de *Excel* pueden tener varias hojas de datos, debemos indicar a la función cual es el número de hoja (`sheetIndex`) que queremos leer. Por ejemplo:

```
b = read.xlsx("matriz.xls", sheetIndex=1, header=F)
```

1.8. Los gráficos en R

El programa *R*, como ya se ha señalado, ofrece unas posibilidades de hacer gráficos muy superiores a las que ofrecen otros software estadísticos. Ya vimos ejemplos de gráficos realizados en *R* en las Figuras 1.1 y 1.2.

Iremos introduciendo los códigos para elaborar gráficos a medida que vayamos avanzando en contenidos para llevar a cabo los diferentes análisis estadísticos vistos en las asignaturas *Análisis de Datos I y II*.

A modo de ejemplo, a continuación veremos dos tipos de representaciones gráficas. Una muy sencilla, referida a análisis univariado, y otra más compleja referida a funciones de densidad de probabilidad.

Diagrama de barras

Para realizar un *diagrama de barras*, en primer lugar hay que definir unos datos para *X*. Para ello, tal y como hemos visto en apartados anteriores, puede utilizarse el comando `c()`. A continuación, se utiliza el comando `barplot(X)` para obtener la gráfica de la Figura 1.18:

```
X <- c(1,2,3,4,5,6)
barplot(X)
```

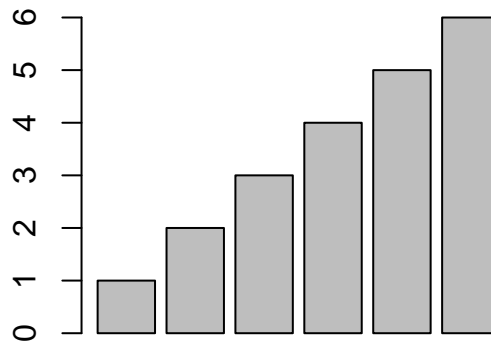


Figura 1.18: Diagrama de barras

Representación gráfica de funciones

Para representar gráficamente una función, en primer lugar tenemos que definir el conjunto de valores que forman el eje de ordenadas y lo guardamos en un vector. A continuación obtenemos el valor de la función para cada uno de dichos valores y lo guardamos en un segundo vector. Después le pasamos ambos valores a la función `plot`.

Por ejemplo, supongamos que vamos a representar la siguiente función:

$$f(x) = 6x(1 - x),$$

definida en el intervalo $(0, 1)$. El siguiente código R muestra cómo hacer la representación gráfica. La primera línea de código utiliza el comando `seq` para definir el vector de valores $x = 0, 0,01, 0,02, \dots, 1$. La segunda línea calcula la función $f(x) = 6x(1 - x)$ y guarda sus valores en el vector `f`. La tercera línea contiene el comando `plot` para representar el gráfico utilizando los pares de puntos (x_i, f_i) . El resultado aparece en la Figura 1.19.

```
x <- seq(0, 1, by=0.01)
f <- 6*x*(1-x)
plot(x, f, type="l", lwd=2, ylab="", col="red")
```

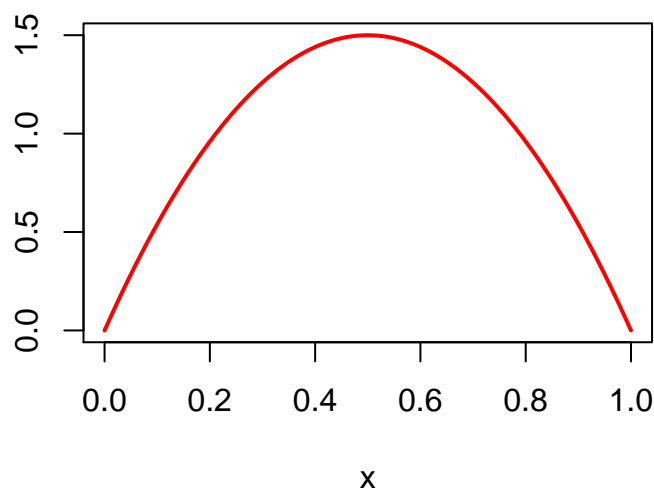


Figura 1.19: Representación gráfica de $f(x) = 6x(1-x)$

A continuación veremos cómo representar varias funciones en los mismos ejes de coordenadas. Esto ocurre por ejemplo cuando queremos representar dos funciones de densidad distintas o una función de densidad y su correspondiente función de distribución.

Vamos a elaborar una gráfica con la misma función $f(x)$ vista en el ejemplo anterior y además la función

$$g(x) = 18x(1 - x)^4$$

Para ello, en primer lugar definimos los valores de x y calculamos los valores de f y g . A continuación hay que representar $f(x)$ mediante el comando `plot`, y se utiliza el comando `lines` para añadir líneas a un gráfico ya existente. Por último, utilizando el comando `legend` se añade una leyenda a la figura para especificar cuál es la línea correspondiente a cada función. En la Figura 1.20 puede verse el resultado.

```
x <- seq(0, 1, by=0.01)
f <- 6*x*(1-x)
g <- 18*x*(1-x)^4
plot(x, f, type="l", lwd=2, ylab="", xlab=substitute(italic("x")),
     ylim=c(0,1.6), col="red")
lines(x, g, lwd=2, col="blue")
legend(0.75, 1.6, c("f(x)", "g(x)"), lty=c(1,1), lwd=2, col=c("red", "blue"))
```

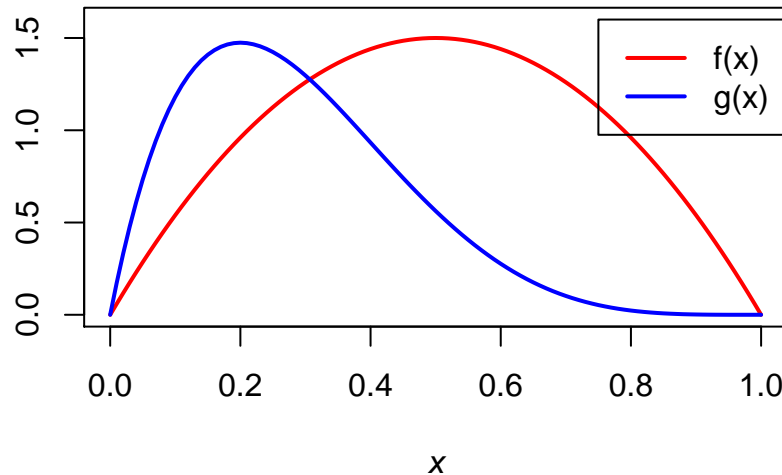


Figura 1.20: Representación de dos funciones en una misma gráfica

1.9. Ejercicios propuestos

Disponemos de los siguientes datos de la variable X medida en dos grupos:

Grupo 1 (X_1): 1, 2, 3, 4, 5, 6

Grupo 2 (X_2): 3, 5, 1

Responde a las siguientes cuestiones (para cada una de ellas indica la sintaxis R que has empleado y el resultado que has obtenido):

1. Define los grupos de puntuaciones X_1 y X_2 en R (indica los comandos de R que has utilizado para ello)
2. Indica cuál es el tamaño muestral en cada grupo (indica los comandos de R que has utilizado para responder a esta cuestión)
3. Calcula la media y varianza para X_1 y X_2 e indica los comandos de R que has empleado para ello. Anota también el resultado que te da R para cada uno de estos estadísticos.
4. Piensa ahora cómo calcularías la media y varianza de X_1 sin llamar a las funciones `mean()` y `var()`. Lo que te pedimos aquí es que escribas tú los comandos. Ten en cuenta que `var()` te da la *cuasivarianza*. Por tanto, si aplicas la fórmula [3.5] de la página 95 del libro de Botella, Suero y Ximénez (2012), tus resultados no van a coincidir con los que obtienes con el comando `var()` de R . [Como este ejemplo tiene pocos números, siempre puedes calcular estos estadísticos a mano, o con ayuda de la calculadora, y así comprobar si tus resultados son correctos].
5. Puedes repetir el ejercicio 4 con la variable X_2
6. Calcula ahora la **Media Total** para X . Piensa cómo hacerlo. Te doy una pista. Vas a tener que trabajar con la fórmula que vimos en las propiedades de la media (pág. 99 del libro de Botella et al., 2012). Verás que, con lo que sabes de R , tras esta primera sesión, te resultará muy fácil escribir esto. Hazlo por tu cuenta y comprueba que el resultado te da 3,33.
7. Si te animas a seguir probando con R , puedes escribir también la fórmula para la desviación típica, asimetría y curtosis de X_1 y X_2 .

2 Análisis descriptivos

En este capítulo veremos cómo llevar a cabo los análisis descriptivos con una y dos variables que hemos estudiado en la asignatura *Análisis de Datos I*. Para explicar la redacción de los códigos *R* nos basaremos en los ejemplos del libro de Ximénez y Revuelta (2011), donde se explica el manejo del programa *SPSS*. Vamos por tanto a llevar a cabo esos mismos análisis pero en lugar de con el *SPSS* con el *Lenguaje R*.

Para ilustrar el manejo de *R* usaremos los datos descritos en el apartado 1.7.2 del capítulo anterior referidos al fichero *practica.sav*. Son datos recogidos en una muestra de 200 participantes en variables relacionadas con los rasgos de personalidad del Big five, aunque también hay otras variables de tipo demográfico como el sexo, la edad, la provincia, etc. (para más detalles véase el anexo 1). El fichero con estos datos se puede descargar desde: <https://www.psicologiauam.es/carmen.ximenez/Practicas1.html>

2.1. Iniciar la sesión con RStudio y preparar Script

Antes de comenzar con la descripción de los análisis descriptivos es importante que recordemos cómo funciona *RStudio*. Cada vez que inicies la sesión has de tener en cuenta lo siguiente:

- Debes tener una carpeta de trabajo (en tu USB o disco duro del ordenador) donde se encuentren todos los archivos que necesites para llevar a cabo los análisis. Normalmente suelen ser el archivo de datos (por ejemplo, el fichero *practica.sav*) y el *Script* donde tengas escritos tus comandos, que si fuese nuevo necesariamente habrá de guardarse en esa carpeta de trabajo.
- Para que *RStudio* entienda que trabajaremos con esa carpeta, lo primero que ha de hacerse es *Vincular la carpeta* al Script al iniciar sesión. Esto puede hacerse tal y como se muestra en la figura 1.13 vista en el capítulo anterior.
- Como ya sabemos, *en R se trabaja definiendo objetos*. Esto es, **el archivo de datos se concibe también como un objeto**. Por tanto, cada vez que inicies sesión, además de vincular tu carpeta al Script, tendrás que llamar a tu **archivo de datos**. Es probable que para ello necesites alguna librería. En nuestro caso, como trabajaremos con un archivo SPSS, tendremos que definir un objeto que permita leer los datos en el fichero *practica.sav*. Adicionalmente, tendremos que activar la librería *foreign*, que nos permitirá llevar a cabo esta

operación. Para ello, como ya vimos en el apartado 1.7.2 del capítulo anterior, tendrás que escribir los siguientes comandos:

```
library("foreign")
practica <- read.spss("practica.sav", to.data.frame=TRUE)
```

```
## re-encoding from UTF-8
```

Con esto lo que hemos hecho es definir el objeto `practica`, que contiene los datos de las 17 variables incluidas en el fichero `practica.sav` (ver anexo 1). Hecho esto, ya podemos llamar al objeto `practica` para solicitar a *R* que nos proporcione análisis descriptivos o que elabore gráficos. Veremos cómo hacerlo en los siguientes apartados.

2.2. Análisis descriptivos univariantes

En este apartado veremos cómo usar *R* para llevar a cabo análisis descriptivos con una sola variable. Seguiremos el mismo esquema que en el manual de la asignatura *Análisis de Datos I* de Botella, Suero y Ximénez (2012) y el cuaderno de prácticas de *Análisis de datos con SPSS* de Ximénez y Revuelta (2011).

Para una primera aproximación al resumen descriptivo de los datos puede usarse el comando `summary()`. Dentro del paréntesis iría el objeto que queramos analizar. Siguiendo con nuestro ejemplo, podemos empezar resumiendo las 17 variables que hay en `practica`. Se haría del siguiente modo:

```
summary(practica)
```

```
##      Clave      genero      edad      peso      estatura
## Min.   : 1.00  Mujer: 81  Min.   :22.00  Min.   :39.0  Min.   :1.520
## 1st Qu.: 50.75  Varon:119 1st Qu.:27.00  1st Qu.:51.0  1st Qu.:1.620
## Median :100.50          Median :29.00  Median :58.0  Median :1.660
## Mean   :100.50          Mean   :29.35  Mean   :58.3  Mean   :1.676
## 3rd Qu.:150.25          3rd Qu.:31.00  3rd Qu.:64.0  3rd Qu.:1.720
## Max.   :200.00          Max.   :46.00  Max.   :92.0  Max.   :1.930
##      prov      idprov      rama
## Length:200      Min.   :11.00  Ciencias Experimentales y de la Salud:35
## Class :character 1st Qu.:12.00  Ciencias Sociales y Juridicas      :79
## Mode  :character Median :28.00  Enseñanzas tecnicas                :37
##          Mean   :29.25  Humanidades                        :32
##          3rd Qu.:46.00  Otros/Varios                      :17
##          Max.   :52.00
```

2.2 Análisis descriptivos univariantes

```
##      licen          inteli      compren      orient
## Length:200      Min.   : 9.00    Min.   : 8.00    Min.   : 5.972
## Class :character 1st Qu.:15.00    1st Qu.:23.00    1st Qu.: 14.318
## Mode  :character Median :18.00    Median :26.00    Median : 26.435
##                Mean  :18.55    Mean  :25.77    Mean   : 41.216
##                3rd Qu.:22.00    3rd Qu.:29.00    3rd Qu.: 66.019
##                Max.   :29.00    Max.   :32.00    Max.   :142.287
##      extra      respon      emocio      sincer
## Min.   :29.00    Min.   :33.00    Min.   :36.00    Min.   :12.00
## 1st Qu.:39.00    1st Qu.:43.00    1st Qu.:45.00    1st Qu.:20.00
## Median :41.00    Median :45.00    Median :48.00    Median :24.00
## Mean   :41.36    Mean   :46.12    Mean   :48.62    Mean   :23.19
## 3rd Qu.:44.00    3rd Qu.:49.00    3rd Qu.:52.00    3rd Qu.:26.00
## Max.   :53.00    Max.   :59.00    Max.   :60.00    Max.   :43.00
##      fumar
## No fumador:138
## Fumador   : 62
##
##
##
##
```

Como puede verse en el resultado, se han analizado las 17 variables que hay en `practica`. Para cada una de las variables cuantitativas se ofrece: el valor mínimo y máximo, el cuartil 1 y 3, la mediana y la media. Para las variables cualitativas solamente se incluye el recuento de frecuencias para cada categoría.

Si hubiéramos querido analizar una sola de las variables, por ejemplo la `edad` o la `rama de estudios`, tendríamos que especificarlo en el argumento. Una de la formas de hacerlo es mediante el comando `$`. Veamos un ejemplo para `edad` y `rama`:

```
summary(practica$edad)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 22.00   27.00   29.00   29.35   31.00   46.00
```

```
summary(practica$rama)
```

```
## Ciencias Experimentales y de la Salud      Ciencias Sociales y Juridicas
##                                     35                                     79
##                               Enseñanzas tecnicas                               Humanidades
##                                     37                                     32
##                               Otros/Varios
##                                     17
```

2.2.1. Tablas de frecuencias

Comenzaremos viendo cómo elaborar una *Tabla de frecuencias*. Continuando con el ejemplo de la variable `edad`, utilizaremos la siguiente sintaxis para obtener:

- Las frecuencias absolutas (`t1`)
- Las frecuencias relativas (`t2`)
- Las frecuencias absolutas acumuladas (`t3`)
- Las frecuencias relativas acumuladas (`t4`),

y colocar todo en una tabla (mediante la función `data.frame`):

```
(t1 = table(practica$edad))    # frecuencias absolutas, Edad
```

```
##
## 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 40 41 42 46
##  1  2  9 15 20 16 30 27 22 14 10  8  8  5  1  4  2  2  1  2  1
```

```
(t2 = prop.table(t1))        # frecuencias relativas, Edad
```

```
##
##    22    23    24    25    26    27    28    29    30    31    32    33    34
## 0.005 0.010 0.045 0.075 0.100 0.080 0.150 0.135 0.110 0.070 0.050 0.040 0.040
##    35    36    37    38    40    41    42    46
## 0.025 0.005 0.020 0.010 0.010 0.005 0.010 0.005
```

```
(t3 = cumsum(t1))           # frecuencias absolutas acumuladas, Edad
```

```
##  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  40  41  42  46
##   1   3  12  27  47  63  93 120 142 156 166 174 182 187 188 192 194 196 197 199
##  46
## 200
```

```
(t4 = cumsum(t2))           # frecuencias acumuladas relativas, Edad
```

```
##    22    23    24    25    26    27    28    29    30    31    32    33    34
## 0.005 0.015 0.060 0.135 0.235 0.315 0.465 0.600 0.710 0.780 0.830 0.870 0.910
##    35    36    37    38    40    41    42    46
## 0.935 0.940 0.960 0.970 0.980 0.985 0.995 1.000
```

```
data.frame(t1, t3, t2, t4)      # tabla de frecuencias completa
```

##	Var1	Freq	t3	Var1.1	Freq.1	t4
## 22	22	1	1	22	0.005	0.005
## 23	23	2	3	23	0.010	0.015
## 24	24	9	12	24	0.045	0.060
## 25	25	15	27	25	0.075	0.135
## 26	26	20	47	26	0.100	0.235
## 27	27	16	63	27	0.080	0.315
## 28	28	30	93	28	0.150	0.465
## 29	29	27	120	29	0.135	0.600
## 30	30	22	142	30	0.110	0.710
## 31	31	14	156	31	0.070	0.780
## 32	32	10	166	32	0.050	0.830
## 33	33	8	174	33	0.040	0.870
## 34	34	8	182	34	0.040	0.910
## 35	35	5	187	35	0.025	0.935
## 36	36	1	188	36	0.005	0.940
## 37	37	4	192	37	0.020	0.960
## 38	38	2	194	38	0.010	0.970
## 40	40	2	196	40	0.010	0.980
## 41	41	1	197	41	0.005	0.985
## 42	42	2	199	42	0.010	0.995
## 46	46	1	200	46	0.005	1.000

Como puede verse, el resultado, estéticamente hablando, es poco vistoso, pero incluye toda la información necesaria para elaborar una tabla de frecuencias: Los valores de la variable X_i (**Var1**), las frecuencias absolutas n_i (**Freq**), las frecuencias absolutas acumuladas n_a (**t3**), las frecuencias relativas p_i (**Freq.1**) y las frecuencias relativas acumuladas p_a (**t4**), de donde también pueden deducirse los centiles C_k .

2.2.2. Gráficos

En cuanto a las representaciones gráficas, en principio veremos cómo elaborar las gráficas apropiadas en el caso univariante para variables medidas en escalas nominales, ordinales y cuantitativas. No obstante, es posible elaborar otros muchos gráficos, como iremos viendo en posteriores capítulos del libro.

Por ejemplo, continuando con el ejemplo de la variable **edad**, como es cuantitativa, puede elaborarse el histograma o el diagrama de barras de las Figuras 2.1 y 2.2.

```
hist(practica$edad, main=NULL)
```

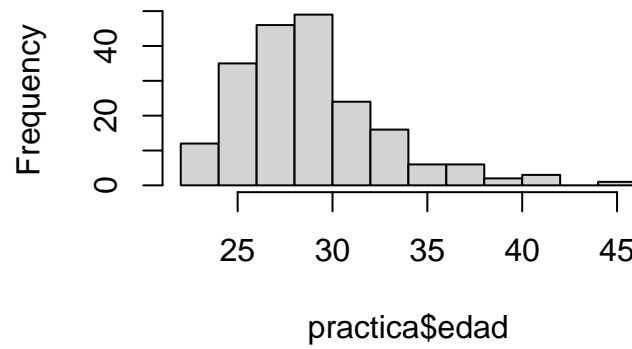


Figura 2.1: Histograma para Edad

```
barplot(table(practica$edad))
```

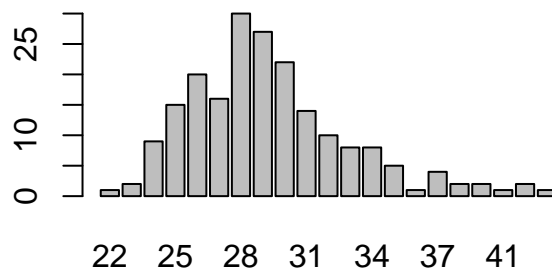


Figura 2.2: Diagrama de barras para Edad

En cuanto a la variable **rama de estudios**, que es nominal, elegimos un pictograma:

```
pie(table(practica$rama))
```

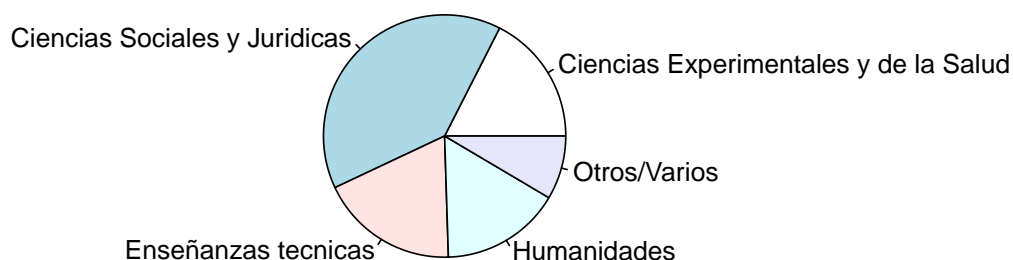


Figura 2.3: Pictograma para Rama de estudios

2.2.3. Estadísticos

En este apartado veremos cómo obtener los estadísticos para una sola variable. Siguiendo el mismo orden que en la asignatura *Análisis de Datos I*, comenzaremos con las medidas de posición, y continuaremos con las de tendencia central, variación y el análisis de la forma de la distribución (asimetría y curtosis).

Medidas de posición: los Centiles

Los *centiles* son puntuaciones que dejan por debajo de sí a un porcentaje $k\%$ de sujetos y por encima al $(100 - k\%)$ restante. Continuando con el ejemplo del fichero **practica**, vamos a solicitar ciertos centiles para la variable **edad**: C_{10} , C_{25} , C_{50} , C_{75} y C_{90} . Para ello puede emplearse el comando **quantile** y la siguiente sintaxis:

```
quantile(practica$edad, c(.10, .25, .50, .75, .90))
```

```
## 10% 25% 50% 75% 90%
## 25 27 29 31 34
```

Como puede verse, el $k\%$ aparece en la parte superior y la puntuación correspondiente en la parte inferior. Esto es, $C_{10} = 25$, $C_{25} = 27$, $C_{50} = 29$, $C_{75} = 31$ y $C_{90} = 34$.

Medidas de Tendencia central

Las medidas de tendencia central (MTC) son índices que representan la magnitud general de un conjunto de N valores. Las más empleadas son la *media*, la *mediana* y la *moda*.

Para calcular estos indicadores se utilizan los comandos `mean`(media), `median` (mediana) y `mfv`(moda). El comando `mfv` funciona instalando y cargando la librería "`modeest`". Continuando con el ejemplo del fichero `practica`, calcularemos cada una de las MTC referida a las variables `edad`, `peso` y `rama`. La sintaxis a emplear es la siguiente:

```
mean(practica$edad)
```

```
## [1] 29.35
```

```
median(practica$peso)
```

```
## [1] 58
```

```
library("modeest")  
mfv(practica$rama)
```

```
## [1] Ciencias Sociales y Juridicas  
## 5 Levels: Ciencias Experimentales y de la Salud ...
```

Los resultados ofrecen la *media* y la *mediana* para `edad` y `peso`, respectivamente. En el caso de la *moda*, para su cálculo hemos tenido que cargar la librería "`modeest`" (en caso de no tenerla instalada en tu ordenador, debes instalarla desde la solapa Packages de la Figura 1.11 tal y como quedó explicado en el apartado 1.5.2 del capítulo anterior. Ten en cuenta que esta librería sólo está disponible para versiones de *R* de 3.5.2 o superiores). En el ejemplo, la *moda* para `rama` es 2 (categoría "Ciencias sociales y Jurídicas") porque hay el máximo de observaciones; esto es, 79 observaciones.

Recordemos que la elección de la MTC depende de la escala de medida empleada para la variable. La *media* sólo puede calcularse para variables cuantitativas, la *mediana* para variables medidas en escala ordinal o cuantitativa, y la *moda* suele usarse para variables medidas en escala nominal.

Variabilidad

Como sabemos, las muestras de datos no pueden describirse únicamente mediante la tendencia central. Dos conjuntos de puntuaciones pueden tener la misma media y ser, sin embargo, muy distintos. Las MTC deben complementarse con las medidas de variación (MV), que hacen referencia al grado en que los datos se parecen entre sí.

Las MV son índices que representan la variabilidad o dispersión de un conjunto de N valores. Las más empleadas son la *varianza* y la *desviación típica*.

Para calcular estos indicadores se utilizan los comandos `var`(varianza) y `sd`(desviación típica). No es preciso cargar ninguna librería para que funcionen estos comandos.

Continuando con el ejemplo del fichero `practica`, calcularemos la *varianza* y la *desviación típica* para la variable `edad`. La sintaxis a emplear es la siguiente:

```
var(practica$edad)
```

```
## [1] 15.03266
```

```
sd(practica$edad)
```

```
## [1] 3.877198
```

R ofrece los valores de los estadísticos *varianza* y *desviación típica* para `edad` pero, al igual que ocurría con SPSS, realmente son la cuasivarianza y su raíz cuadrada.

Coefficiente de Variación de Pearson

A continuación vamos a ver cómo calcular con *R* otro estadístico de variabilidad que vimos en clase y resulta de mucha utilidad cuando estamos comparando la variabilidad de dos o más grupos en una misma variable X_i y las medias en los grupos son muy diferentes. El *Coefficiente de Variación* de Pearson (*CV*) es el cociente entre la desviación típica y la media multiplicado por 100. A mayor valor en *CV*, mayor variación y viceversa.

El estadístico *CV* no suele estar disponible en los menús desplegables de los softwares más conocidos (por ejemplo, el SPSS), pero es muy sencillo escribir el código en *R* para obtener el valor de *CV*. Continuando con el ejemplo anterior, el *CV* para la variable `edad` puede obtenerse mediante la siguiente sintaxis:

```
(CV_edad = (sd(practica$edad)/mean(practica$edad)) * 100)
```

```
## [1] 13.21021
```

Representación gráfica de la variabilidad

En este apartado veremos también cómo elaborar gráficos que informan de forma eficaz sobre la variabilidad de una variable X . Uno de los más conocidos es el *Diagrama de cajas y bigotes* (denominado **boxplot** en inglés).

A modo de ejemplo, elaboraremos el boxplot de la variable **edad**. Para ello emplearemos la siguiente sintaxis:

```
boxplot(practica$edad)
```

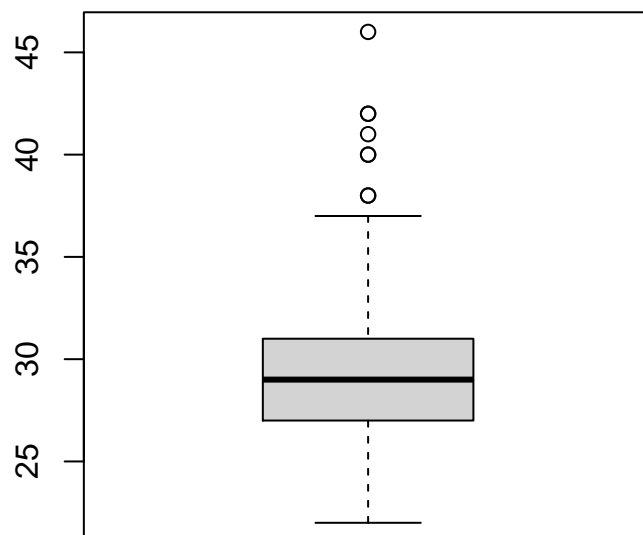


Figura 2.4: Diagrama de cajas para Edad

Como resultado se obtiene la gráfica de la Figura 2.4, donde los bigotes indican el valor mínimo y máximo observado en la variable **edad** y las líneas de la caja indican el valor de los tres cuartiles, siendo la línea más gruesa el cuartil 2 (el C_{50}) y las líneas inferior y superior indican los cuartiles 1 y 2 (los C_{25} y C_{75}), respectivamente. En cuanto a los puntos que aparecen por encima del bigote superior, son los “valores atípicos.”

También es posible elaborar un **diagrama de cajas y bigotes** para una misma variable diferenciando entre grupos. A continuación se muestra la sintaxis empleada para estudiar la variabilidad de la **edad** por separado en los varones y mujeres de la muestra:

```
boxplot(practica$edad ~ practica$genero)
```

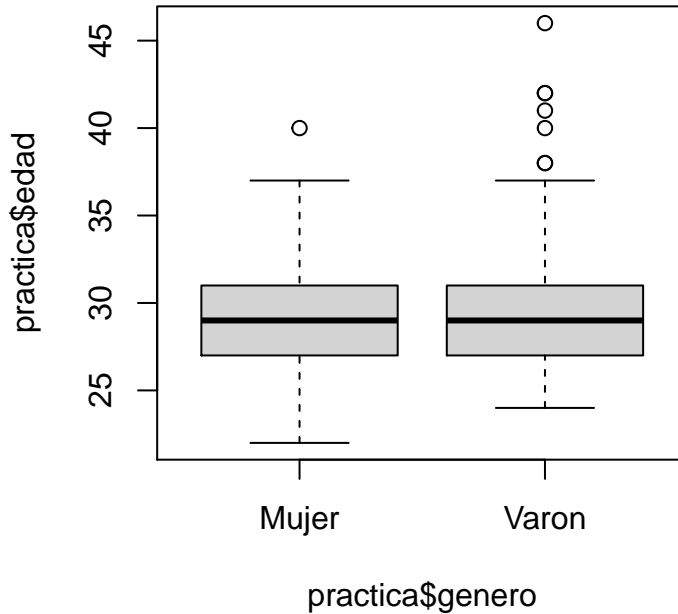


Figura 2.5: Diagrama de cajas para Edad por genero

Asimetría y curtosis

En cuanto a las propiedades de la forma de la distribución, suele estudiarse la *asimetría* (grado en que los datos se reparten equilibradamente por encima y por debajo de la tendencia central) y la *curtosis* (grado de apuntamiento de la distribución de frecuencias).

El programa *R* ofrece diferentes procedimientos para calcular los índices de *asimetría* y *curtosis*. Por simplicidad, aquí únicamente expondremos uno de ellos, que trabaja con los comandos `skewness` y `kurtosis`. Para utilizar estos comandos debe cargarse la librería "`moments`", que suele estar instalada en *RStudio* pero, si no fuera así, habría que instalarla, tal y como se expuso en el apartado 1.5.2 del capítulo anterior.

Para calcular los índices de *asimetría* y *curtosis* para la variable **edad** puede emplearse la siguiente sintaxis:

```
library(moments)

##
## Attaching package: 'moments'

## The following object is masked from 'package:modeest':
##
##      skewness

skewness(practica$edad)

## [1] 1.19575

kurtosis(practica$edad)

## [1] 5.183692
```

En el ejemplo, la distribución de la variable `edad` es asimétrica positiva y leptocúrtica, como también puede observarse en el gráfico de barras y el histograma que ya vimos en las Figuras 2.1 y 2.2 del apartado 2.2.2.

2.2.4. Transformación de puntuaciones

Con frecuencia resulta útil transformar las puntuaciones originales para facilitar su interpretación. En este apartado veremos algunas de las transformaciones más empleadas: las puntuaciones diferenciales, las puntuaciones típicas y las escalas derivadas.

Puntuaciones diferenciales

Una de las transformaciones más sencillas de los datos originales consiste en calcular las *puntuaciones diferenciales*, que se denotan como x_i y expresan las puntuaciones originales de los sujetos como “desviaciones a la media de su grupo.” Esta transformación permite comparar a los sujetos en función únicamente de su desviación en cuanto a la tendencia central de su grupo, y no tiene en cuenta la variabilidad.

Por ejemplo, para calcular las puntuaciones diferenciales en la variable `edad` para los 10 primeros sujetos del archivo se puede emplear la siguiente sintaxis:

```
dedad <- (practica$edad - mean(practica$edad))
dedad[1:10]
```

```
## [1] 4.65 4.65 -4.35 -3.35 -2.35 -0.35 -1.35 -2.35 3.65 -1.35
```

```
mean(dedad)
```

```
## [1] -1.420723e-15
```

```
sd(dedad)
```

```
## [1] 3.877198
```

Como puede verse, la media de las puntuaciones diferenciales es siempre 0, y su desviación típica es la misma que la obtenida para las puntuaciones directas.

Puntuaciones típicas

Las puntuaciones típicas conllevan una transformación similar a la de las puntuaciones diferenciales pero que también tiene en cuenta la variación. Se denotan como z_i , y son el cociente entre la puntuación diferencial y la desviación típica.

Las puntuaciones típicas tienen la propiedad de que siempre toman la misma media y varianza, 0 y 1, respectivamente.

Continuando con el ejemplo de la variable `edad`, para obtener sus puntuaciones típicas y comprobar que su media es 0 y su varianza 1, se puede utilizar la siguiente sintaxis:

```
zedad <- (practica$edad - mean(practica$edad)) / sd(practica$edad)
mean(zedad)
```

```
## [1] -3.640775e-16
```

```
sd(zedad)
```

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

Las puntuaciones típicas, al igual que los centiles, son también *medidas de posición*, ya que permiten hacer una comparación directa entre las puntuaciones de los sujetos en relación a su grupo. Esta comparación es mucho más completa que la que se realiza con las puntuaciones diferenciales ya que las puntuaciones típicas tienen en cuenta tanto la tendencia central como la variación de los datos.

Escalas derivadas

Las puntuaciones típicas, al tener como media 0 y desviación típica 1, ofrecen muchos valores decimales positivos y negativos. Para facilitar su interpretación y evitar los números negativos, suelen transformarse en las denominadas *escalas derivadas* que se definen como $T_i = a \cdot z_i + b$.

Las escalas derivadas tienen la propiedad de que siempre toman la misma media y varianza, b y a^2 , respectivamente.

Continuando con el ejemplo de la variable **edad**, vamos a obtener una escala derivada **Tedad** con media 50 y desviación típica 10 y demostrar sus propiedades. Para ello utilizaremos la siguiente sintaxis:

```
Tedad <- zedad * 10 + 50  
mean(Tedad)
```

```
## [1] 50
```

```
sd(Tedad)
```

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

2.2.5. Seleccionar casos

Terminaremos este apartado de análisis univariante, explicando cómo seleccionar casos para llevar a cabo análisis descriptivos por grupos. Por ejemplo, para analizar los datos únicamente de los varones emplearemos el comando **subset**:

```
practica2 <- subset.data.frame(practica, genero=="Varon")
```

Como puede verse, hemos llamado **practica2** al nuevo fichero, que solo contiene a los 119 varones del fichero original (**practica**).

Podemos repetir algunos de los análisis que habíamos llevado a cabo en anteriores apartados. Por ejemplo, que nos muestre el resumen descriptivo de todas las variables, que nos dé la tabla de frecuencias para la variable **edad** así como las representaciones gráficas (histograma y diagrama de barras) sólo para los varones:

```
summary(practica2)
```

2.2 Análisis descriptivos univariantes

```
##      Clave      genero      edad      peso      estatura
## Min.      : 3.00  Mujer: 0  Min.      :24.00  Min.      :45.00  Min.      :1.530
## 1st Qu.: 52.50  Varon:119 1st Qu.:27.00  1st Qu.:58.00  1st Qu.:1.630
## Median : 97.00      Median :29.00  Median :63.00  Median :1.690
## Mean   : 97.65      Mean   :29.58  Mean   :64.24  Mean   :1.703
## 3rd Qu.:141.50      3rd Qu.:31.00  3rd Qu.:68.50  3rd Qu.:1.750
## Max.    :199.00      Max.    :46.00  Max.    :92.00  Max.    :1.930
##      prov      idprov      rama
## Length:119      Min.      :12.00  Ciencias Experimentales y de la Salud:26
## Class :character 1st Qu.:12.00  Ciencias Sociales y Juridicas      :40
## Mode  :character Median :28.00  Enseñanzas tecnicas                :27
##      Mean   :29.21  Humanidades                        :16
##      3rd Qu.:46.00  Otros/Varios                      :10
##      Max.    :52.00
##      licen      inteli      compren      orient
## Length:119      Min.      : 9.00  Min.      :12.00  Min.      : 5.972
## Class :character 1st Qu.:16.00  1st Qu.:23.00  1st Qu.: 12.015
## Mode  :character Median :18.00  Median :26.00  Median : 16.804
##      Mean   :18.62  Mean   :25.71  Mean   : 29.901
##      3rd Qu.:21.00  3rd Qu.:29.00  3rd Qu.: 36.626
##      Max.    :28.00  Max.    :32.00  Max.    :113.114
##      extra      respon      emocio      sincer
## Min.      :29.00  Min.      :34.00  Min.      :36.00  Min.      :12.00
## 1st Qu.:38.00  1st Qu.:42.00  1st Qu.:45.00  1st Qu.:20.00
## Median :41.00  Median :45.00  Median :48.00  Median :24.00
## Mean   :41.01  Mean   :45.61  Mean   :48.54  Mean   :22.99
## 3rd Qu.:44.00  3rd Qu.:49.00  3rd Qu.:52.00  3rd Qu.:25.00
## Max.    :53.00  Max.    :56.00  Max.    :60.00  Max.    :34.00
##      fumar
## No fumador:100
## Fumador   : 19
##
##
##
```

```
table(practica2$edad)      # frecuencias absolutas, Edad
```

```
##
## 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 40 41 42 46
## 5  8 15  8 22 14 11  8  7  6  2  2  1  3  2  1  1  2  1
```

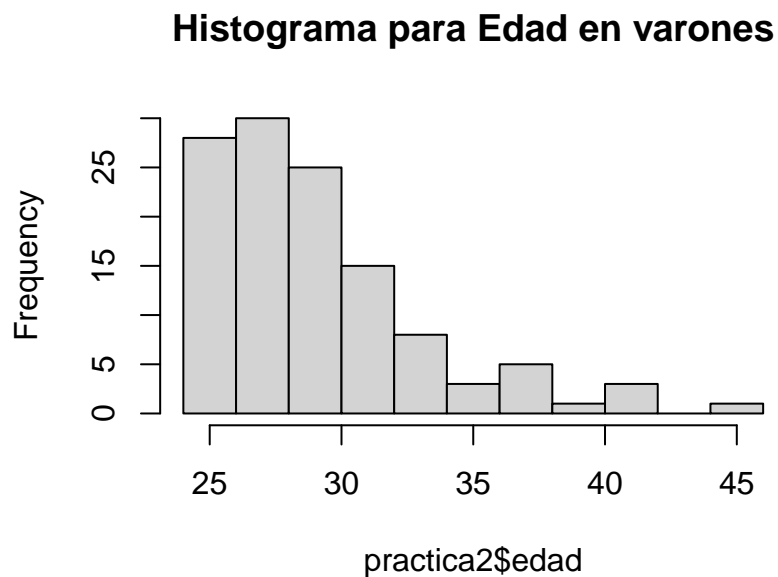
2 Análisis descriptivos

Poner información en tabla de frecuencias

```
data.frame(prop.table(table(practica2$edad)))
```

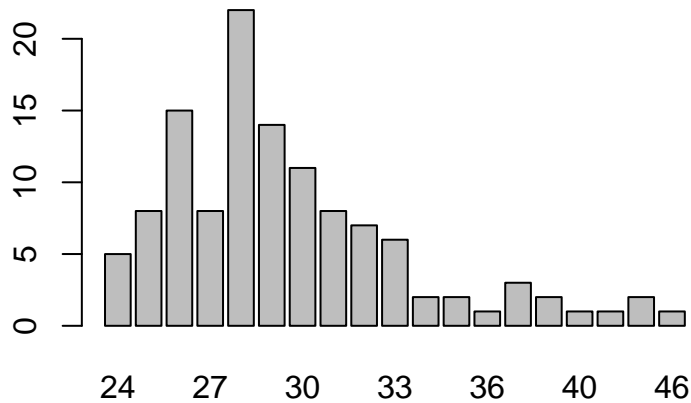
##	Var1	Freq
## 1	24	0.042016807
## 2	25	0.067226891
## 3	26	0.126050420
## 4	27	0.067226891
## 5	28	0.184873950
## 6	29	0.117647059
## 7	30	0.092436975
## 8	31	0.067226891
## 9	32	0.058823529
## 10	33	0.050420168
## 11	34	0.016806723
## 12	35	0.016806723
## 13	36	0.008403361
## 14	37	0.025210084
## 15	38	0.016806723
## 16	40	0.008403361
## 17	41	0.008403361
## 18	42	0.016806723
## 19	46	0.008403361

```
hist(practica2$edad, main="Histograma para Edad en varones")
```




```
barplot(table(practica2$edad), main="Diagrama de barras para Edad en varones")
```

Diagrama de barras para Edad en varones



Hay otras formas de resumir la información por grupos, que no requieren filtrar los casos. Por ejemplo, con *R* también es posible **segmentar el archivo**, como hacíamos con el programa *SPSS*. En este caso, se necesita utilizar el comando `aggregate`.

Veamos un ejemplo para la variable `edad`, donde se solicitan los cálculos de la *media* y la *desviación típica*, segmentando por `sexo`:

```
aggregate(practica$edad, by=list(practica$genero), mean)
```

```
##   Group.1      x
## 1   Mujer 29.01235
## 2   Varon 29.57983
```

```
aggregate(practica$edad, by=list(practica$genero), sd)
```

```
##   Group.1      x
## 1   Mujer 3.455046
## 2   Varon 4.138484
```

Para un resumen de todos los códigos de *R* para llevar a cabo análisis descriptivos univariantes puede verse el anexo 2.

2.3. Análisis descriptivos bivariantes

En este apartado veremos cómo llevar a cabo análisis descriptivos con dos variables. Seguiremos el mismo esquema que en el manual de la asignatura *Análisis de Datos I* de Botella, Suero y Ximénez (2012) y el cuaderno de prácticas de *Análisis de datos con SPSS* de Ximénez y Revuelta (2011).

2.3.1. Covarianza y Correlación

Comenzaremos viendo cómo cuantificar la relación lineal entre dos variables cuantitativas. Para ello, calcularemos dos índices: la **covarianza** y la **correlación**.

Continuando con el ejemplo del archivo `practica`, cuantificaremos el grado de relación lineal que existe entre las variables `peso` y `estatura`. Para ello usaremos los comandos `cov` y `cor`:

```
cov(practica$peso, practica$estatura)
```

```
## [1] 0.7461789
```

```
cor(practica$peso, practica$estatura)
```

```
## [1] 0.8568278
```

Como puede verse, la covarianza entre `peso` y `estatura` es 0,75 y la correlación de Pearson es 0,86.

2.3.2. Gráficos de dispersión

Una primera aproximación al análisis de la relación lineal es mediante su representación gráfica. En el caso de dos variables cuantitativas, puede elaborarse un *Diagrama de Dispersión*. La forma de la “nube de puntos” nos permitirá valorar de forma visual el grado de relación lineal entre las variables.

Para obtener un *diagrama de dispersión* con *R* se utiliza el comando `plot(x,y)`. En el argumento pueden definirse las propiedades más básicas del gráfico. Por ejemplo: el título (comando `main`), las leyendas de los ejes (`xlab` e `ylab`), así como los límites de los ejes (`xlim` e `ylim`).

Continuando con el ejemplo del `peso` y la `estatura`, para obtener un diagrama de dispersión de la Figura 2.6 puede emplearse la siguiente sintaxis:

```
plot(practica$peso, practica$estatura,
     main = "Gráfico de dispersión",
     xlab = "Peso", ylab = "Estatura",
     xlim=c(40, 90), ylim=c(1.5, 1.90))
points(practica$peso, practica$estatura, col = "black", pch = 19)
```

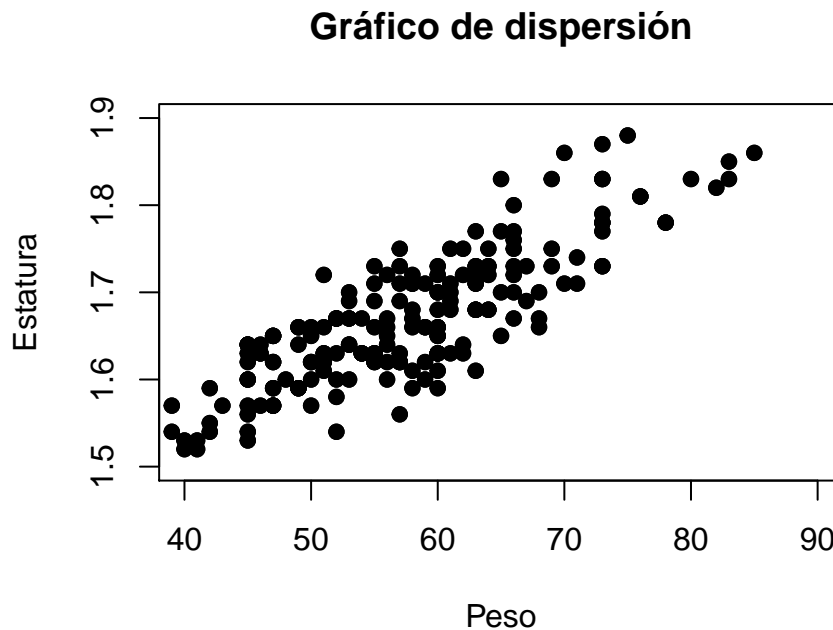


Figura 2.6: Diagrama de dispersión para Peso y Estatura

En la sintaxis se ha indicado también el color de los puntos (`col= "black"`), que es en negro (si no se indica nada, aparecen transparentes) y también se ha definido su forma (`pch = 19`, que son círculos rellenos del color definido en `col`).

2.3.3. Matriz de Varianzas-covarianzas y Matriz de Correlaciones

Cuando trabajamos con tres o más variables conviene organizar la información de forma resumida en una *Matriz de varianzas-covarianzas* o en una *Matriz de correlaciones*. Para elaborar estas matrices se utilizan también los comandos `cov` y `cor`, pero en este caso incluyendo como argumento todas las variables que queramos que aparezcan dentro de las respectivas matrices.

A continuación se muestra un ejemplo sobre cómo elaborar una matriz de varianzas-covarianzas y una matriz de correlaciones incluyendo las variables `extraversión`, `responsabilidad`, `estabilidad emocional` y `sinceridad`:

```
cov(practica[, c("extra", "respon", "emocio", "sincer")])
```

```
##           extra   respon   emocio   sincer
## extra  18.482814  9.422111  8.467337  3.695075
## respon  9.422111 21.979271  9.831030  4.247487
## emocio  8.467337  9.831030 24.748116  2.237437
## sincer  3.695075  4.247487  2.237437 23.682312
```

```
cor(practica[, c("extra", "respon", "emocio", "sincer")])
```

```
##           extra   respon   emocio   sincer
## extra  1.0000000  0.4674740  0.39590566  0.17661485
## respon  0.4674740  1.0000000  0.42152304  0.18617159
## emocio  0.3959057  0.4215230  1.00000000  0.09242038
## sincer  0.1766148  0.1861716  0.09242038  1.00000000
```

La *matriz de correlaciones* se puede expresar también de forma gráfica mediante los correspondientes diagramas de dispersión (véase Figura 2.7):

```
plot(practica[, c("extra", "respon", "emocio", "sincer")])
```

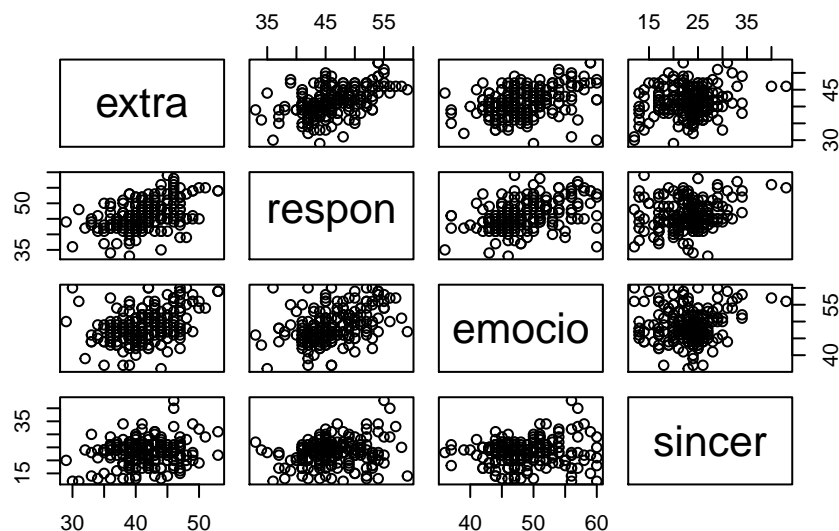


Figura 2.7: Diagrama de dispersión múltiple

Tal y como vimos en el capítulo anterior, si quisiéramos obtener estos mismos análisis pero solo para la muestra de mujeres, podríamos emplear la siguiente sintaxis:

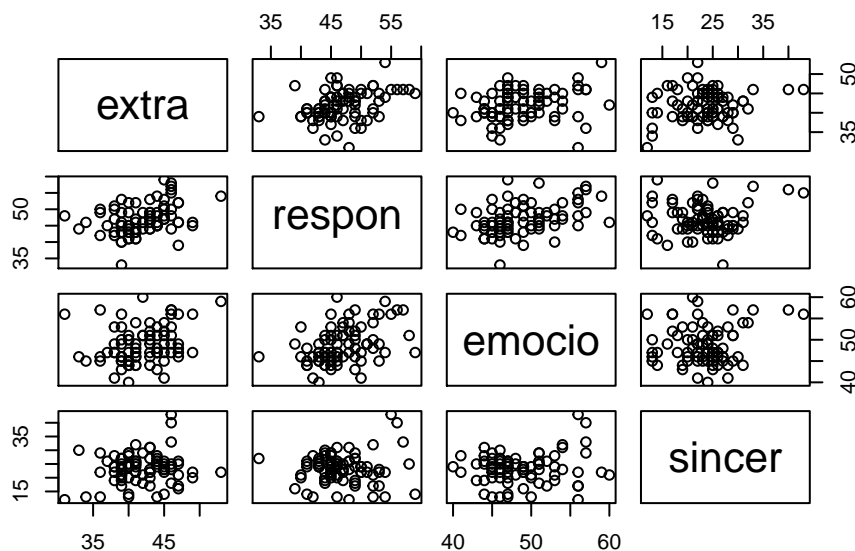
```
cov(subset(practica, genero== "Mujer")[, c("extra", "respon",
                                           "emocio", "sincer")])
```

```
##           extra    respon    emocio    sincer
## extra  14.684568  6.336111  4.544136  3.160185
## respon  6.336111 21.200000  8.447222  1.691667
## emocio  4.544136  8.447222 19.213272  3.195370
## sincer  3.160185  1.691667  3.195370 30.227778
```

```
cor(subset(practica, genero== "Mujer")[, c("extra", "respon",
                                           "emocio", "sincer")])
```

```
##           extra    respon    emocio    sincer
## extra  1.0000000 0.35910733 0.2705329 0.14999571
## respon 0.3591073 1.00000000 0.4185481 0.06682574
## emocio 0.2705329 0.41854805 1.0000000 0.13259203
## sincer 0.1499957 0.06682574 0.1325920 1.00000000
```

```
plot(subset(practica, genero== "Mujer")[, c("extra", "respon",
                                           "emocio", "sincer")])
```



2.3.4. Puntuaciones combinadas

A continuación veremos cómo elaborar combinaciones lineales de variables del tipo:

$$Y_i = X_1 + X_2 + \dots + X_p$$

y cómo calcular la media y varianza de Y y demostrar sus propiedades.

Veamos un ejemplo. Si definimos la variable $Y = \text{extraversión} + \text{estabilidad emocional}$, entonces para calcular su media y varianza usaremos la siguiente sintaxis:

```
Y = practica$extra + practica$emocio  
mean(Y)
```

```
## [1] 89.985
```

```
var(Y)
```

```
## [1] 60.1656
```

Se obtiene que la media de Y es 89,99 y la varianza de Y es 60,17.

Se trata de una transformación del tipo $Y_i = X_1 + X_2$ donde la media de Y es $\bar{Y}_i = \bar{X}_1 + \bar{X}_2$ y la varianza de Y es $S_Y^2 = S_1^2 + S_2^2 + 2 \cdot S_{12}$.

Para demostrar estas propiedades podemos usar la siguiente sintaxis:

```
mean(practica$extra)
```

```
## [1] 41.36
```

```
mean(practica$emocio)
```

```
## [1] 48.625
```

```
cov(practica[, c("extra", "emocio")])
```

```
##           extra      emocio  
## extra 18.482814  8.467337  
## emocio  8.467337 24.748116
```

Como puede verse, en el caso de la media de Y se cumple que: $\bar{Y}_i = \bar{X}_1 + \bar{X}_2 = 41,36 + 48,63 = 89,99$. Y en el caso de la varianza de Y se cumple que: $S_Y^2 = S_1^2 + S_2^2 + 2 \cdot S_{12} = 18,48 + 24,75 + 2 \cdot 8,47 = 60,17$.

2.3.5. Regresión lineal simple

El análisis de regresión lineal es una técnica estadística que se utiliza para analizar la relación lineal entre variables. En la investigación psicológica suele emplearse para pronosticar valores en una variable dependiente (Y) a partir de las puntuaciones en una variable independiente (X) y se denomina *Regresión de Y sobre X*, donde:

$$Y'_i = A_{yx} + B_{yx} \cdot X_i$$

Veamos un ejemplo. Llevaremos a cabo la regresión de la variable **peso** (variable dependiente) sobre la variable **estatura** (variable independiente). Para obtener los coeficientes de la ecuación de regresión utilizaremos la función `lm`. En el argumento, para indicar el rol de las variables se utiliza el símbolo `~`. En nuestro ejemplo:

```
regresion <- lm(peso ~ estatura, data = practica) # VD: peso; VI: estatura
summary(regresion)
```

```
##
## Call:
## lm(formula = peso ~ estatura, data = practica)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.9764  -4.1448  -0.2696   3.8717  11.6217
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -118.375      7.565  -15.65  <2e-16 ***
## estatura      105.437      4.509   23.38  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.351 on 198 degrees of freedom
## Multiple R-squared:  0.7342, Adjusted R-squared:  0.7328
## F-statistic: 546.8 on 1 and 198 DF, p-value: < 2.2e-16
```

Los resultados de `summary` nos dan tanto los análisis descriptivos como los inferenciales (aún no estudiados en la asignatura de primero pero se verán en el apartado 6.2.1 del capítulo 6). Lo que más nos interesa de estos resultados son los valores de A_{yx} y B_{yx} que son -118,38 y 105,44, respectivamente. Según esto, la ecuación de regresión puede escribirse como:

$$Peso'_i = -118,38 + 105,44 \cdot Estatura_i$$

Bondad del Modelo

El comando `summary` proporciona también el valor del *Coefficiente de determinación* (Multiple R-squared). En este caso, $R^2 = 0,7342$, lo que indica que el modelo explica un 73,42% de la varianza (*bondad de ajuste del modelo* en términos relativos).

También es posible obtener la valoración de la bondad del modelo en términos absolutos (como varianza explicada). Para ello se necesita obtener las *puntuaciones pronosticadas por el modelo* (Y'_i) y los *residuos* ($Y_i - Y'_i$) para los sujetos del fichero. A continuación se muestran los comandos necesarios para obtener estos valores (`predict` y `resid`) y los resultados para los 10 primeros sujetos del archivo *practica.sav*:

```
pre <- predict(regresion, data.frame(practica)) # valores pronosticados
pre[1:10]
```

```
##          1          2          3          4          5          6          7          8
## 53.48708 62.97639 42.94340 66.13950 58.75892 54.54145 60.86766 54.54145
##          9         10
## 60.86766 53.48708
```

```
res <- resid(regresion) # residuos
res[1:10]
```

```
##          1          2          3          4          5          6
## -7.4870818 -11.9763940  2.0565983 -2.1394980 -0.7589219 -9.5414499
##          7          8          9         10
## -0.8676580 -9.5414499 -0.8676580  0.5129182
```

Obtenidos los valores pronosticados (`pre`) y los residuos (`res`), es posible llevar a cabo la *Descomposición de la varianza del criterio*. Esto es:

$$S_Y^2 = S_{Y'}^2 + S_{Y-Y'}^2$$

En nuestro ejemplo, se calcula la varianza en la variable dependiente (`peso`) y las varianzas de los pronósticos (`pre`) y de los residuos (`res`). Esto es:

```
var(practica$peso) # varianza de la VD
```

```
## [1] 107.1638
```

```
var(pre) # varianza explicada por el modelo
```

```
## [1] 78.67472
```



```
var(res)           # varianza no explicada por el modelo
```

```
## [1] 28.48908
```

Como se observa en los resultados la varianza de la variable **peso** es 107,16 y puede descomponerse en la siguiente suma: $107,16 = 78,67 + 28,49$. En términos relativos diríamos que el modelo explica un 73,42% de la varianza y deja sin explicar el restante 26,58%.

Representación gráfica de la recta de regresión

Para obtener una representación gráfica de la recta de regresión Y' superpuesta (comando `abline`) en el diagrama de dispersión de X e Y puede utilizarse la siguiente sintaxis:

```
plot(practica$estatura, practica$peso, main = "Regresión de Peso
      sobre Estatura", xlab = "Estatura", ylab = "Peso")
abline(regresion)
```

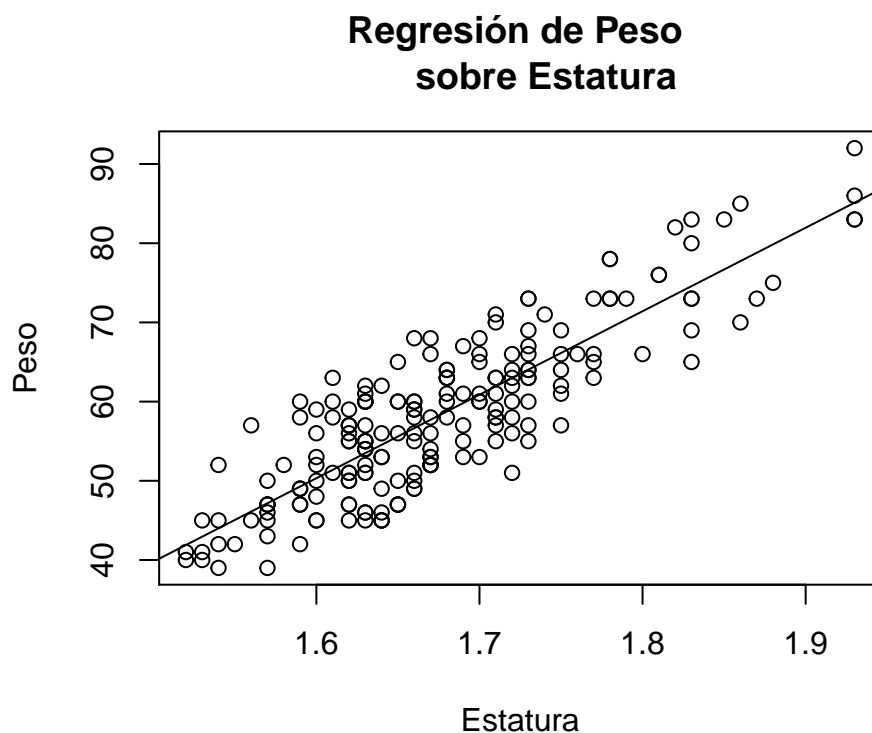


Figura 2.8: Regresión lineal de Peso sobre Estatura

2.3.6. Tablas de contingencia

Cuando se trabaja con variables cualitativas se realizan distribuciones conjuntas de sus categorías. Para ello se elaboran las denominadas *Tablas de contingencia*.

Por ejemplo, las variables `sexo` y `fumar` son dos variables cualitativas. Para elaborar una Tabla de Contingencia de ambas variables incluyendo las categorías de `sexo` como filas y las de `fumar` como columnas, se utiliza el comando `fable` en la siguiente sintaxis:

```
(tc <- ftable(practica$genero, practica$fumar)) # frecuencias absolutas
```

```
##           No fumador Fumador
##
## Mujer           38      43
## Varon           100      19
```

Para obtener las tablas de frecuencias en términos relativos hay que aplicar una serie de comandos que nos permitan obtener en primer lugar las frecuencias marginales (`addmargins`) y a continuación las tres tablas de frecuencias relativas: las conjuntas y las condicionales por filas y por columnas. La sintaxis a emplear es la siguiente:

```
addmargins(tc) # frecuencias marginales
```

```
##           Sum
##      38 43  81
##     100 19 119
## Sum 138 62 200
```

```
prop.table(tc) # frecuencias relativas conjuntas
```

```
##           No fumador Fumador
##
## Mujer           0.190  0.215
## Varon           0.500  0.095
```

```
prop.table(tc, 1) # frecuencias relativas condicionadas por filas
```

```
##           No fumador  Fumador
##
## Mujer    0.4691358 0.5308642
## Varon    0.8403361 0.1596639
```

```
prop.table(tc,2) # frecuencias relativas condicionadas por columnas
```

```
##           No fumador  Fumador
##
## Mujer    0.2753623 0.6935484
## Varon    0.7246377 0.3064516
```

A modo de ejemplo, a partir de estas tres tablas podemos dar las siguientes interpretaciones:

- Del total de sujetos, un 19 % son mujeres y no fuman.
- De las mujeres, un 53,09 % fuman.
- De los que fuman, un 69,35 % son mujeres.

Por último, la representación gráfica de la distribución de frecuencias conjuntas de la Figura 2.9 se obtiene mediante:

```
barplot(tc, main = "Grafica Tabaquismo y Genero",
        xlab = "Mujer: negro, Varon: gris", ylab = "frecuencias")
```

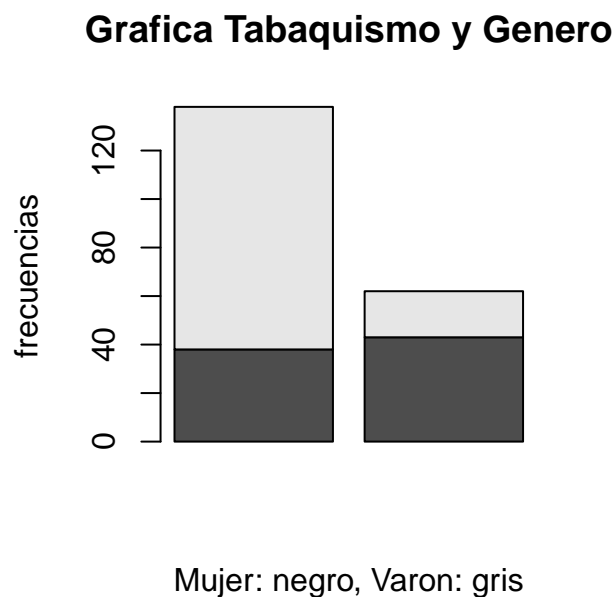


Figura 2.9: Gráfica de la distribución de frecuencias conjuntas

Para un resumen de todos los códigos de R para llevar a cabo los análisis descriptivos bivariantes vistos aquí puede verse el anexo 3.

2.4. Ejercicios propuestos

1. ¿Cuál es la **estatura** mínima y máxima de los sujetos de la muestra? ¿Y la de los varones? ¿Y la de las mujeres?
2. ¿Qué porcentaje de sujetos tiene una **estatura** menor de 1,65 m.?
3. ¿Cuál es el valor central de la variable **estatura**? ¿Y de la variable **peso**?
4. Obtenga el valor del **peso** que es superado por el 15 % de la muestra
5. ¿Cuántos sujetos fuman?
6. ¿Cuántos sujetos son de provincias de Andalucía?
7. ¿En cuál de las cuatro características de personalidad (**extraversión**, **responsabilidad**, **estabilidad emocional** o **sinceridad**) han obtenido los sujetos menores puntuaciones?
8. Obtenga la representación gráfica más adecuada para las siguientes variables: **sexo**, **estatura**, **inteli** y **rama**.
9. ¿Qué porcentaje de varones obtiene en **extraversión** una puntuación de más de 40 puntos?
10. Confeccione el *diagrama de cajas* para la variable **estatura** para la muestra de varones, para la de mujeres y para la muestra total y trate de interpretar el resultado.
11. Confeccione la gráfica que considere más adecuada para la variable **licen**.
12. Sabiendo que para que los sujetos sean seleccionados por la empresa es necesario que obtengan en **responsabilidad** como mínimo una puntuación de 52 ¿Cuántos sujetos de la muestra total serán seleccionados?
13. ¿Quiénes obtienen mayores puntuaciones en **estabilidad emocional**, los varones o las mujeres?
14. ¿Son los sujetos de humanidades igual de homogéneos en la variable **orientación espacial** que los de enseñanzas técnicas?
15. Obtenga los estadísticos descriptivos univariados y la representación gráfica más adecuada para la variable **sinceridad** en la muestra total, en la de varones y en la de mujeres.
16. Elabore una escala derivada con media 60 y desviación típica 10 para la variable **estabilidad emocional** y su representación gráfica.
17. Confeccione el histograma para la variable **edad** en la muestra total, en la de varones y en la de mujeres. Comente los resultados obtenidos en cada grupo.

18. Elabore un informe descriptivo sobre la variable **edad** expresada en *meses*. En dicho informe tienen que aparecer los estadísticos de tendencia central, de variabilidad, las propiedades de la distribución y una representación gráfica.
19. Se desea comparar las características físicas de los sujetos evaluados en esta muestra con las de otra muestra de sujetos norteamericanos. Para ello, utilizaremos algunos datos del fichero **practica**. Más concretamente, las variables **estatura** y **peso**. El objetivo es comparar nuestros datos descriptivos en estas variables con los de una muestra norteamericana, que tiene similares características. Tenemos el problema de que en Estados Unidos el peso y la estatura se miden en escalas diferentes a las nuestras. En concreto, el peso en libras y la estatura en pulgadas, siendo:

1 LIBRA = 453,6 gramos

1 PULGADA = 2,54 cm

Lo primero es obtener los datos de ambas variables en la escala de medida norteamericana. Después hay que elaborar un informe con los estadísticos descriptivos. EL INFORME ha de incluir lo siguiente:

- a) La media y la varianza de las variables **estatura** y **peso** en la escala norteamericana
 - b) Un gráfico que dé cuenta de la variabilidad en cada una de las variables para los varones y las mujeres
 - c) Información sobre la forma de la distribución (asimetría y curtosis)
 - d) Las puntuaciones típicas para la variable **estatura**
 - e) Una escala derivada T con media 100 y desviación típica 20 para la variable **estatura** y su representación gráfica
 - f) La covarianza entre ambas variables (calcularlo en formato español y norteamericano).
20. Calcule la media y la varianza para la variable que resulta de la suma de las variables **inteligencia**, **comprensión verbal** y **orientación espacial**.
21. Se desea predecir las puntuaciones en **estabilidad emocional** a partir de una de las siguientes variables: **inteligencia**, **extraversión**, **responsabilidad** y **sinceridad**. Seleccione la variable más apropiada como predictora, justificando la elección. A continuación, conteste a las siguientes cuestiones:
 - a) Represente gráficamente la relación entre la variable predictora escogida y el criterio.
 - b) Obtenga la ecuación de regresión correspondiente (en puntuaciones directas, diferenciales y típicas).
 - c) Descomponga la varianza del criterio para el modelo anterior e interprete la bondad del modelo (de forma descriptiva y con el gráfico de dispersión con la recta superpuesta).

- d) ¿Cuál es la proporción de varianza explicada de la variable **estabilidad emocional** a partir de la variable predictora que se haya empleado?
 - e) Si un sujeto obtiene una puntuación de 6 puntos en la variable predictora, ¿Cuál es su puntuación pronosticada en **estabilidad emocional**?
 - f) Si se quisiera predecir las puntuaciones en **estabilidad emocional** a partir de dos variables, ¿Cuál añadiría? ¿Por qué? ¿Cuánto mejoraría la bondad del nuevo modelo?
22. Obtenga la tabla de frecuencias conjuntas para las variables **sexo** y **rama** y la gráfica de barras con ambas variables.
23. ¿Cuál es el porcentaje de sujetos que son varones y de humanidades? ¿y el de los que dentro de los de enseñanzas técnicas, son mujeres?
24. Elabore la misma tabla de frecuencias que en el punto anterior pero separando entre fumadores y no fumadores
25. Repita lo mismo que en los ejercicios 22 a 24 con las variables **fumar** y **rama** e interprete los resultados obtenidos.
26. Confeccione una gráfica en la que aparezca la distribución de frecuencias para la variable **edad** en varones y mujeres e interprete los resultados.
27. Confeccione una gráfica que represente la relación entre las variables **rama**, **sexo** y **responsabilidad**.

Nota: todos los ejercicios se refieren al archivo **practica** (véase anexo 1). Para responder a estos ejercicios quizá te ayude consultar las tablas de los anexos 2 y 3, que incluyen un resumen de los comandos que se han visto en este capítulo para llevar a cabo análisis descriptivos con una y dos variables con R , respectivamente.

3 Probabilidad: introducción a los modelos de distribución

En este capítulo veremos cómo asociar áreas de probabilidad a variables aleatorias discretas y continuas. Comenzaremos explicando los comandos que utiliza *R* para definir conceptos importantes como son los de *función de probabilidad* o *función de densidad de probabilidad* $f(x_i)$, *función de distribución* $F(x_i)$ y *función de distribución inversa* $F^{-1}(p)$, que es la puntuación X_i que corresponde a cierta $F(x_i)$.

A continuación, veremos en detalle los modelos de distribución de probabilidad vistos en la asignatura de *Análisis de Datos I* (i.e., modelo binomial, normal, t de Student, chi-cuadrado de Pearson y modelo F de Snedecor) y cómo asociar áreas de probabilidad a los valores de la variable, así como valores a ciertas áreas de probabilidad.

3.1. Conceptos previos

El lenguaje *R* tiene asociadas cuatro funciones para cada distribución de probabilidad, que utilizan los comandos: **d**, **p**, **q** y **r** seguido del nombre de la distribución:

Función	Descripción
d	Función de probabilidad o función de densidad de probabilidad, $f(x_i)$
p	Función de distribución, $F(x_i)$
q	Función de distribución inversa, $F^{-1}(p)$, o valor X_i que corresponde a una $F(x_i)$
r	Función que proporciona una muestra aleatoria (ver apartado 3.8)

Función de probabilidad y función de densidad de probabilidad

El comando **d** proporciona la función de probabilidad $f(x_i) = P(X = x_i)$ correspondiente a cada valor de una variable aleatoria discreta X (e.g., una binomial) así como la función de densidad asociada a cierto valor de una variable aleatoria continua (e.g., una normal). El comando **d** también permite elaborar la representación gráfica de $f(x_i)$. Para ello, en primer lugar se define el rango de valores de la variable X , después se calcula el valor de la función de probabilidad (o de densidad de probabilidad) para cada valor X_i y finalmente se representa ambas cantidades.

Función de distribución

El comando `p` proporciona la probabilidad acumulada asociada a cierto valor de una variable X (la función de distribución $F(x_i)$) tanto para variables aleatorias discretas como continuas. Esto es útil tanto para los modelos de distribución de probabilidad como para el contexto de los *contrastes de hipótesis* (por ejemplo, si en un contraste unilateral izquierdo sobre una media hemos encontrado que el estadístico Z toma el valor -1,68 y queremos conocer el nivel crítico o la probabilidad que queda a la izquierda de dicho estadístico: $P(Z \leq -1,68)$, usaremos la función `p`).

Función de distribución inversa

El comando `q` proporciona la *Función de distribución inversa*: $F^{-1}(p)$, donde p es la función de distribución $F(x_i)$. Es decir, dada una probabilidad acumulada p , el comando `q` proporciona el valor de X_i al que le corresponde dicha probabilidad acumulada. Este comando es útil para resolver ejercicios de probabilidad donde queremos conocer la puntuación X_i que corresponde a cierta área de probabilidad acumulada ($F(x_i)$), pero también para resolver ejercicios de contraste de hipótesis (e.g. un contraste de hipótesis unilateral izquierdo sobre la media), y saber cuál es el punto crítico que deja a su izquierda la probabilidad de 0,05.

3.2. Distribuciones incluidas en R

R tiene disponibles todos los siguientes modelos de probabilidad:

Distribución	Función de probabilidad o de densidad de probabilidad
Beta	<code>dbeta</code>
Binomial	<code>dbinom</code>
Cauchy	<code>dcauchy</code>
Chi-cuadrado	<code>dchisq</code>
Exponencial	<code>dexp</code>
F	<code>df</code>
Gamma	<code>dgamma</code>
Ggeométrica	<code>dgeom</code>
Hipergeométrica	<code>dhyper</code>
Log-normal	<code>dlnorm</code>
Multinomial	<code>dmultinom</code>
Binomial negativa	<code>dnbinom</code>
Normal	<code>dnorm</code>
Poisson	<code>dpois</code>
t de Student	<code>dt</code>
Uniforme	<code>dunif</code>
Weibull	<code>dweibull</code>

Aquí veremos solo los que aparecen en la tabla en negrita (para otros modelos véase Revuelta y Ponsoda, 2005). Todas las distribuciones utilizan comandos similares y permiten hacer las mismas operaciones (asignar áreas de probabilidad, valores a ciertas áreas de probabilidad, etc.). Nos detendremos en más detalle en las distribuciones binomial y normal para ver más tarde la chi-cuadrado de Pearson, t de Student y F de Snedecor.

3.3. Modelo Binominal

En primer lugar, tenemos que definir nuestra variable aleatoria X . Para ello, generaremos una muestra aleatoria de datos mediante el comando `seq`. A modo de ejemplo, trabajemos con una m.a.s. de $N = 12$ sujetos y con una binomial con $\pi = 0,20$. Generaremos también su representación gráfica en un diagrama de barras (comando `barplot`):

```
Xb <- seq(0, 12, by = 1)
fb <- dbinom(Xb, 12, .20)
barplot(fb, xlab="X", ylab="f(x)")
```

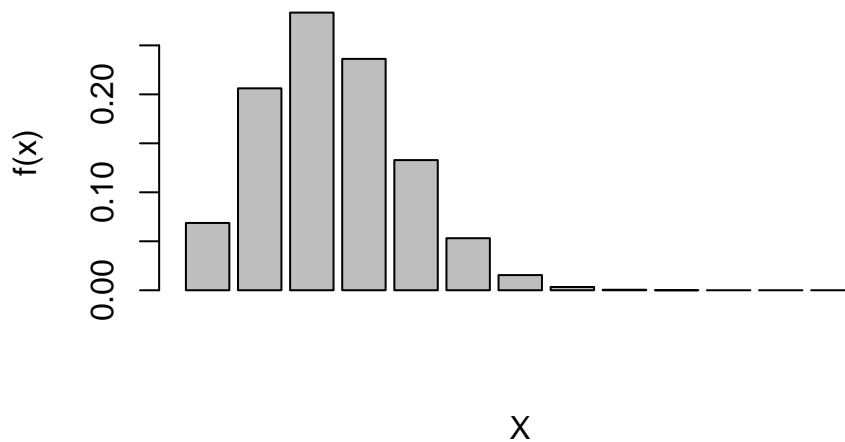


Figura 3.1: Representación gráfica de una binomial, $B(12; 0,20)$

Aquí hemos utilizado el comando `seq` para generar una muestra aleatoria de datos donde el mínimo es 0 y el máximo 12, y corresponden a los números enteros del 0 al 12 (`by = 1`) que definen la variable `Xb`. La función de probabilidad (denominada `fb`) es una binomial referida a la variable `Xb` con $N = 12$ y $\pi = 0,20$. Como puede verse en la Figura 3.1, el gráfico de barras (`barplot`) para este ejemplo tiene forma asimétrica positiva.

3 Probabilidad: introducción a los modelos de distribución

Si hubiéramos definido una binomial con $\pi = 0,80$, como en el siguiente ejemplo, la distribución sería asimétrica negativa (véase Figura 3.2):

```
Xb <- seq(0, 12, by = 1)
fb <- dbinom(Xb, 12, .80)
barplot(fb, xlab="X", ylab="f(x)")
```

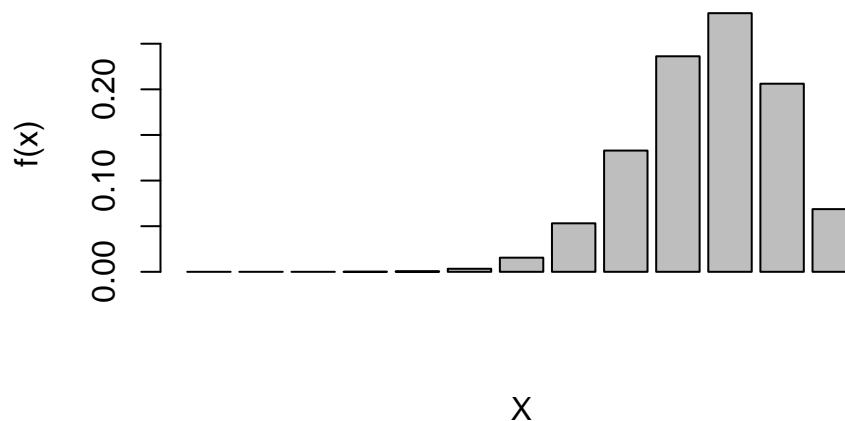


Figura 3.2: Representación gráfica de una binomial, $B(12; 0,80)$

Continuando con el ejemplo de la binominal con $N = 12$ y $\pi = 0,20$, realizaremos ejercicios similares a los vistos en clase. A modo de ejemplo, comenzaremos calculando:

- La función de probabilidad para $X_i = 0$ (es decir $f(0) = P(X_i = 0)$)
- La función de distribución para $X_i = 5$ (es decir $F(5) = P(X_i \leq 5)$)

Para ello, utilizaremos los siguientes comandos de *R*:

```
# Función de probabilidad para x = 0
(dbinom(0, 12, .20))      # f(0) en una B(12; 0,20)
```

```
## [1] 0.06871948
```

```
# Función de distribución para x = 5
(pbinom(5, 12, .20))      # F(5) en una B(12; 0,20)
```

```
## [1] 0.9805947
```

Como puede verse: $f(0) = P(X_i = 0) = 0,07$ y $F(5) = P(X_i \leq 5) = 0,98$.

Veamos ahora otro ejemplo. Si quisiéramos conocer la **probabilidad del área derecha**, trabajaríamos con comandos similares que los que nos han permitido calcular $F(5)$, pero en este caso, para pedirle a *R* que nos dé el área contraria tendríamos que trabajar con el comando `lower.tail=FALSE`. Esto es:

```
# Área derecha a una F(x)
(pbinom(5, 12, .20, lower.tail=FALSE)) # 1-F(5): Área derecha al valor x= 5
```

```
## [1] 0.01940528
```

Como puede verse: $P(X_i > 5) = 1 - F(5) = 0,02$.

Por último, veremos **cómo calcular el área comprendida entre dos valores**. Por ejemplo para calcular $P(2 \leq X \leq 4)$ tendríamos que restar dos funciones de distribución. En este caso $F(4) - F(1)$. Esto puede indicarse en *R* del siguiente modo:

```
# Área comprendida entre dos valores
sum(dbinom(2:4, 12, .20)) # P(2 <= X <= 4) para una B(12; 0,20)
```

```
## [1] 0.6525666
```

Como puede verse, la $P(2 \leq X \leq 4) = 0,65$.

A continuación veremos ejemplos de **Función de distribución inversa**, $F^{-1}(p)$, o lo que es lo mismo, cómo obtener el valor X_i que corresponde a cierta función de distribución $F(x_i)$. Continuando con el ejemplo de nuestra variable X que se distribuye según $B(12; 0,20)$, obtendremos los centiles C_{25} y C_{50} . Para ello, utilizaremos los siguientes comandos de *R*:

```
# Valor de X que corresponde al Centil 25 en una B(12; 0,20)
(qbinom(.25, 12, .20))
```

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

```
# Valor de X que corresponde al Centil 50 en una B(12; 0,20)
(qbinom(.50, 12, .20))
```

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

Como puede verse, $C_{25} = 1$ y $C_{50} = 2$.

Es posible realizar otros ejemplos cambiando los parámetros de la binomial. Al final de este capítulo hay ejercicios relacionados con la binomial.

3.4. Modelo Normal

Comenzaremos con los modelos de distribución para variables continuas y nos detendremos especialmente en el modelo normal. En cuanto a su representación gráfica, de forma similar a como hemos visto con la binomial, en primer lugar tenemos que definir nuestra variable aleatoria X . Para ello, generaremos una muestra aleatoria de datos mediante el comando `seq`. En este caso trabajaremos con la normal unitaria y por tanto con puntuaciones típicas z_i en un rango de -3 a +3 puntos en intervalos de 0,01 puntos. A continuación generaremos su representación gráfica en un diagrama de líneas (comando `plot` y `type="l"` donde l es línea), como muestra la Figura 3.3:

```
x <- seq(-3, 3, by=0.01)
f <- dnorm(x)
plot(x, f, type="l", lwd=1.5)
```

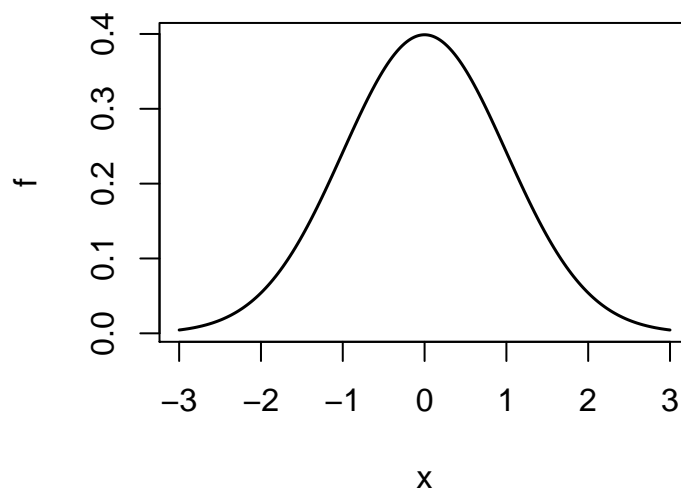


Figura 3.3: Representación gráfica de la Normal unitaria, $N(0; 1)$

A continuación veremos ejemplos de diferentes familias de variables normales. Si trabajamos con una variable X con valores entre $X_{min} = 70$ y $X_{max} = 130$, que sigue el modelo normal con media $\mu = 100$, veamos cómo sería la forma de la distribución al variar el parámetro σ . Más concretamente, cuando $\sigma = 15$ (gráfico con línea continua de color negro), $\sigma = 10$ (gráfico con línea discontinua de color rojo) y $\sigma = 5$ (gráfico con línea de puntos de color azul).

Para elaborar estos tres gráficos utilizaremos los siguientes comandos:

```

x <- seq(70, 130, by=0.1)      # Generar valores entre 70 y 130 puntos
f1 <- dnorm(x, mean=100, sd=15) # Función de densidad para una N(100; 15)
f2 <- dnorm(x, mean=100, sd=10) # Función de densidad para una N(100; 10)
f3 <- dnorm(x, mean=100, sd=5)  # Función de densidad para una N(100; 5)
plot(x, f1, type="l", lwd=1.5, ylim=c(0, 0.09), ylab="densidad")
# añadir líneas al grafico anterior
lines(x, f1, type="l", lwd=1.5, lty=5, col="black", text(100, .021, "N(100; 15)"))
lines(x, f2, type="l", lwd=1.5, lty=5, col="red", text(100, .043, "N(100; 10)"))
lines(x, f3, type="l", lwd=1.5, lty=3, col="blue", text(100, .083, "N(100; 5)"))

```

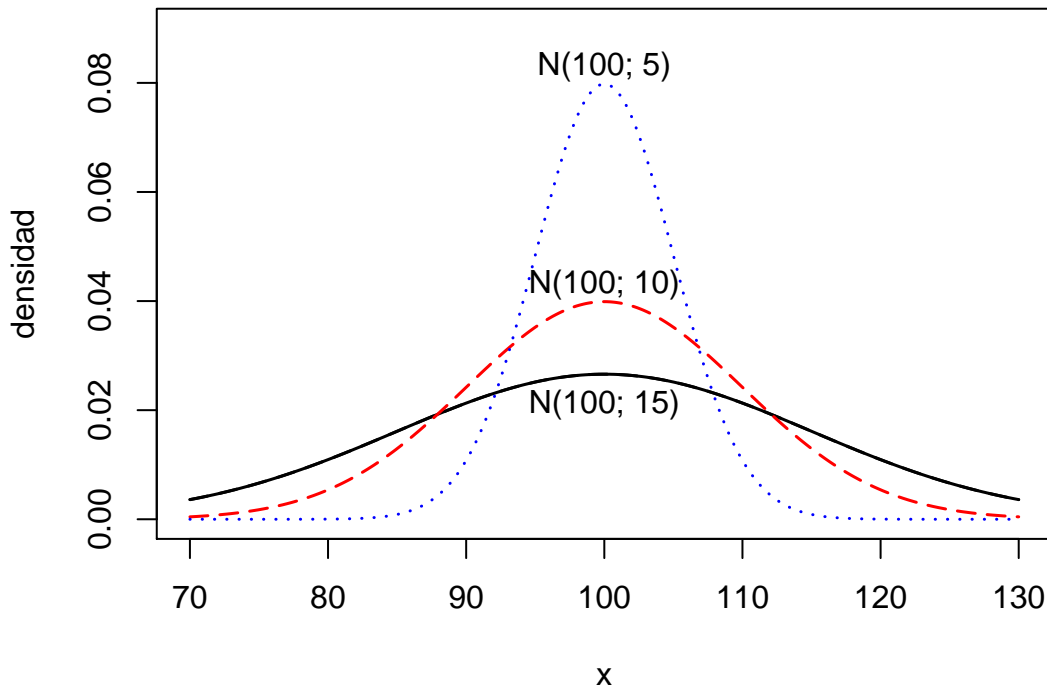


Figura 3.4: Representación gráfica de la normal con diferentes varianzas

Como puede verse en la Figura 3.4, la forma del gráfico varía notablemente al alterarse el valor de σ .

A continuación veremos cómo asignar áreas de probabilidad a ciertos valores de la variable. Trabajaremos con la normal unitaria $N(0; 1)$. En el siguiente ejemplo se solicita a R la función de distribución $F(z_i)$ para una $z_i = -1.68$, así como la probabilidad del lado derecho para esa misma puntuación típica ($z_i = -1.68$):

3 Probabilidad: introducción a los modelos de distribución

```
pnorm(-1.68) # Área acumulada F(z) para z=-1,68
```

```
## [1] 0.04647866
```

```
pnorm(-1.68, lower.tail=FALSE) # Área derecha (1-F(z)) para z=-1,68
```

```
## [1] 0.9535213
```

Como puede verse: $P(z \leq -1,68) = 0,05$ y $P(z \geq -1,68) = 0,95$.

En caso de que queramos obtener la puntuación típica z_i que corresponde a cierta área de probabilidad acumulada $F(z_i)$, usaremos el comando `qnorm`. Veamos algunos ejemplos:

```
qnorm(0.05) # Proporciona la z para F(z) = .05
```

```
## [1] -1.644854
```

```
qnorm(0.05, lower.tail=FALSE) # Proporciona la z para 1-F(z) = .05
```

```
## [1] 1.644854
```

```
qnorm(0.95) # Proporciona la z para F(z) = .95
```

```
## [1] 1.644854
```

Como puede verse: $z_{0,05} = -1,64$ y $z_{0,95} = 1,64$.

A continuación resolveremos algunos ejercicios que son muy similares a los que hemos resuelto a mano con la ayuda de las tablas de probabilidad:

```
#¿Cuál es la probabilidad de puntuar en X como máximo 12 puntos?  
pnorm((12-10)/2)
```

```
## [1] 0.8413447
```

```
#¿Qué puntuación X corresponde al Centil 75?  
(p1 <- qnorm(0.75)) # valor en puntuación típica
```

```
## [1] 0.6744898
```

```
(X1 <- p1 * 2 + 10)    # Transformación a puntuaciones directas
```

```
## [1] 11.34898
```

La respuesta a cada una de las cuestiones es:

1. Probabilidad de puntuar en X como máximo 12 puntos: $P(X \leq 12) = 0,84$
2. Puntuación X que corresponde al Centil 75: $C_{25} = 11,35$.

3.5. Modelo Chi-cuadrado de Pearson

A continuación veremos el modelo chi-cuadrado de Pearson. Como se ha hecho con los anteriores modelos de distribución, comenzaremos elaborando su representación gráfica. En primer lugar tenemos que definir nuestra variable aleatoria X . Para ello, generaremos una muestra aleatoria de datos mediante el comando `seq`.

En este caso, como los valores de la variable son siempre positivos, trabajaremos con una variable definida en un rango de valores de 0 a 60 puntos tomados en intervalos de 0,1 puntos.

A continuación generaremos su representación gráfica en un diagrama de líneas mediante el comando `plot`.

La siguiente sintaxis muestra varios ejemplos de distribuciones chi-cuadrado (con 10, 20 y 40 grados de libertad), todos ellos representados en la Figura 3.5:

```
x2 <- seq(0, 60, by=0.1) # Generar valores entre 0 y 60 en intervalos de 0.1 puntos
f1c <- dchisq(x2, 10)    # Función de densidad para una chi-cuadrado con gl = 10
f2c <- dchisq(x2, 20)    # Función de densidad para una chi-cuadrado con gl = 20
f3c <- dchisq(x2, 40)    # Función de densidad para una chi-cuadrado con gl = 40
plot(x2, f1c, type="l", lwd=1.5, ylim=c(0, 0.10), ylab="densidad")
# añadir la línea al grafico anterior
lines(x2, f1c, type="l", lwd=1.5, lty=5, col="black", text(14, .09, "chi(10)"))
lines(x2, f2c, type="l", lwd=1.5, lty=5, col="red", text(25, .06, "chi(20)"))
lines(x2, f3c, type="l", lwd=1.5, lty=3, col="blue", text(46, .04, "chi(40)"))
```

A continuación se muestran ejemplos sobre cómo asignar áreas de probabilidad a ciertos valores de una chi-cuadrado con 10 grados de libertad. Se solicita a R la función de distribución $F(x_i)$ para una $X_i = 3,94$, así como la probabilidad asociada al lado derecho de esa misma puntuación ($\chi^2_{10} = 3,94$):

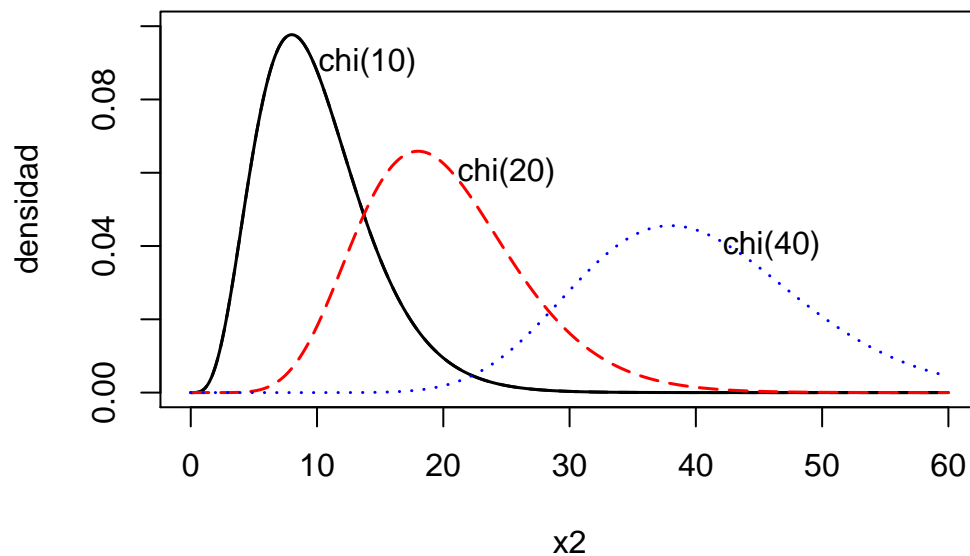


Figura 3.5: Representación gráfica de distribuciones chi-cuadrado con diferentes gl

```
# Función de distribución para 3,94 en una Chi-cuadrado con gl = 10
pchisq(3.94, 10)
```

```
## [1] 0.04998691
```

```
# 1-F(x) para 3,94 en una chi-cuadrado con 10 grados de libertad
pchisq(3.94, 10, lower.tail=FALSE)
```

```
## [1] 0.9500131
```

Como puede verse: $P(\chi_{10}^2 \leq 3,94) = 0,05$ y $P(\chi_{10}^2 \geq 3,94) = 0,95$.

También es posible obtener la puntuación asociada a cierta función de distribución. Por ejemplo, continuando con el ejemplo de una chi-cuadrado con 10 grados de libertad, el C_{25} corresponde a la puntuación $\chi_{10}^2 = 6,74$:

```
# Puntuación chi-cuadrado con 10 grados de libertad para F(x) = .25
qchisq(.25, 10)
```

```
## [1] 6.737201
```


3.6. Modelo t de Student

A continuación veremos el modelo t de Student con k grados de libertad (gl). Como se ha hecho con los anteriores modelos de distribución, comenzaremos elaborando su representación gráfica. En primer lugar tenemos que definir nuestra variable aleatoria X . Para ello, generaremos una muestra aleatoria de datos mediante el comando `seq`. En este caso, los valores de la variable pueden ser positivos y negativos, siendo la media 0, y trabajaremos con una variable definida en un rango de valores entre -3 y 3 puntos tomados en intervalos de 0,1 puntos. A continuación generaremos su representación gráfica en un diagrama de líneas mediante el comando `plot`.

La siguiente sintaxis muestra varios ejemplos de distribuciones t de Student (con 10, 40 y 60 grados de libertad), representados en la Figura 3.6:

```
# Generar valores entre -3 y 3 puntos en intervalos de 0.1 puntos
T <- seq(-3, 3, by=0.1)
f1t <- dt(T, 10)           # Función de densidad para una t de Student con gl=10
f2t <- dt(T, 40)           # Función de densidad para una t de Student con gl=40
f3t <- dt(T, 60)           # Función de densidad para una t de Student con gl=60
plot(T, f1t, type="l", lwd=1.5, ylim=c(0, 0.40), ylab="densidad")
# añadir la línea al grafico anterior
lines(T,f1t,type="l",lwd=1.5,lty=5,col="black")
lines(T,f2t,type="l",lwd=1.5,lty=5,col="red")
lines(T,f3t,type="l",lwd=1.5,lty=3,col="blue")
```

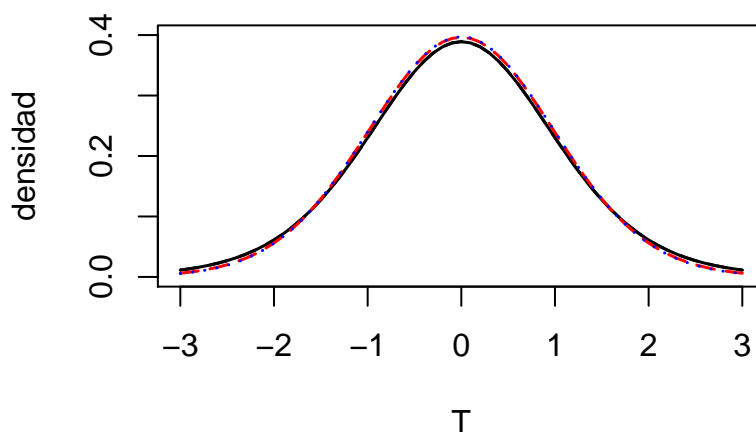


Figura 3.6: Representación gráfica de distribuciones t de Student con diferentes gl

3 Probabilidad: introducción a los modelos de distribución

Como puede verse, la diferencia entre `f1t`, `f2t` y `f3t` no es tan marcada como sucedía con los anteriores modelos.

A continuación se muestran ejemplos sobre cómo asignar áreas de probabilidad a ciertos valores de una *t* de Student con 10 grados de libertad. Se solicita a *R* la función de distribución $F(x_i)$ para una $X_i = 2$, así como la probabilidad asociada al lado derecho de esa misma puntuación ($t_{10} = 2$):

```
pt(2, 10) # Función de distribución para X=2 en una t de Student con gl=10
```

```
## [1] 0.963306
```

```
pt(2, 10, lower.tail=FALSE) # 1-F(x) para 2 en una t de Student con gl=10
```

```
## [1] 0.03669402
```

Como puede verse: $P(t_{10} \leq 2) = 0,96$ y $P(t_{10} \geq 2) = 0,04$.

También es posible obtener la puntuación asociada a cierta función de distribución. Por ejemplo, continuando con el ejemplo de una *t* de Student con 10 grados de libertad, el C_{25} corresponde a la puntuación $t_{10} = -0,70$:

```
qt(.25, 10) # Puntuación t de Student con gl=10 para F(x) = .25
```

```
## [1] -0.6998121
```

3.7. Modelo F de Snedecor

Por último veremos el modelo F de Snedecor con m y n grados de libertad. Como se ha hecho con los anteriores modelos de distribución, comenzaremos elaborando su representación gráfica. En primer lugar tenemos que definir nuestra variable aleatoria X . Para ello, generaremos una muestra aleatoria de datos mediante el comando `seq`.

En este caso, los valores de la variable sólo pueden ser positivos. Trabajaremos con una variable definida en un rango de valores entre 0 y 5 puntos tomados en intervalos de 0,1 puntos. A continuación generaremos su representación gráfica en un diagrama de líneas mediante el comando `plot`.

La siguiente sintaxis muestra varios ejemplos de distribuciones F de Snedecor (con 5 y 15 grados de libertad, con 10 y 20 grados de libertad y con 20 y 30 grados de libertad), y su representación gráfica aparece en la Figura 3.7:

```
# Generar valores entre 0 y 5 puntos en intervalos de 0.1 puntos
X <- seq(0, 5, by=0.1)
f1f <- df(X, 5, 15)      # Función de densidad para una F de Snedecor con gl 5 y 15
f2f <- df(X, 10, 20)     # Función de densidad para una F de Snedecor con gl 10 y 20
f3f <- df(X, 20, 30)     # Función de densidad para una F de Snedecor con gl 20 y 30
plot(X, f1f, type="l", lwd=1.5, ylim=c(0, 1.05), ylab="densidad")
# añadir la línea al grafico anterior
lines(X,f1f,type="l",lwd=1.5,lty=5,col="black",text(2.9, .17, "F(5, 15)",col="black"))
lines(X,f2f,type="l",lwd=1.5,lty=5,col="red", text(2.3, .34, "F(10, 20)",col="red"))
lines(X,f3f,type="l",lwd=1.5,lty=5,col="blue",text(1.8, .87, "F(20, 30)",col="blue"))
```

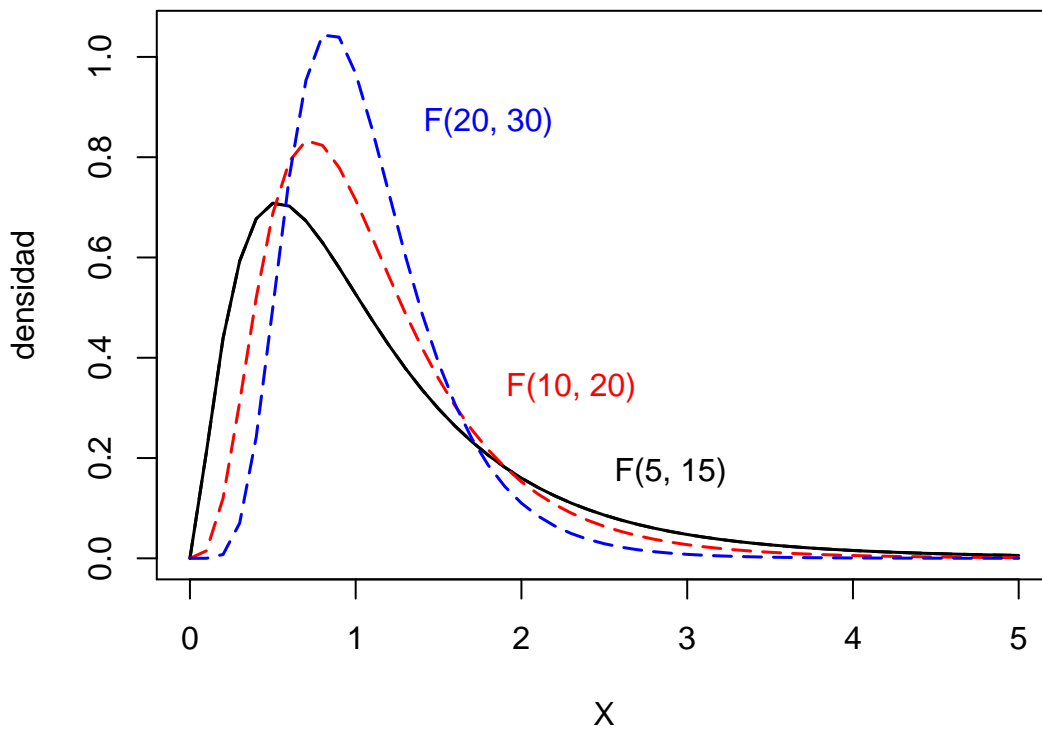


Figura 3.7: Representación gráfica de distribuciones F de Snedecor con diferentes gl

3 Probabilidad: introducción a los modelos de distribución

A continuación se muestran ejemplos sobre cómo asignar áreas de probabilidad a ciertos valores de F de Snedecor con 10 y 15 grados de libertad. Se solicita a R la función de distribución $F(x_i)$ para una $X_i = 2,544$, así como la probabilidad asociada al lado derecho de esa misma puntuación ($F_{10,15} = 2,544$):

```
# Función de distribución para 2,544 en una F con gl 10 y 15
pf(2.544, 10, 15)
```

```
## [1] 0.9500195
```

```
# 1-F(x) para 2,544 en una F con gl 10 y 15
pf(2.544, 10, 15, lower.tail=FALSE)
```

```
## [1] 0.04998046
```

Como puede verse: $P(F_{10,15} \leq 2,544) = 0,95$ y $P(F_{10,15} \geq 2,544) = 0,05$.

También es posible obtener la puntuación asociada a cierta función de distribución. Por ejemplo, continuando con el ejemplo de una F de Snedecor con 10 y 15 grados de libertad, el C_{25} corresponde a la puntuación $F_{10,15} = 0,65$:

```
qf(.25, 10, 15)      # Puntuación F con gl 10 y 15 para F(x) = .25
```

```
## [1] 0.6519848
```

En los anteriores apartados hemos visto cómo manejar los diferentes modelos de distribución de probabilidad. Los comandos vistos aquí permitirán al estudiante utilizar un software en lugar de las tablas de probabilidad en papel (**para un resumen de los códigos de R para Probabilidad puedes consultar el anexo 4**). Esto facilitará el trabajo, al tiempo que permitirá acceder a cualquier valor con exactitud (recordemos que las tablas de libro están limitadas a ciertos valores y habíamos aprendido a “aproximar” los que no venían en tablas, y esto ya no será necesario). Adicionalmente, los gráficos que hemos visto permiten comprender mejor los modelos de distribución y cómo cambian en función de sus parámetros.

Terminaremos explicando en un apartado final en más detalle la idea de “simulación de datos” que hemos introducido al explicar cómo elaborar los gráficos de los diferentes modelos de distribución vistos en este capítulo.

3.8. Simular datos

En este tema se ha introducido el concepto de *generación de datos* para elaborar las gráficas de cada uno de los modelos de distribución que hemos explicado.

A la generación de datos se la denomina también **simulación de datos**. Finalizaremos este capítulo ampliando un poquito más el concepto de *simulación de datos* e introduciendo ejemplos sencillos (para iniciarse en el concepto de simulación en más detalle puedes consultar el libro de Revuelta y Ponsoda, 2003).

la Función `rnorm` permite simular una muestra de datos de tamaño N . Vamos a comprobarlo con un pequeño estudio de simulación. Tomaremos una muestra de tamaño $N = 9$ de una distribución $N(100; 15)$ y a continuación calcularemos los siguientes estadísticos: \bar{X} , S_X^2 y $S_{\bar{X}}^2$.

```
muestra <- rnorm(9, 100, 15)
print(muestra)
```

```
## [1] 106.12187  87.83156  82.88985 121.67456  86.87056 147.53921 100.13610
## [8] 107.34460 106.59021
```

```
mean(muestra)
```

```
## [1] 105.2221
```

```
sd(muestra)
```

```
## [1] 20.10104
```

```
sd(muestra)/sqrt(length(muestra))
```

```
## [1] 6.700347
```

Como puede verse, se han generado datos para 9 personas donde: $\bar{X} = 96,60$, $S_X^2 = 13,79$ y $S_{\bar{X}}^2 = 4,60$.

Veremos a continuación un ejemplo más sofisticado. Simularemos 100 matrices de datos de tamaño nueve para una distribución $N(100; 15)$. A continuación calcularemos la media de cada muestra, obtendremos el histograma de frecuencias de las 100 medias (ver Figura 3.8), la media de las medias muestrales ($E(\bar{X})$) y su desviación típica ($\sigma_{\bar{X}}$).

```
muestra <- rnorm(900, 100, 15)
muestra <- matrix(muestra, nrow=9)
medias <- colMeans(muestra)
hist(medias, ylab="Frecuencia", main="Histograma de medias")
legend(82, 30,
      c(paste("Media = ", sprintf("%.2f", mean(medias))),
        paste("Sd = ", sprintf("%.2f", sd(medias)))),
      bty = "n")
```

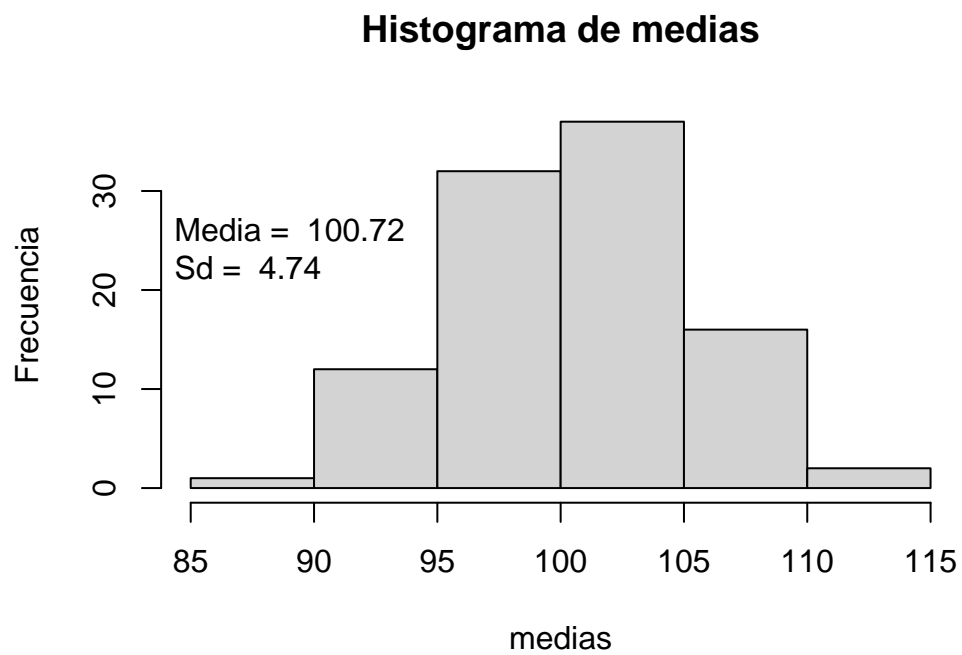


Figura 3.8: Histograma de las 100 medias simuladas

3.9. Ejercicios propuestos

Conteste a las siguientes preguntas utilizando la sintaxis del lenguaje R:

Ejercicio 1: Binomial

El 70 % de pacientes que sufren una determinada enfermedad se curan al aplicarles un tratamiento. Si tomamos una m.a.s. de 10 pacientes con esa enfermedad y pasan el tratamiento...

1. ¿Cuál es la probabilidad de que se recuperen todos?
2. ¿Cuál es la probabilidad de que se recuperen al menos la mitad?
3. ¿Cuál es la probabilidad de que se recuperen como máximo 6 pacientes?
4. ¿Cuál es la probabilidad de que se recuperen entre 2 y 4 (ambos inclusive)?

Ejercicio 2: Normal

La variable CI se distribuye $N(100; 15)$ en la población de estudiantes universitarios. Se selecciona una m.a.s. de 200 sujetos. Si se extrae un sujeto al azar...

1. ¿Cuál es la probabilidad de que puntúe en CI como máximo 120?
2. ¿Cuál es la probabilidad de que puntúe en CI al menos 95?
3. ¿Qué puntuación en CI tiene una probabilidad acumulada de 0,5793?
4. ¿Qué puntuación en CI ocupa el centil 65?

Ejercicio 3: Chi-cuadrado de Pearson

La variable X se distribuye según el modelo chi-cuadrado con 10 grados de libertad. Si se extrae un sujeto al azar...

1. Calcule la probabilidad de que no supere el valor 9,342
2. Calcule la puntuación X que corresponde a la probabilidad acumulada de 0,70
3. Calcule la probabilidad de que la variable X adopte como mínimo el valor 15,987

Ejercicio 4: t de Student

La variable X se distribuye según el modelo t de Student con 25 grados de libertad. Si se extrae un sujeto al azar...

1. Calcule la probabilidad de que no supere el valor 2,06
2. Calcule el valor de X tal que la probabilidad de obtener como máximo ese valor sea 0,70

3. Si dividimos la distribución en cuatro partes iguales, ¿Qué valores generan dicha partición?

Ejercicio 5: F de Snedecor

La variable X se distribuye según el modelo $F_{7,8}$ Si se extrae un sujeto al azar...

1. Calcule la probabilidad de que no supere el valor 3,5
2. Calcule el valor X tal que la probabilidad de obtener como mínimo ese valor sea 0,75
3. Calcule el valor de X tal que la probabilidad de obtener como máximo ese valor sea 0,90
4. ¿Qué valor corresponde al primer cuartil?

Ejercicio 6: Comparación de distribuciones

Vamos a comparar las distribuciones $N(0; 1)$, t_1 y t_{10} :

1. Represente gráficamente las tres funciones de densidad
2. Obtenga el centil 95 de cada distribución
3. Tome 50 muestras de tamaño 100 de cada una de estas distribuciones y obtenga el valor esperado y la desviación típica de la distribución muestral de la media

Distribución	Centil	$\hat{\mu}_{\bar{X}}$	$\hat{\sigma}_{\bar{X}}$
$N(0;1)$			
t_1			
t_{10}			

Nota: Para responder a estos ejercicios quizá te ayude la tabla del anexo 4, que incluye un resumen de los comandos que se han visto en este capítulo para llevar a cabo ejercicios sobre modelos de distribuciones de probabilidad.

4 Contrastes de hipótesis sobre uno y dos parámetros

En este capítulo veremos diferentes contrastes sobre uno y dos parámetros: el contraste para una y dos medias, el de igualdad de varianzas y el contraste sobre una correlación.

4.1. Lectura y preparación de datos del archivo *terapia.dat*

Tanto en este capítulo como en los siguientes, utilizaremos el archivo de datos *terapia.dat* para ejemplificar los contrastes de hipótesis (el archivo aparece descrito en el anexo 5). Recuerda también que antes de empezar a trabajar debes seguir la rutina para iniciar la sesión con *RStudio* y preparar tu script (repasa el apartado 2.1).

Como ya sabemos, antes de realizar contrastes de hipótesis es conveniente inspeccionar los datos mediante análisis descriptivos para obtener una primera impresión de las variables. Algunos comandos útiles para este propósito son:

```
typeof(datos)  # Indica qué tipo de objeto es datos
names(datos)   # Muestra los nombres de las variables
std(datos)     # Muestra los primeros casos de cada variable
```

Según vemos en este ejemplo, las variables están contenidas dentro del objeto `datos`. Recordemos que si lo que queremos es acceder a una variable dentro de `datos`, tenemos que utilizar la sintaxis `datos$variable`. Por ejemplo, escribiendo `sd(datos$noche1)` obtenemos la desviación típica de la variable `noche1`.

Otra forma sencilla de acceder a las variables es llamándolas por su nombre, omitiendo el prefijo `datos$`. Para ello utilizamos el comando `attach`, que le dice a *R* que utilice las variables directamente sin tener que poner el nombre del objeto en el que están almacenadas. Por ejemplo, la media de `noche1` puede obtenerse del siguiente modo:

```
attach(datos)
mean(noche1)
```

Una vez que hayamos realizado la operación `attach(datos)` ya no es necesario volverla a repetir. En adelante *R* sabrá encontrar todas las variables que están dentro del objeto `datos` (descrito en el anexo 5). Nos referiremos a `datos` para explicar los contrastes vistos en los siguientes apartados.

4.2. Prueba Z para una media

La prueba Z se utiliza en el contraste de hipótesis sobre una media cuando la varianza poblacional, σ^2 , es conocida. El estadístico de contraste Z es:

$$Z = \frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}}$$

y su distribución es la Normal(0,1). No existe una función específica de *R* para calcular el estadístico Z , por lo que debemos programarla en *R*.

Por ejemplo, supongamos que disponemos de los siguientes datos referidos a puntuaciones en el “Test PMA de Thurstone” de un grupo de ocho personas con Alzheimer precoz:

94 102 88 91 97 94 101 93

El propósito del análisis es “averiguar si el cociente intelectual medio es menor en la población de la que proceden estas personas que en la población general.” Para ello realizamos un contraste unilateral izquierdo con las siguientes hipótesis:

$$\begin{aligned} H_0 : \mu &\geq 100 \\ H_1 : \mu &< 100 \end{aligned}$$

Como sabemos que el cociente intelectual tiene $\sigma = 15$ en la población, podemos hacer el contraste con el estadístico Z . Para ello programamos el cálculo de Z del siguiente modo:

```
x <- c(94,102,88,91,97,94,101,93)
z <- (mean(x) - 100)/(15/sqrt(8))
```

El resultado de este cálculo es $Z = -0,94$. Para tomar una decisión sobre H_0 podemos emplear dos maneras equivalentes:

- Buscar el **punto crítico**, z_α , que es el valor de Z que deja a su izquierda la probabilidad $\alpha = 0,05$. Como vimos en el capítulo anterior, esto puede hacerse con el comando `qnorm(0.05)`, donde se ha utilizado un nivel de significación $\alpha = 0,05$.

- Calcular el **nivel crítico** unilateral izquierdo: $p = P(Z \leq -0,94)$. Para ello empleamos la función `pnorm(z)`, que proporciona la probabilidad acumulada de la distribución normal para el estadístico de contraste definido en `z`.

El punto crítico con $\alpha = 0,05$ es $z_\alpha = -1,64$, y el nivel crítico vale $p = 0,17$. Con cualquiera de estos métodos llegamos a la decisión de mantener H_0 . En consecuencia, no podemos concluir que en la población de personas con Alzheimer precoz el cociente intelectual medio sea inferior al de la población general.

4.3. Prueba T sobre una media

Cuando el valor de σ^2 es desconocido, los contrastes sobre medias se realizan con la prueba T . En R está implementada la función `t.test` para realizar el contraste sobre una y dos medias.

Como ejemplo de contraste sobre una media con σ^2 desconocida, podemos contrastar la hipótesis de que la variable `noche1`, contenida en `datos`, tiene media 5 en la población. Las hipótesis son:

$$H_0 : \mu = 5$$

$$H_1 : \mu \neq 5$$

El código R para este contraste es:

```
attach(datos)
t.test(nochel, mu=5)
```

La salida de resultados muestra el estadístico de contraste (`t = 2.4611`) y el nivel crítico (`p-value = 0.0236`), por lo que la decisión sobre H_0 será una u otra dependiendo del nivel de α empleado. La hipótesis nula se mantiene con $\alpha = 0,01$ y se rechaza con $\alpha = 0,05$.

```
##
## One Sample t-test
##
## data:  noche1
## t = 2.4611, df = 19, p-value = 0.0236
## alternative hypothesis: true mean is not equal to 5
## 95 percent confidence interval:
##  5.108417 6.341583
## sample estimates:
## mean of x
##      5.725
```

Los resultados también muestran el intervalo de confianza para la media poblacional con $\alpha = 0,05$, que es: (5.108417, 6.341583).

Otras opciones

La función `t.test` admite algunas variaciones para realizar contrastes unilaterales y cambiar el nivel de confianza. El argumento `alternative` permite especificar la forma de la hipótesis alternativa para indicar si el contraste es unilateral izquierdo (`less`), unilateral derecho (`greater`) o bilateral (`two.sided`). El argumento `conf.level` se utiliza para indicar el nivel de confianza. Veamos algunos ejemplos:

```
t.test(noche1, mu=5)
t.test(noche1, mu=5, alternative="greater")
t.test(noche1, mu=5, alternative="less", conf.level=0.99)
```

4.4. Contraste sobre dos medias y dos varianzas independientes

En un diseño inter-sujetos, disponemos de dos grupos formados por diferentes personas y queremos comparar las medias de los grupos. A este análisis se le denomina contraste de dos medias independientes. El contraste tiene dos versiones: una en la que se asumen varianzas iguales en los dos grupos comparados y otra en la que las varianzas de los grupos son distintas. En ambos casos se calcula un estadístico T , pero el modo de calcularlo y sus grados de libertad son distintos.

Para escoger una de estas versiones de la prueba T (varianzas poblacionales iguales o distintas), debemos empezar realizando un **Contraste sobre la igualdad de las varianzas** de los grupos, cuyas hipótesis son:

$$\begin{aligned}H_0 : \sigma_1^2 &= \sigma_2^2 \\ H_1 : \sigma_1^2 &\neq \sigma_2^2\end{aligned}$$

En función del resultado del contraste de varianzas, nos decantaremos por la prueba T para dos medias independientes con varianzas iguales o con varianzas distintas.

Supongamos que queremos comparar las medias de `noche1` en varones y mujeres. El primer paso es utilizar la función `var.test` para realizar el *contraste de igualdad de varianzas*. El argumento que le pasamos a `var.test` es la fórmula `noche1~sexo`, que significa que la variable dependiente es `noche1` y, por tanto, vamos a comparar la varianza de `noche1` en los grupos formados por la variable independiente `sexo`.

```
var.test(nochel~sexo)
```

Como puede verse más abajo, el contraste de hipótesis sobre la igualdad de varianzas proporciona un estadístico de contraste F y su nivel crítico (p-value). En este ejemplo el resultado no es estadísticamente significativo (p-value = 0.9966).

```
##
## F test to compare two variances
##
## data:  noche1 by sexo
## F = 1.0029, num df = 9, denom df = 9, p-value = 0.9966
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.2491025 4.0376106
## sample estimates:
## ratio of variances
##          1.002885
```

Al haber mantenido la hipótesis de igualdad de varianzas, aplicaremos la prueba T de igualdad de medias en la versión que asume varianzas iguales. Las hipótesis son:

$$H_0 : \mu_1^2 = \mu_2^2$$

$$H_1 : \mu_1^2 \neq \mu_2^2$$

Utilizaremos la función `t.test`, donde el argumento `var.equal=TRUE` de `t.test` indica que se aplique la prueba T asumiendo varianzas iguales en la población:

```
t.test(nochel~sexo, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data:  noche1 by sexo
## t = -1.0015, df = 18, p-value = 0.3299
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -1.8277227  0.6477227
## sample estimates:
## mean in group 0 mean in group 1
##          5.43          6.02
```

El resultado del contraste de medias incluye el estadístico T (resulta ser $t = -1.0015$), el nivel crítico ($p\text{-value} = 0.3299$) y el intervalo de confianza para la diferencia de medias $(-1.8277227, 0.6477227)$. Por tanto, al no ser el resultado significativo, no se puede rechazar la hipótesis nula sobre igualdad de medias.

Por último, supongamos que el *contraste de igualdad de varianzas* hubiese rechazado la hipótesis de homocedasticidad (o igualdad de varianzas). Entonces habríamos realizado la prueba T en su versión de varianzas distintas. Lo habríamos hecho igual que en el ejemplo anterior pero omitiendo el argumento `var.equal=TRUE`, es decir, mediante:

```
t.test(noche1~sexo)
```

Nótese por tanto que esta prueba requiere dos contrastes: el de igualdad de varianzas y el de igualdad de medias, y en ese orden.

4.5. Contraste sobre dos medias relacionadas

En un diseño intra-sujetos, disponemos de dos medias calculadas con los mismos sujetos. Por ejemplo, en un diseño secuencial se evalúa a un grupo de personas antes y después de una intervención (un tratamiento, un programa de aprendizaje, etc.). En un diseño transversal, se observan dos variables en un mismo momento, como podían ser las puntuaciones en dos test psicológicos diferentes. En ambos casos se aplica la prueba T sobre medias relacionadas.

La prueba T para dos medias relacionadas se aplica cuando queremos contrastar la igualdad de dos medias poblacionales que corresponden a dos variables medidas sobre los mismos sujetos, en un *diseño de medidas repetidas*. Por ejemplo, pueden tomarse datos de un determinado rasgo psicológico antes y después de una intervención. Un segundo diseño que da origen a un contraste de medias relacionadas es el *diseño de sujetos equiparados*, en el cual tenemos dos grupos de diferentes sujetos pero que se han emparejado utilizando una variable de control.

Para realizar un contraste sobre dos medias relacionadas en R , simplemente tenemos que pasarle a `t.test` las dos variables cuyas medias queremos comparar. Continuando con el ejemplo de *terapia.dat*, supongamos que queremos comparar las variables `noche1` y `noche2` (que han sido medidas en la misma muestra). Para obtener una primera impresión de los datos podemos comenzar realizando unos análisis gráficos (diagrama de cajas en Figura 4.1 y diagrama de dispersión en Figura 4.2) con el siguiente código:

```
boxplot(noche1,noche2)
```

```
rm(list = ls())
"noche1" = c(4.0,5.8,4.1,4.3,5.4,5.7,4.3,7.6,5.7,3.8,
             7.5,6.4,4.0,7.7,6.4,5.9,7.2,5.6,5.7,7.4)
"noche2" = c(4.5,6.4,4.9,5.0,6.1,6.3,4.8,7.9,6.0,4.6,
             8.0,7.0,4.5,8.1,6.9,6.6,7.8,6.2,6.1,7.6)
boxplot(noche1,noche2)
```

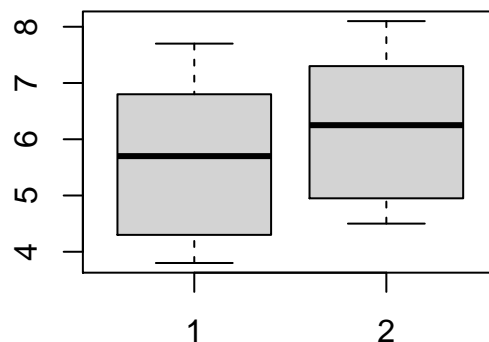


Figura 4.1: Diagrama de cajas para Noche1 y Noche2

A continuación realizamos el contraste mediante `t.test`. En el contraste de dos medias relacionadas esta función recibe como argumentos de entrada las dos variables cuyas medias queremos comparar y el código `paired=T`, que indica que son muestras relacionadas. Esto es:

```
t.test(noche1, noche2, paired=T)
```

Como puede verse, el resultado indica que las dos medias difieren de forma significativa:

```
##
## Paired t-test
##
## data:  noche1 and noche2
## t = -14.769, df = 19, p-value = 7.229e-12
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
## -0.6165269 -0.4634731
## sample estimates:
## mean of the differences
## -0.54
```

4.6. Contraste sobre una correlación

En un diseño intra-sujetos podemos hacer dos contrastes, el de dos medias relacionadas y el *contraste sobre una correlación*. Como en cualquier otro contraste, en el contraste de una correlación conviene comenzar haciendo análisis descriptivos y gráficos que nos den una impresión de los datos antes de hacer la prueba de significación. Como ya vimos en el capítulo 2, el gráfico relevante para una correlación de Pearson es *el diagrama de dispersión*.

Continuando con el ejemplo anterior, para elaborar un gráfico de dispersión para las variables `noche1` y `noche2` con *R* (ver Figura 4.2) puede usarse la siguiente sintaxis (para más detalles puede repasarse el capítulo 2 y el ejemplo de la Figura 2.6):

```
plot(noche1,noche2)
points(noche1,noche2, col = "black", pch = 19)

rm(list = ls())
"noche1" = c(4.0,5.8,4.1,4.3,5.4,5.7,4.3,7.6,5.7,3.8,
            7.5,6.4,4.0,7.7,6.4,5.9,7.2,5.6,5.7,7.4)
"noche2" = c(4.5,6.4,4.9,5.0,6.1,6.3,4.8,7.9,6.0,4.6,
            8.0,7.0,4.5,8.1,6.9,6.6,7.8,6.2,6.1,7.6)
plot(noche1,noche2)
points(noche1,noche2, col = "black", pch = 19)
```

Como también sabemos, para calcular la correlación de Pearson entre `noche1` y `noche2` usaremos la función `cor`:

```
cor(noche1,noche2)
```

```
## [1] 0.9938469
```

La hipótesis nula es la de correlación cero en la población, es decir:

$$\begin{aligned} H_0 : \rho &= 0 \\ H_1 : \rho &\neq 0 \end{aligned}$$

El contraste se realiza mediante la función `cor.test`:

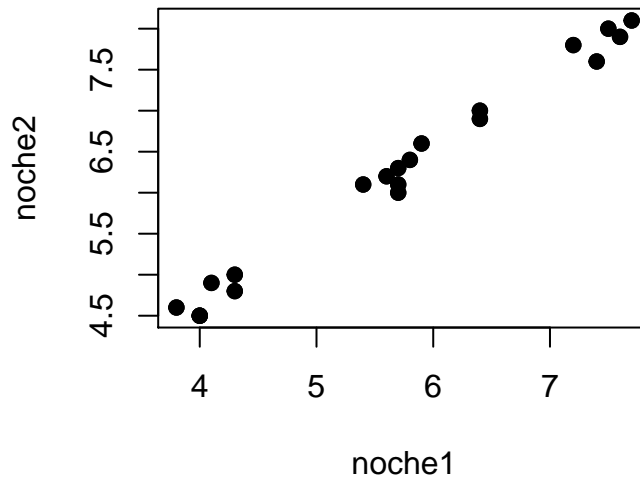


Figura 4.2: Diagrama de dispersión para Noche1 y Noche2

```
cor.test(nochel1,noche2)
```

La salida de resultados de `cor.test` muestra el estadístico de contraste ($t = 38.068$), el nivel crítico ($p\text{-value} < 2.2e-16$, que en este ejemplo lleva a rechazar la hipótesis nula de independencia lineal), y el estimador por intervalos para la correlación (0.9841558-0.9976176). También se muestra el valor de la correlación de Pearson ($cor = 0.9938469$), que es estadísticamente significativa:

```
##
## Pearson's product-moment correlation
##
## data:  noche1 and noche2
## t = 38.068, df = 18, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9841558 0.9976176
## sample estimates:
##      cor
## 0.9938469
```

De forma similar a los capítulos anteriores, se ha elaborado **un resumen de los códigos de R para llevar a cabo contrastes sobre una o dos medias, y puede consultarse el anexo 6.1.**

4.7. Ejercicios propuestos

1. Los resultados obtenidos por un gabinete psicológico durante los últimos 15 años indican que los pacientes con insomnio duermen un promedio de 7 horas durante la tercera noche. Contraste dicha afirmación utilizando $\alpha = 0,05$.
2. Se desea saber si los sujetos que han recibido la terapia 2 duermen durante la segunda noche el mismo número de horas que los que no la han recibido. Para responder utilice $\alpha = 0,01$.
3. Se desea saber si los sujetos duermen más horas durante la tercera noche que durante la primera. Para responder utilice $\alpha = 0,05$.
4. Contraste la hipótesis de que la media en horas dormidas durante la tercera noche es de 8 horas ($\alpha = 0,05$).
5. En una facultad se quiere saber si existen diferencias entre las calificaciones obtenidas en la asignatura *Análisis de datos II* entre los estudiantes matriculados por la mañana y por la tarde. Para ello, se toma una muestra aleatoria de seis alumnos de cada turno. Obtenga una conclusión con $\alpha = 0,01$.

<i>Mañana :</i>	5	7	6	3	1	9
<i>Tarde :</i>	0	2	7	6	5	2

6. Una escuela de secundaria ha contratado a un psicólogo clínico para ingeniar una terapia que sirva para reducir el nivel de ansiedad de sus alumnos ante los exámenes. Para ello selecciona aleatoriamente a 8 alumnos y les mide su nivel de ansiedad en los exámenes de febrero con una escala de 1 a 10 puntos. A continuación les aplica la terapia y vuelve a medir su nivel de ansiedad en los exámenes de Junio. Los resultados aparecen en la tabla inferior. Según esto, ¿Qué concluirá el psicólogo sobre la eficacia de su terapia con $\alpha = 0,01$?

<i>Antes :</i>	10	8	9	7	8	5	9	7
<i>Después :</i>	6	5	7	6	4	5	4	6

Nota: los ejercicios 1 a 4 se refieren al archivo `terapia.dat` (véase anexo 5). Los ejercicios 5 y 6 requieren introducir los datos en *R* según se vio en el capítulo 1. Para responder a estos ejercicios quizá te ayude consultar la tabla del anexo 6.1, que incluye un resumen de los comandos que se han visto en este capítulo para llevar a cabo contrastes de medias con una y dos variables.

5 Contrastes sobre proporciones

5.1. Contraste sobre una proporción

El contraste de una proporción tiene dos versiones, *la prueba binomial*, que suele realizarse con muestras pequeñas, hasta de $n = 25$, y *la aproximación normal a la binomial*, que se usa con muestras mayores ($n > 25$), y simplifica los cálculos.

En este capítulo continuaremos con el ejemplo del archivo *terapia.dat* (ver anexo 5). Vamos a contrastar la hipótesis nula de que la proporción de personas que requieren la **terapia 2** es 0,60, es decir $H_0 : \pi = 0,60$, y para ello utilizaremos la distribución binomial y la normal. Al ser una muestra pequeña ($n = 20$), lo adecuado en este ejemplo sería utilizar únicamente la prueba binomial. Sin embargo, mostraremos también el resultado de la aproximación normal para ilustrar cómo se hace e interpreta en *R*.

5.1.1. Prueba binomial

Recordemos que para realizar el contraste con la distribución binomial necesitamos tres datos: el número de éxitos en la muestra (x), el tamaño muestral (n) y el valor de π a contrastar en H_0 (esto es, π_0). Para realizar la prueba *binomial* utilizamos la función `binom.test`, que recibe como argumentos de entrada estos tres datos. En el siguiente ejemplo, el número de éxitos lo hemos calculado mediante `sum(terapia2)`, dado que la variable `terapia2` contiene ceros y unos, y su suma es igual al número de éxitos; el tamaño muestral lo calculamos mediante `length(terapia2)`, que indica la longitud de este vector de datos; y el tercer valor que pasamos a `binom.test` es el valor de π a contrastar (usaremos $\pi_0 = 0,60$):

```
binom.test(sum(terapia2), length(terapia2), 0.60)
```

Al ejecutar la función `binom.test` se muestra, entre otros resultados, un nivel crítico igual a 1 y el intervalo de confianza para la proporción. Dichos resultados nos llevan a mantener nuestra H_0 .

```
##  
## Exact binomial test
```

```
##
## data:  sum(terapia2) and length(terapia2)
## number of successes = 12, number of trials = 20, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.6
## 95 percent confidence interval:
##  0.3605426 0.8088099
## sample estimates:
## probability of success
##                                0.6
```

La salida de resultados de `binom.test` muestra también el intervalo de confianza para π . En este ejemplo el intervalo es (0.3605, 0.8088). Por tanto, el valor a contrastar en la hipótesis nula (π_0) está incluido dentro del intervalo de confianza, lo que también nos lleva a mantener dicha hipótesis.

5.1.2. Contraste mediante la aproximación normal

La *aproximación normal a la binomial* se realiza mediante la función `prop.test`, la cual recibe los mismos argumentos de entrada que la función `binom.test`:

```
prop.test(sum(terapia2), length(terapia2), 0.6)
```

La salida de resultados de `prop.test` muestra el estadístico X^2 , que no es más que el estadístico Z elevado al cuadrado, y utiliza la distribución chi-cuadrado para obtener el nivel crítico. El intervalo de confianza para la proporción no es idéntico al que ofrece `binom.test` porque `binom.test` lo calcula a partir de una distribución binomial y `prop.test` lo hace a partir de una normal.

```
##
## 1-sample proportions test without continuity correction
##
## data:  sum(terapia2) out of length(terapia2), null probability 0.6
## X-squared = 0, df = 1, p-value = 1
## alternative hypothesis: true p is not equal to 0.6
## 95 percent confidence interval:
##  0.3865815 0.7811935
## sample estimates:
##      p
## 0.6
```

5.2. Contraste sobre dos proporciones independientes

El contraste sobre dos proporciones independientes compara la proporción de desenlaces calculada sobre dos grupos de personas distintas. En nuestro ejemplo, compararemos la proporción de varones y mujeres que duermen más de seis horas la noche 3. Las hipótesis son:

$$H_0 : \pi_1 = \pi_2$$

$$H_1 : \pi_1 \neq \pi_2$$

En primer lugar necesitamos crear una variable dicotómica que tome los valores 0 (dormir 6 o menos horas la noche 3) y 1 (dormir más de 6 horas). A continuación calculamos el número de varones en la muestra, y el número de varones y mujeres que duermen más de 6 horas. Los comandos de *R* que podemos usar son:

```
noche3dicotomica <- as.integer(noches3 > 6) # Creamos la variable dicotomica
n1 <- sum(sexo) # Numero de varones
n0 <- length(sexo)-n1 # El numero de mujeres es n - numero de varones
x0 <- sum(noches3dicotomica[sexo==0]) # Numero de mujeres que duermen mas de 6 horas
x1 <- sum(noches3dicotomica[sexo==1]) # Numero de varones que duermen mas de 6 horas
```

Para realizar el contraste llamamos a `prop.test`, y le pasamos dos vectores llamados `x` y `n`. El vector `x` contiene el número de desenlaces en los dos grupos comparados, y el vector `n` contiene los tamaños de los grupos. También hay que utilizar el argumento `alternative` para especificar si el contraste es unilateral o bilateral. Por tanto, el código sería:

```
prop.test(x = c(x0,x1), n = c(n0,n1), alternative = "two.sided")
```

La salida de resultados muestra el estadístico X^2 , distribuido según chi-cuadrado con 1 grado de libertad, que no es más que Z elevado al cuadrado. También podemos ver el nivel crítico (`p-value = 0.6256`) que indica que debemos mantener H_0 .

```
## Warning in prop.test(x = c(x0, x1), n = c(n0, n1), alternative = "two.sided"):  
## Chi-squared approximation may be incorrect  
  
##  
## 2-sample test for equality of proportions with continuity correction  
##  
## data:  c(x0, x1) out of c(n0, n1)  
## X-squared = 0.2381, df = 1, p-value = 0.6256
```

```
## alternative hypothesis: two.sided
## 95 percent confidence interval:
## -0.6919928  0.2919928
## sample estimates:
## prop 1 prop 2
##      0.6      0.8
```

En este ejemplo, al igual que en el del apartado 5.1 sobre el contraste de una proporción, tenemos muestras demasiado pequeñas para poder utilizar la aproximación normal. Estos contrastes se han incluido únicamente para ilustrar su manejo en *R*, pero en estudios reales es necesario disponer de al menos 25 observaciones por muestra.

5.3. Contraste de bondad de ajuste

El contraste de bondad de ajuste puede entenderse como una generalización del contraste de una proporción para el caso en que la variable tenga más de dos categorías de respuesta. Por ejemplo, en los datos que estamos analizando tenemos la variable `edad` con tres valores que indican si el sujeto es joven, de edad intermedia o adulto. Vamos a contrastar la hipótesis de que la distribución de `edad` es uniforme:

$$H_0 : \pi_1 = \pi_2 = \pi_3$$

$$H_1 : \pi_j \neq \pi_{j'} \quad \text{para } j \neq j'$$

Para realizar este contraste, en primer lugar se utiliza `table` para obtener la distribución de frecuencias de `edad`. Guardamos esta distribución de frecuencias en el objeto `tab`, y a continuación pasamos `tab` a la función `chisq.test` para que calcule el estadístico X^2 de bondad de ajuste. Por defecto, `chisq.test` asume que la hipótesis nula a contrastar es la de *distribución uniforme*. Los comandos son:

```
tab <- table(edad)
chisq.test(tab)
```

El resultado muestra el valor de X^2 y un nivel crítico (`p-value = 0.9512`) que nos lleva a mantener H_0 :

```
##
## Chi-squared test for given probabilities
##
## data:  tab
## X-squared = 0.1, df = 2, p-value = 0.9512
```

También es posible realizar contrastes de bondad de ajuste en los que se especifique cuál es la probabilidad de cada categoría de la variable. Por ejemplo, supongamos que queremos contrastar las hipótesis:

$$H_0 : \pi_1 = 0,35; \pi_2 = 0,25; \pi_3 = 0,40$$

$$H_1 : H_0 \text{ es falsa}$$

Para ello, creamos un vector con las probabilidades teóricas y se lo pasamos a `chisq.test`:

```
prob <- c(0.35, 0.25, 0.40)
chisq.test(tab, p = prob)
```

El nivel crítico resulta ser $p = 0,522$, por lo que mantenemos H_0 y no podemos descartar que esas probabilidades sean correctas en la población.

5.4. Contraste de independencia en tablas de contingencia

El contraste de independencia contrasta la hipótesis de que las dos variables con las que se crea una tabla de contingencia son estadísticamente independientes. Este contraste puede verse como una extensión del contraste de dos proporciones independientes, pero a diferencia de este último, el contraste de independencia puede aplicarse aunque una de las variables analizadas tenga más de dos categorías. Comenzaremos replicando el análisis visto en el apartado 5.2 sobre dos proporciones independientes.

5.4.1. Ejemplo con dos proporciones independientes

Anteriormente hemos realizado el contraste sobre dos proporciones independientes para ver si la proporción de varones y mujeres que duermen más de seis horas la noche 3 es distinta. A nivel descriptivo, como vimos en el capítulo 2, tenemos dos variables dicotómicas, `sexo` y `noche3dicotomica`, cuya tabla de contingencia es:

```
table(sexo,noche3dicotomica)
```

```
##      noche3dicotomica
## sexo 0 1
##    0 4 6
##    1 2 8
```

La tabla de contingencia muestra que hay 6 mujeres y 8 varones que duermen más de seis horas. El contraste de independencia aplicado sobre esta tabla somete a contraste la hipótesis nula de que ambas variables son independientes, y es equivalente al contraste sobre dos proporciones independientes visto en el apartado 5.2. El contraste se realiza con el comando:

```
chisq.test(sexo,noche3dicotomica)
```

El resultado nos muestra un nivel crítico no significativo ($p\text{-value} = 0.6256$). Adicionalmente, *R* nos da un aviso de precaución acerca de que la distribución chi-cuadrado puede no ser correcta, lo que significa que dicho nivel crítico podría no estar bien calculado. Esto se debe a que tenemos una muestra demasiado pequeña como para poder utilizar chi-cuadrado (el estadístico chi-cuadrado no es más que una suma de variables normales elevadas al cuadrado. Por tanto, chi-cuadrado solo puede aplicarse cuando la muestra sea suficientemente grande como para poder utilizar la aproximación normal a la binomial).

```
## Warning in chisq.test(sexo, noche3dicotomica): Chi-squared approximation may be
## incorrect
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  sexo and noche3dicotomica
## X-squared = 0.2381, df = 1, p-value = 0.6256
```

5.4.2. Caso general

Según hemos visto, para realizar el análisis de independencia se utiliza la función `chisq.test`. Este contraste es aplicable con independencia del número de niveles que tenga cada variable. Por ejemplo, supongamos que queremos contrastar la independencia entre las variables `terapia1` y `edad`. Como en cualquier contraste, el primer paso es obtener información descriptiva del análisis que vamos a realizar, en este ejemplo obtenemos la tabla de contingencia entre ambas variables:

```
table(terapia1,edad)
```

A continuación utilizamos `chisq.test` para realizar el contraste de independencia.


```
chisq.test(terapia1,edad)
```

El nivel crítico muestra un resultado no significativo. Sin embargo, al igual que ocurría en el ejemplo anterior, la salida de resultados de la función `chisq.test` advierte que la aproximación chi-cuadrado puede que no sea correcta debido al reducido tamaño muestral.

5.4.3. Prueba exacta de Fisher

La prueba chi-cuadrado de Pearson requiere utilizar muestras de un cierto tamaño (no puede aplicarse si hay más de un 20 % de casillas en la tabla de contingencia con una frecuencia menor de cinco). El procedimiento habitual para contrastar la hipótesis de independencia con muestras pequeñas es la prueba exacta de Fisher, que en *R* se obtiene con el comando `fisher.test`. Por ejemplo, el código para aplicar la prueba exacta de Fisher para contrastar la independencia entre `terapia2` y `terapia3` es:

```
fisher.test(terapia2,terapia3)
```

```
##
## Fisher's Exact Test for Count Data
##
## data:  terapia2 and terapia3
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.163885 11.908012
## sample estimates:
## odds ratio
##  1.376442
```

La salida de resultados de `fisher.test` muestra el nivel crítico (`p-value`) y el intervalo de confianza para la razón de ventajas (`odds ratio`). Ambos resultados nos llevan a mantener la hipótesis nula de independencia.

5.4.4. Índices de asociación entre variables categóricas

Existen diferentes medidas que nos informan de la fuerza de la asociación entre dos variables categóricas. A diferencia de los contrastes de hipótesis, para calcular estas medidas no basta con utilizar las funciones incluidas en el paquete base de *R*, es necesario cargar librerías específicas que veremos a continuación.

5.4.4.1. Índice de riesgo

El índice de riesgo se utiliza ampliamente en epidemiología y en el contexto clínico para evaluar la asociación entre un factor de riesgo y el incremento en un determinado desenlace. A modo de ejemplo, las variables `terapia1` y `terapia2` indican quienes han necesitado terapia para el insomnio y la ansiedad, respectivamente. Supongamos que queremos evaluar el riesgo asociado al factor *ansiedad* con respecto al desenlace *insomnio*. Para calcular el índice de riesgo necesitamos utilizar el paquete `epitools`. Para instalar este paquete puede utilizarse el comando:

```
install.packages("epitools")
```

Una vez instalado el paquete, tenemos que cargarlo en *R* utilizando el comando `library`. A continuación definimos la tabla de contingencia entre `terapia2` y `terapia1` utilizando `table`, y le pasamos dicha tabla a la función `epitab` para que calcule el *índice de riesgo*. El argumento `riskratio` de `epitab` indica que hay que calcular el índice de riesgo:

```
library(epitools)
tab <- table(terapia2,terapia1)
epitab(tab,method="riskratio")
```

El orden en que pasamos las variables a `table` es importante porque en este análisis una de ellas es la variable independiente (la que forma grupos) y la otra es la variable dependiente (aquella cuya proporción de desenlaces se compara entre los grupos). Debemos pasar a `table` primero la variable independiente y después la dependiente.

El resultado de `epitab` muestra la tabla de contingencia, en la cual el valor 1 representa a aquellos que han necesitado terapia. A continuación, la columna `p1` muestra la proporción de los que necesitan la terapia 1 en cada grupo de terapia 2. También tenemos el índice de riesgo en la columna `riskratio`, el intervalo de confianza para el índice de riesgo y el nivel crítico de la prueba exacta de Fisher.

```
##
## Fisher's Exact Test for Count Data
##
## data:  terapia2 and terapia3
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.163885 11.908012
## sample estimates:
## odds ratio
##  1.376442
```

```
## Warning in chisq.test(xx, correct = correction): Chi-squared approximation may
## be incorrect

## $tab
##      terapia2
## terapia1 0    p0 1    p1 riskratio    lower    upper    p.value
##      0 2 0.25 6 0.75 1.0000000      NA      NA      NA
##      1 6 0.50 6 0.50 0.6666667 0.3333984 1.333073 0.3728507
##
## $measure
## [1] "wald"
##
## $conf.level
## [1] 0.95
##
## $pvalue
## [1] "fisher.exact"
```

5.4.4.2. Razón de ventajas

La razón de ventajas, o razón de posibilidades, se utiliza como medida de asociación general entre variables dicotómicas. A diferencia del índice de riesgo, no asume una distinción entre variable independiente y variable dependiente, por lo que el orden en que se pasen las variables a la función `table` no tiene relevancia.

La razón de ventajas la calculamos con la función `oddsratio`, que también está incluida en la librería `epitools`. Continuando con el ejemplo del apartado 5.4.4.1, la llamada a la función es:

```
oddsratio(tab,method="riskratio")
```

La salida de `oddsratio` muestra varios resultados entre los que se incluye la razón de ventajas (`oddsratio`) y los límites de su intervalo de confianza (denominados `lower` y `upper`). También aparece la tabla de contingencia y el nivel crítico (`p.value`) de la prueba exacta de Fisher.

```
##
## Fisher's Exact Test for Count Data
##
## data:  terapia2 and terapia3
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
```

```
## 95 percent confidence interval:
## 0.163885 11.908012
## sample estimates:
## odds ratio
## 1.376442

## Warning in chisq.test(xx, correct = correction): Chi-squared approximation may
## be incorrect

## $data
##      terapia2
## terapia1 0 1 Total
##      0      2 6      8
##      1      6 6     12
##      Total 8 12     20
##
## $measure
##      odds ratio with 95% C.I.
## terapia1 estimate      lower      upper
##      0 1.000000      NA      NA
##      1 0.362467 0.03606234 2.48675
##
## $p.value
##      two-sided
## terapia1 midp.exact fisher.exact chi.square
##      0      NA      NA      NA
##      1 0.3138366 0.3728507 0.2635525
##
## $correction
## [1] FALSE
##
## attr(,"method")
## [1] "median-unbiased estimate & mid-p exact CI"
```

Una característica del *Lenguaje R*, que explica que hoy en día se haya convertido en el estándar para el análisis de datos, es que cualquier usuario puede programar sus propias librerías y ponerlas a disposición de la comunidad científica. Esto ha producido que *R* sea el sistema informático que más análisis distintos permite realizar, dado que existe una enorme comunidad de estadísticos y metodólogos implementando sus análisis en *R* y compartiéndolos con otros investigadores. Por otra parte, también tiene la consecuencia de que existen duplicidades entre librerías, y un mismo estadístico puede estar disponible en dos o más librerías diferentes bajo nombres distintos. Un caso de esto es la razón de ventajas. En este capítulo hemos visto como calcularla mediante la función

`oddsratio` de la librería `epitools`, aunque también está disponible en otras librerías. Por ejemplo la librería `questionr`, para el análisis de datos de encuestas, incorpora la función `odds.ratio`. Para los propósitos de este libro, es suficiente con utilizar la librería `epitools`.

5.4.4.3. Medidas de asociación

Otras medidas de asociación entre variables dicotómicas son el coeficiente de contingencia (C) y el coeficiente V de Cramér. El coeficiente V puede aplicarse a tablas de cualquier tamaño, aunque cuando se aplica a tablas 2×2 suele denominarse coeficiente de correlación *phi* (ϕ). Estos coeficientes son transformaciones del estadístico X^2 y solamente pueden tomar valores positivos. Cuando la hipótesis de independencia se ajusta perfectamente a los datos, X^2 toma el valor 0 y estos coeficientes también. Cuanto mayor sea el valor de los coeficientes, más fuerte es la asociación entre variables. A diferencia del estadístico X^2 , estos coeficientes no pueden ser mayores de 1, por lo que pueden interpretarse en términos relativos, siendo el valor 1 indicador de una asociación perfecta entre variables.

Los coeficientes C y V pueden calcularse en R utilizando la librería `DescTools`, que podemos instalar con el comando:

```
install.packages("DescTools")
```

Supongamos que queremos calcular el coeficiente C entre `edad` y `terapia2`. Podemos hacerlo usando la función `ContCoef`, y con el argumento `correct=TRUE` le estamos diciendo a `DescTools` que calcule un coeficiente de contingencia corregido para que su rango esté comprendido entre 0 y 1, de modo que podamos interpretar cómo de cerca está el resultado de su valor máximo:

```
library(DescTools)
ContCoef(edad,terapia2,correct=TRUE)
```

```
## [1] 0.2517947
```

El coeficiente V lo calculamos con la función `CramerV(edad,terapia2)`. Continuando con el ejemplo:

```
CramerV(edad,terapia2)
```

```
## [1] 0.1809367
```

En caso de tener una tabla de tamaño 2×2 , podemos calcular la correlación ϕ utilizando la función `Phi`. Por ejemplo, la correlación ϕ entre `terapia1` y `terapia2` es:

```
Phi(terapia1,terapia2)
```

```
## [1] 0.25
```

Como el coeficiente V es equivalente a ϕ cuando la tabla es de tamaño 2×2 , habríamos encontrado un resultado igual al anterior utilizando el comando `CramerV(terapia1,terapia2)`.

5.5. Prueba de homogeneidad marginal o de McNemar

La prueba de homogeneidad marginal contrasta la hipótesis de que dos variables dicotómicas medidas sobre los mismos sujetos siguen la misma distribución. En concreto, si π_1 es la probabilidad del desenlace en la primera variable y π_2 es la probabilidad en la segunda variable, la hipótesis nula es $H_0 : \pi_1 = \pi_2$. A este contraste también se le denomina prueba de dos proporciones relacionadas porque la proporciones muestrales de desenlace (P_1 y P_2) se calculan con los mismos sujetos.

Por ejemplo, supongamos que queremos contrastar la hipótesis de que la probabilidad de requerir la terapia 2 y la terapia 3 es la misma. Cómo las variables `terapia2` y `terapia3` han sido medidas sobre los mismos 20 sujetos, se trata de dos proporciones relacionadas.¹ La hipótesis nula es $H_0 : \pi_2 = \pi_3$, referida a la proporción de éxitos en `terapia2` y `terapia3`. El contraste podemos realizarlo con la función `mcnemar.test`, y el código:

```
mcnemar.test(terapia2,terapia3)
```

La salida de resultados muestra el estadístico X^2 y el nivel crítico, que indica que no se encuentran diferencias estadísticamente significativas en las proporciones comparadas.

```
##
## McNemar's Chi-squared test with continuity correction
##
## data:  terapia2 and terapia3
## McNemar's chi-squared = 0, df = 1, p-value = 1
```

¹Según hemos visto en el apartado 5.2, en el contraste de dos proporciones independientes se comparan dos proporciones medidas sobre distintos sujetos. Por ejemplo, la proporción de hombres que requieren la terapia 1 frente a la proporción de mujeres que la requieren.

5.6. Ejercicios propuestos

1. ¿La proporción de sujetos que reciben la terapia contra estados de ansiedad generalizada supera el valor 0,55? ($\alpha = 0,05$).
2. ¿Puede afirmarse que la proporción de sujetos que recibe la terapia para reducir el insomnio difiere de la que la que recibe la terapia para combatir la ansiedad generalizada? ($\alpha = 0,05$).
3. ¿Puede afirmarse que al menos mitad de los sujetos han recibido la terapia contra el insomnio? ($\alpha = 0,01$).
4. Uno de los psicólogos del gabinete que está trabajando con estos pacientes con problemas de insomnio opina que, en la población, el 60 % de los pacientes son varones. Comprobar esta hipótesis con $\alpha = 0,05$.
5. Suponga que disponemos de los siguientes datos acerca de si los sujetos son fumadores o no lo son:

<i>n</i>	1	2	3	3	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
<i>fumar</i>	1	0	1	1	1	0	1	0	1	1	0	0	1	0	0	0	0	1	0	0

Obtenga la tabla de contingencia entre las variables *fumar* y *edad*. ¿Puede afirmarse que existe relación entre ambas variables con $\alpha = 0,01$?

6. ¿Puede afirmarse que el porcentaje de varones que han recibido la terapia contra el insomnio es diferente que el porcentaje de mujeres que la han recibido? ($\alpha = 0,01$).
7. ¿Existe relación entre las variables *fumar* y recibir *terapia1*? ($\alpha = 0,05$).
8. Se está investigando el posible efecto del tabaco sobre el cáncer de pulmón. Para ello, se han recogido los siguientes datos en un hospital. Introduzca los datos en *R* y contraste la hipótesis de ausencia de relación, calcule el índice de riesgo, proporción de cáncer en cada grupo, la *V* de Crámer y el Odds-Ratio.

		Cáncer de pulmón	
		Sí	No
Tabaco	Sí	688	650
	No	21	59

Nota: todos los ejercicios, excepto el último, se refieren al archivo *terapia.dat* (véase anexo 5). **Para responder a estos ejercicios quizá te ayude consultar la tabla del anexo 6.2**, que incluye un resumen de los comandos que se han visto en este capítulo para llevar a cabo contrastes sobre proporciones y tablas de contingencia.

6 Análisis de correlación y regresión lineal

El análisis de correlación y regresión lineal ya lo vimos en el capítulo 2 a nivel descriptivo. En este capítulo lo abordaremos a nivel inferencial. Para ello emplearemos las funciones `cor`, `cor.test` y `lm`.

Las funciones `cor` y `cor.test` ya las hemos visto en el apartado 4.6 referido al contraste sobre una correlación, por lo que aquí no presentan novedad. Por su parte, la función `lm`, que recibe este nombre como acrónimo de *linear model*, permite realizar todo tipo de análisis de regresión lineal, además del de regresión simple, que es el que ya conocemos.

6.1. Análisis de correlación

Como ya sabemos, la función `cor` calcula la *correlación de Pearson* entre dos variables, X e Y , medidas a nivel cuantitativo. Por ejemplo, la correlación entre `noche1` y `noche2` se obtiene mediante:

```
cor(noche1,noche2)
```

En el capítulo 2 ya vimos cómo elaborar diagramas de dispersión para dos variables (ver Figura 2.6). Cuando trabajamos con más de dos variables también puede hacerse un diagrama de dispersión que nos dé una impresión visual rápida de las posibles asociaciones entre pares de variables. Aunque ya vimos cómo se elabora este tipo de gráfico en el capítulo 2 (véase Figura 2.7), también puede utilizarse la función `pairs` para este objetivo. En este caso se comienza creando una matriz de datos con las variables en las columnas. Por ejemplo, podemos utilizar `cbind` para crear una matriz denominada `noche` que contenga en sus columnas las variables `noche1`, `noche2` y `noche3`. A continuación pasamos la matriz `noche` a `pairs` para obtener los diagramas de dispersión que aparecen en la Figura 6.1:

```
noche <- cbind(noche1, noche2, noche3)
pairs(noche)
```

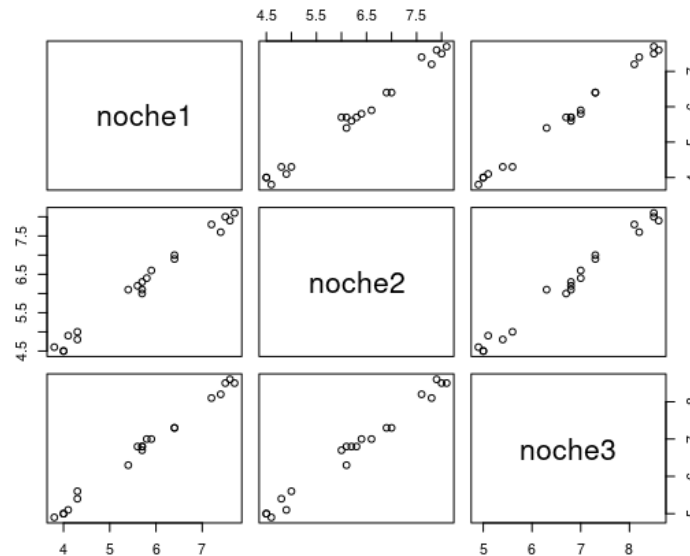


Figura 6.1: Diagramas de dispersión para tres variables

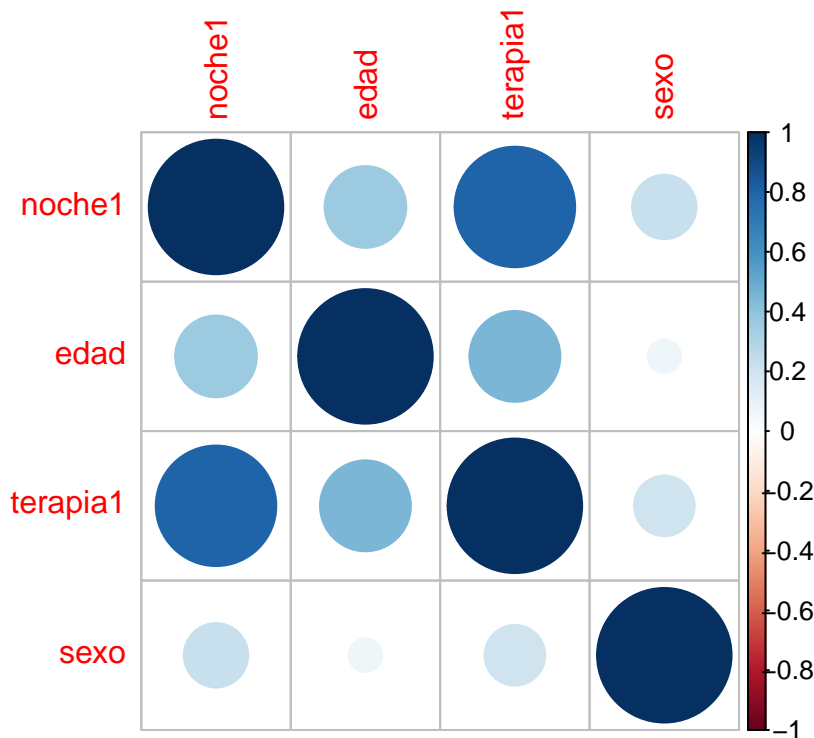
Si quisiéramos calcular la matriz de correlaciones entre las tres variables, simplemente le pasamos la matriz de datos al comando `cor`. Para contrastar la hipótesis nula $H_0 : \rho = 0$ referida a la correlación entre dos variables se utiliza `cor.test`. Hay que tener en cuenta que la función `cor.test` no puede aplicarse a una matriz de correlaciones. Es necesario realizar el contraste individualmente para cada par de variables.

```
cor(noche)
cor.test(noche1, noche2)
cor.test(noche1, noche3)
cor.test(noche2, noche3)
```

R incluye algunas librerías para representar gráficamente las matrices de correlaciones y obtener una impresión visual del patrón de asociación entre las variables. Una de estas librerías es `corrplot`, que podemos instalar con el comando `install.packages("corrplot")`. Supongamos que queremos visualizar la matriz de correlaciones entre las variables `noche1`, `edad`, `terapia1` y `sexo`. El siguiente código crea la matriz de datos con `cbind`, le pasa dicha matriz a `cor` para que calcule la matriz de correlaciones, y le pasa la matriz de correlaciones a `corrplot` para que la represente gráficamente:

```
library(corrplot)
corrplot(cor(cbind(noche1,edad,terapia1,sexo)))
```

Como muestra la Figura 6.2, la salida de resultados de `corrplot` muestra las correlaciones con círculos (el tamaño indica la magnitud de la correlación y el color el signo).

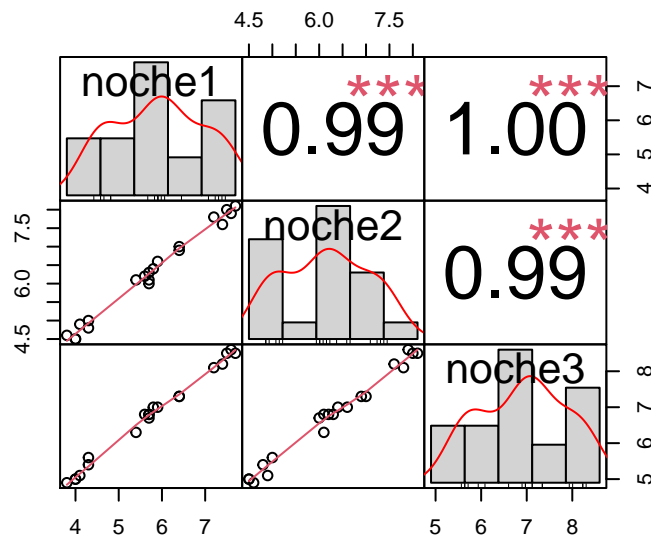
Figura 6.2: Ejemplo de la función `corrplot`

Otra función interesante para la representación gráfica de la correlación es `chart.Correlation`, que está incluida en la librería `PerformanceAnalytics`, y que podemos instalar con el código `install.packages("PerformanceAnalytics")`. Esta función representa los histogramas de frecuencias, los diagramas de dispersión, las correlaciones, y muestra cuáles de ellas son significativas. Por ejemplo, podemos aplicar `chart.Correlation` a las variables `noche1`, `noche2` y `noche3` con el código:

```
library(PerformanceAnalytics)
chart.Correlation(cbind(noche1,noche2,noche3))
```

Como puede verse en la Figura 6.3, las correlaciones aparecen en la parte superior del gráfico, encontrándose los histogramas en la diagonal principal, los gráficos de dispersión debajo de la diagonal a la izquierda, y los valores de la correlación de Pearson encima de la diagonal en la parte derecha (los tres asteriscos indican que todas las correlaciones son significativas al nivel $\alpha = 0,001$).

```
## character(0)
```

Figura 6.3: Ejemplo de la función `chart.Correlation`

6.2. Análisis de regresión

6.2.1. Regresión lineal simple

Como ya vimos en el capítulo 2, el análisis de regresión lineal simple se realiza mediante la función `lm`, a la que hay que pasarle una fórmula que especifique cuál es la variable dependiente y la independiente. Por ejemplo, la fórmula `noche2~noche3` define la regresión lineal de `noche2` sobre `noche3`. El siguiente código realiza esta regresión mediante el comando `lm`, guarda el resultado en el objeto `reg` y utiliza `summary` para mostrar la información contenida en `reg`:

```
reg <- lm(noche2~noche3)
summary(reg)
```

La salida de resultados de `summary(reg)` muestra, en primer lugar, los residuos de la regresión y sus cuartiles (1Q, Median y 3Q). Después aparecen los coeficientes estimados de la regresión, la constante y la pendiente, bajo la etiqueta **Coefficients**, así como la prueba de significación para cada uno de estos coeficientes. Finalmente tenemos el coeficiente de determinación R^2 y el estadístico de contraste F sobre la hipótesis nula de independencia lineal.

##

```
## Call:
## lm(formula = noche2 ~ noche3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.22099 -0.12540 -0.03864  0.12386  0.27018
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.33121    0.20441   -1.62    0.123
## noche3       0.97794    0.02982   32.79 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1629 on 18 degrees of freedom
## Multiple R-squared:  0.9835, Adjusted R-squared:  0.9826
## F-statistic: 1075 on 1 and 18 DF,  p-value: < 2.2e-16
```

El análisis de regresión puede completarse de distintas maneras. Como ya vimos en el capítulo 2, podemos realizar un diagrama de dispersión con la línea de pronósticos superpuesta (véase Figura 2.8). Para realizar el diagrama de dispersión utilizamos el comando `plot` y `abline` para superponer la línea de pronósticos. A la función `abline` hay que pasarle el objeto `reg` en el que se guardó el resultado de la regresión lineal. El resultado se muestra en la Figura 6.4.

```
plot(noche3, noche2)
abline(reg)
```

Podemos estudiar la ecuación de regresión con más detalle utilizando las funciones `fitted` y `resid`. La función `fitted` devuelve el vector de valores pronosticados y `resid` devuelve el vector de errores. Una forma de analizar estos vectores es obtener el diagrama de dispersión mediante el comando:

```
plot(fitted(reg), resid(reg))
```

El modelo de regresión lineal simple asume que la dispersión de los residuos es la misma para cada valor pronosticado (homocedasticidad). Por tanto, el gráfico de la Figura 6.5 permite evaluar el supuesto de homocedasticidad, detectando si existen valores pronosticados para los que la variabilidad sea claramente distinta a los otros. En el ejemplo, no se aprecian desviaciones del supuesto de homocedasticidad.

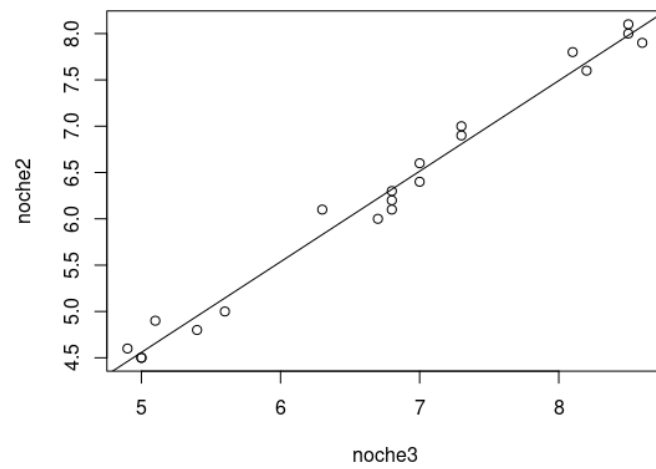


Figura 6.4: Gráfico de dispersión con la línea de pronósticos de la regresión

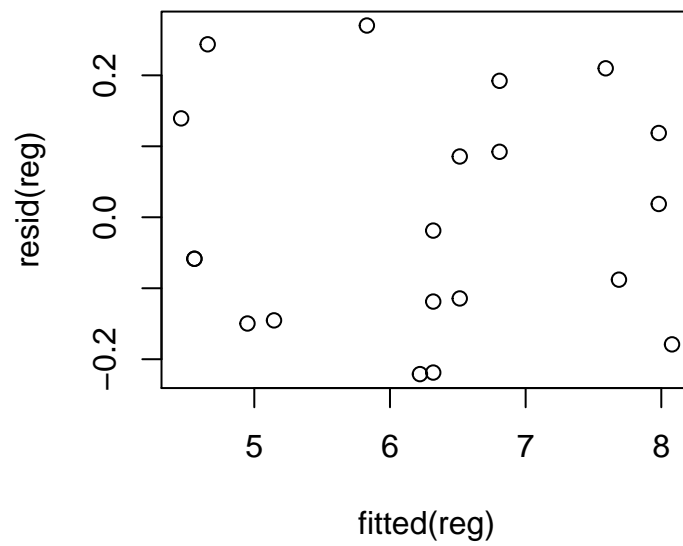


Figura 6.5: Diagrama de dispersión de los residuos y los valores pronosticados

6.2.2. Regresión lineal múltiple

6.2.2.1. Modelo básico

En el **análisis de regresión lineal múltiple** hay más de una variable independiente y las variables independientes se especifican en la fórmula que le pasamos a `lm`. Por ejemplo, la regresión:

$$horas = A + B_1terapia1 + B_2terapia2 + B_3terapia3 + E$$

se especifica mediante el siguiente código:

```
mreg <- lm(horas~terapia1+terapia2+terapia3)
```

En este ejemplo hemos guardado los resultados de la regresión en el objeto `mreg` para poder analizarlos posteriormente.

La siguiente función muestra los coeficientes estimados:

```
summary(mreg)
```

```
##
## Call:
## lm(formula = horas ~ terapia1 + terapia2 + terapia3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3888 -1.8655 -0.6986  1.6203  4.9566
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.3940     1.3383  10.755 9.88e-09 ***
## terapia1       6.1845     1.1216   5.514 4.72e-05 ***
## terapia2     -0.1506     1.1169  -0.135  0.894
## terapia3      1.3103     1.0730   1.221  0.240
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.366 on 16 degrees of freedom
## Multiple R-squared:  0.6733, Adjusted R-squared:  0.612
## F-statistic: 10.99 on 3 and 16 DF, p-value: 0.0003651
```

Los coeficientes muestran que únicamente **terapia1** tiene una pendiente significativa (ver columna Pr(>|t|)), lo que indica que para este ejemplo resulta suficiente utilizar un modelo de regresión lineal simple.

También podemos obtener la tabla de ANOVA de la regresión con el comando:

```
anova(mreg)

## Analysis of Variance Table
##
## Response: horas
##           Df Sum Sq Mean Sq F value    Pr(>F)
## terapia1    1 176.176 176.176 31.4755 3.908e-05 ***
## terapia2    1   0.027   0.027  0.0049  0.9453
## terapia3    1   8.346   8.346  1.4910  0.2397
## Residuals 16  89.556   5.597
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La tabla de ANOVA muestra un estadístico F para contrastar la significación de cada pendiente de la regresión. Únicamente **terapia1** tiene un coeficiente significativo.

6.2.2.2. Regresión por pasos

La *regresión por pasos* es un procedimiento ampliamente utilizado para seleccionar únicamente aquellas variables independientes que resulten útiles para realizar un pronóstico. De este modo, se eliminan de la ecuación de regresión aquellas variables que tengan un coeficiente no significativo (para una explicación sencilla y breve de la regresión por pasos véase el capítulo 3 de Ximénez y San Martín, 2004).

Consideremos el ejemplo de la regresión de **horas** sobre **terapia1**, **terapia2** y **terapia3**. Podría darse el caso de que no todas estas variables independientes sean necesarias para predecir **horas**. Mediante la regresión por pasos se busca un modelo de regresión lo más sencillo posible, en el que únicamente se incluyan aquellas variables predictoras que tengan una relación estadísticamente significativa con la variable dependiente.

Para aplicar la regresión por pasos en R se utiliza el comando **step**, que selecciona el conjunto de variables independientes más adecuado basándose en el estadístico AIC, que mide la distancia del modelo a los datos, de modo que el procedimiento **step** prueba varios modelos hasta encontrar aquel que minimiza el AIC.

Los argumentos de entrada que recibe **step** son el objeto devuelto por la función **lm** y un indicador del método de selección de variables. Estos métodos son:

- El procedimiento *hacia adelante* (**forward**), que parte de un modelo simple y va añadiendo variables independientes que sean significativas.
- El procedimiento *hacia atrás* (**backward**), que parte de un modelo con todas las variables independientes y elimina las que no sean significativas.
- El procedimiento de *pasos sucesivos* (**both**), que combina los anteriores, de modo que en cada paso pueden introducirse y eliminarse variables.

Para especificar el método de selección de variables se utiliza la opción **direction** del comando **step**.

En el ejemplo mencionado, podemos realizar la regresión por pasos mediante el siguiente comando, que devuelve una regresión estimada en la que solamente se retiene **terapia1** como variable independiente, al ser las otras innecesarias:

```
reg <- lm(horas~terapia1+terapia2+terapia3)
step(reg,direction="both")
```

```
## Start:  AIC=37.98
## horas ~ terapia1 + terapia2 + terapia3
##
##           Df Sum of Sq    RSS    AIC
## - terapia2  1      0.102  89.658 36.005
## - terapia3  1      8.346  97.902 37.765
## <none>                        89.556 37.983
## - terapia1  1    170.170 259.726 57.278
##
## Step:  AIC=36.01
## horas ~ terapia1 + terapia3
##
##           Df Sum of Sq    RSS    AIC
## - terapia3  1      8.271  97.929 35.770
## <none>                        89.658 36.005
## - terapia1  1    182.958 272.616 56.247
##
## Step:  AIC=35.77
## horas ~ terapia1
##
##           Df Sum of Sq    RSS    AIC
## <none>                        97.929 35.770
## - terapia1  1    176.18 274.105 54.356
##
```

```
## Call:
## lm(formula = horas ~ terapia1)
##
## Coefficients:
## (Intercept)      terapia1
##      15.100         6.058
```

6.2.3. Regresión lineal múltiple multivariante

Se caracteriza porque tiene más de una variable dependiente y más de una variable independiente. Para realizarla necesitamos crear una matriz que contenga todas las variables dependientes en sus columnas. A continuación calculamos la regresión con `lm`.

Por ejemplo, para estimar simultáneamente las ecuaciones de regresión:

$$\text{noche1} = A + B_{11}\text{terapia1} + B_{12}\text{terapia2} + B_{13}\text{terapia3} + E_1$$

$$\text{noche2} = A + B_{21}\text{terapia1} + B_{22}\text{terapia2} + B_{23}\text{terapia3} + E_2$$

$$\text{noche3} = A + B_{31}\text{terapia1} + B_{32}\text{terapia2} + B_{33}\text{terapia3} + E_3$$

definimos en primer lugar la matriz **noche** que contiene las tres variables dependientes, y a continuación obtenemos la regresión lineal de **noche** sobre las tres variables de **terapia**. El siguiente código realiza este análisis y muestra los resultados, que incluyen una ecuación de regresión lineal para cada variable dependiente:

```
noche <- cbind(noche1,noche2,noche3)
mreg3 <- lm(noche~terapia1+terapia2+terapia3)
summary(mreg3)
```

6.2.4. Regresión no lineal

El *Lenguaje R* también permite estimar modelos de regresión más generales para calcular los pronósticos, como por ejemplo la regresión cuadrática. Veamos un ejemplo. Supongamos que queremos pronosticar el número total de **horas dormidas** a partir de la **edad**, y probamos un modelo de regresión cuadrática para averiguar si tiene mayor capacidad predictiva que el de la regresión lineal simple:

$$\text{horas} = A + B_1\text{edad} + B_2\text{edad}^2 + E$$

Para estimar este modelo programamos la siguiente fórmula, `horas~edad+I(edad^2)`, donde `I(edad^2)` es el código que indica que hay que incluir la **edad** al cuadrado en el modelo. El siguiente código estima esta regresión y guarda los resultados en el objeto que hemos denominado **nlreg**:

```
nlreg <- lm(horas~edad+I(edad^2))
```

Para ver los coeficientes estimados podemos utilizar la función `summary`, que muestra los coeficientes A , B_1 y B_2 , así como su prueba de significación. Como el coeficiente B_2 asociado al término cuadrático ($edad^2$) es significativo, podemos concluir que este modelo predice la variable `horas` mejor que el modelo de regresión lineal simple.

```
summary(nlreg)
```

```
##
## Call:
## lm(formula = horas ~ edad + I(edad^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.4429 -1.9000  0.1667  1.9071  4.1000
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.995       4.207  -0.474  0.641320
## edad           22.671       4.792   4.731  0.000193 ***
## I(edad^2)      -5.276       1.193  -4.422  0.000373 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.542 on 17 degrees of freedom
## Multiple R-squared:  0.5992, Adjusted R-squared:  0.552
## F-statistic: 12.71 on 2 and 17 DF,  p-value: 0.0004219
```

Comparación de modelos

En *R* podemos comparar dos modelos mediante la función `anova`, que muestra un estadístico de contraste F para determinar si la diferencia entre modelos es estadísticamente significativa o no (en el siguiente capítulo se explica en detalle el ANOVA).

Supongamos que vamos a comparar el *modelo de regresión lineal simple* con el *modelo de regresión cuadrática* para ver si el segundo es significativamente mejor que el primero. Como la regresión cuadrática ya está estimada en `nlreg`, solo necesitamos estimar la regresión lineal con `lm` y comparar ambas regresiones mediante `anova`. El código es:

```
reg <- lm(horas~edad)
anova(reg,nlreg)
```

```
## Analysis of Variance Table
##
## Model 1: horas ~ edad
## Model 2: horas ~ edad + I(edad^2)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1      18 236.27
## 2      17 109.87  1      126.4 19.558 0.0003728 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La salida de `anova` muestra un estadístico $F = 19,558$ y un nivel crítico de 0,0003728, que es menor que cualquier valor habitual de α , por lo que podemos rechazar la hipótesis de equivalencia entre modelos y concluir que el modelo de *regresión cuadrática* ofrece mejores pronósticos que el modelo de *regresión lineal*.

Representación gráfica

Finalmente, podemos comparar los dos modelos visualmente realizando un gráfico de dispersión de la nube de datos junto con las líneas de pronósticos. Para ello necesitamos utilizar cuatro funciones:

- La función `seq` permite crear un vector de valores entre 1 y 3 con una separación entre ellos de 0,01. Estos valores representan los valores de la variable independiente con los que se dibujará la curva de pronósticos.
- La función `plot` muestra el diagrama de dispersión.
- La función `lines` dibuja la curva de pronósticos. También se ha utilizado `col` para fijar el color de la línea y `lwd` para especificar su grosor.
- La función `predict` calcula los valores pronosticados dado un modelo de regresión y un vector de valores de la variable independiente.

```
# Crea un vector de valores de la variable independiente
edadVI <- seq(1,3,length=0.01)
# Dibuja el diagrama de dispersión
plot(edad,horas)
# Dibuja los pronosticos de la regresion lineal
lines(edadVI,predict(reg,data.frame(edad=edadVI)),
      col="goldenrod",lwd=1.5)
# Dibuja los pronosticos de la regresion cuadratica
lines(edadVI,predict(nlreg,data.frame(edad=edadVI)),
      col="darkorchid3",lwd=1.5)
```

En el gráfico de la Figura 6.6. puede apreciarse que la nube de datos no se adecúa a la *regresión lineal* porque en el segundo grupo de edad los puntos están por encima de los otros dos. Es por ello que la *regresión cuadrática* realiza mejores pronósticos.

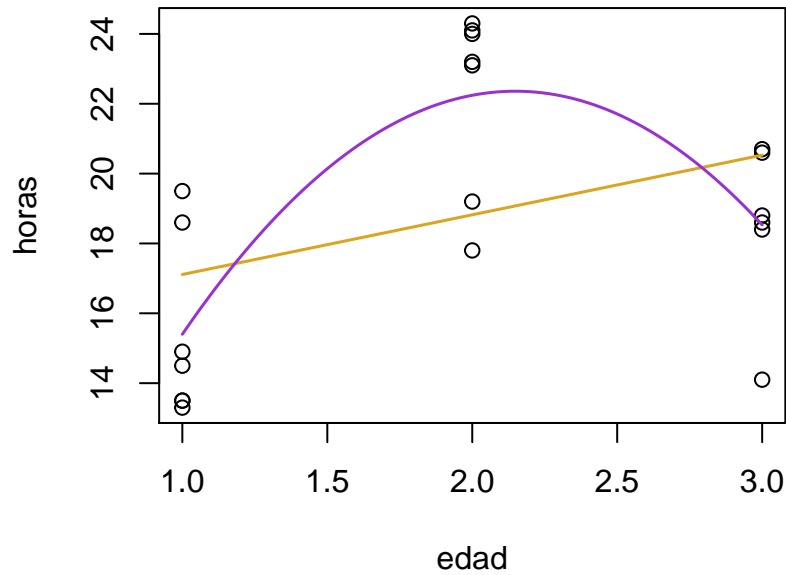


Figura 6.6: Regresión lineal y cuadrática

6.3. Ejercicios propuestos

- Supongamos que se ha medido el número de horas dormidas transcurrido un mes después de la terapia y se obtienen los siguientes resultados:

<i>n</i>	1	2	3	3	5	6	7	8	9	10
<i>noche4</i>	7,53	6,42	9,39	7,18	7,49	9,20	7,24	7,11	8,37	6,29

<i>n</i>	11	12	13	14	15	16	17	18	19	20
<i>noche4</i>	7,51	8,82	5,79	7,16	6,54	7,12	6,10	8,93	9,19	5,99

¿Existe relación lineal entre el promedio de horas dormidas durante las tres primeras noches y las horas dormidas un mes después de la terapia? ($\alpha = 0,01$).

- Calcule la regresión de la variable **noche4** sobre la variable **noche3**.
 - Indique la ecuación de regresión lineal en puntuaciones directas.
 - ¿Cuál es la proporción de varianza en común entre las variables?
 - ¿Existe relación lineal entre las dos variables? ($\alpha = 0,01$).
 - Represente gráficamente la relación entre variables.
 - Averigüe si algún otro modelo (logarítmico, cuadrático, etc.) explicaría mejor la relación entre las dos variables del problema.
 - Indique qué variable (**noche1** o **noche2**) podría añadirse al modelo para mejorar su bondad de ajuste e indique los coeficientes del modelo de regresión múltiple con las dos variables predictoras seleccionadas.

Nota: todos los ejercicios se refieren al archivo *terapia.dat* (véase anexo 5). **Para responder a estos ejercicios quizá te ayude consultar la tabla del anexo 6.3**, que incluye un resumen de los comandos que se han visto en este capítulo para llevar a cabo contrastes de correlación y regresión lineal.

7 Análisis de varianza de un factor y comparaciones múltiples

La técnica del análisis de varianza (en adelante, ANOVA) permite estudiar la posible relación entre una variable dependiente cuantitativa y una variable independiente cualitativa. En concreto, la variable independiente forma J grupos de observaciones, en cada uno de los cuales se calcula la media de la variable dependiente, μ_j . El ANOVA es el contraste de igualdad entre estas J medias, y sus hipótesis son:

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_J$$

$$H_1 : \mu_j \neq \mu_{j'}$$

En este capítulo veremos el ANOVA referido a dos diseños de investigación:

- **Diseño inter-sujetos.** Las personas son diferentes en cada grupo. Por ejemplo, podemos comparar las medias de una determinada variable en personas de diferentes edades: joven, mediana edad y edad avanzada. En un diseño inter-sujetos los diferentes grupos están conformados por distintas personas, por lo que se comparan las medias calculadas con datos de distintos sujetos en cada muestra.
- **Diseño intra-sujetos o de medidas repetidas.** Se trata de un mismo grupo de personas evaluado varias veces. Por ejemplo, en un *diseño longitudinal* en el ámbito de la psicología clínica, pueden tomarse datos de un determinado rasgo psicológico antes, durante y después de una terapia. En otro ejemplo, pueden tomarse datos de un mismo grupo de estudiantes en diferentes asignaturas escolares, lo que se denomina *diseño transversal*, y se caracteriza porque los datos han sido recogidos en un mismo momento temporal. En ambos ejemplos, las diferentes medias que se están comparando se refieren a un mismo grupo de personas sobre el que se han realizado diferentes mediciones.

7.1. Análisis de varianza de un factor

Para realizar un ANOVA es necesario indicar a *R* que *la variable independiente es cualitativa* o nominal, pues de lo contrario interpretará sus valores como cantidades y llevará a cabo un *análisis de regresión lineal*.

Podemos crear una variable cualitativa mediante el comando **factor**. Por ejemplo, en este apartado vamos a utilizar la variable **edad** del fichero *terapia.dat*, que toma los valores 1, 2 y 3, como variable independiente. Estos valores se utilizan en el análisis de varianza como meras etiquetas para indicar a qué grupo pertenece cada individuo.

Para realizar el análisis crearemos una nueva variable, llamada **fedad**, que contendrá los mismos datos que **edad**. La diferencia entre ambas es que le diremos a *R* que **fedad** es nominal y por tanto que no utilice sus valores como si fueran números sino como meras etiquetas. El código *R* que crea la variable **fedad** a partir de **edad** es el siguiente:

```
fedad <- factor(edad)
```

La variable dependiente del análisis es **horas**, que indica el número total de horas dormidas durante tres noches (para más detalles sobre estas variables véase anexo 5).

7.1.1. Análisis descriptivos

Como ya sabemos, antes de realizar un contraste de hipótesis, ya sea un ANOVA o cualquier otro, es muy recomendable explorar los datos mediante análisis descriptivos y gráficos, y observar qué aspecto toman nuestros datos.

En *R* tenemos disponible el comando **aggregate** para calcular estadísticos descriptivos por grupos de observaciones. A este comando tenemos que indicarle cuál es la variable dependiente, la lista de variables independientes y qué estadístico calcular.

En nuestro ejemplo, podemos utilizar la función **aggregate** del siguiente modo para obtener las medias (**mean**) y desviaciones típicas (**sd**) de la variable **horas** en cada grupo definido en **fedad**:

```
aggregate(horas, by=list(fedad), mean)
aggregate(horas, by=list(fedad), sd)
```

El argumento **by=list(fedad)** se utiliza para indicar cuál es la lista de variables independientes, ya que podría haber más de una. El resultado de **aggregate(horas, by=list(fedad), mean)** es:

```
##   Group.1      x
## 1      1 15.40000
## 2      2 22.24286
## 3      3 18.53333
```

Por su parte, la función **aggregate(horas, by=list(fedad), sd)** muestra la desviación típica de cada grupo:


```
##   Group.1      x
## 1      1 2.573584
## 2      2 2.627329
## 3      3 2.396386
```

También podemos hacer un *gráfico de cajas y bigotes* con la función `boxplot`, a la cual hay que indicarle mediante una fórmula cuáles son la variable dependiente y la variable independiente del análisis (esto ya lo vimos en capítulo 2, ver Figura 2.5).

Para realizar el análisis de la variable `horas` por los niveles de `edad`, le pasamos la fórmula `horas~edad` a la función `boxplot`, lo que significa que `horas` es la variable dependiente y `edad` la independiente:

```
boxplot(horas~edad)
```

El resultado aparece en la Figura 7.1.

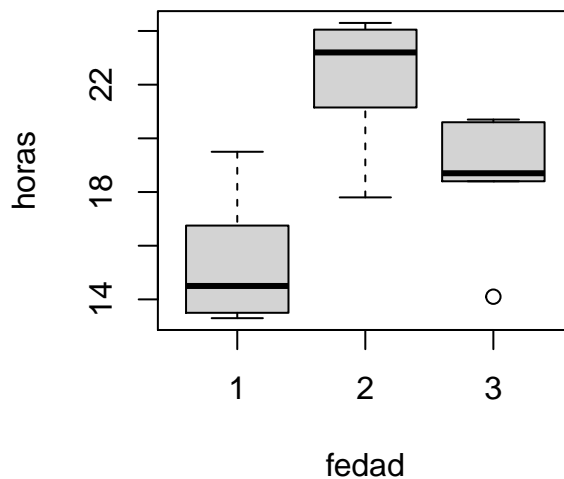


Figura 7.1: Gráfico de cajas y bigotes de horas por edad

7.1.2. Tabla de ANOVA

El ANOVA se realiza mediante la función `aov`, a la cual hay que indicarle cuál es la variable dependiente e independiente utilizando la misma fórmula que hemos visto para `boxplot`. El resultado del análisis lo hemos guardado en el objeto `fit` con el propósito de utilizarlo para posteriores análisis, por ejemplo con las funciones `summary` o `plot`:

```
fit <- aov(horas~edad)
summary(fit)
coef(fit)
plot(fit)
model.tables(fit,"means")
```

A continuación podemos ver el resultado del comando `summary(fit)`, que muestra la tabla de ANOVA con los grados de libertad (Df), sumas de cuadrados (Sum Sq), medias cuadráticas (Mean Sq), el estadístico de contraste (F value) y el nivel crítico ($\Pr(>F)$). En el ejemplo vemos que el nivel crítico es menor que α por lo que rechazamos la H_0 .

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## edad           2   164.2    82.12   12.71 0.000422 ***
## Residuals     17   109.9     6.46
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

7.1.3. Medidas del tamaño del efecto

Cuando se rechaza la H_0 hay que analizar el **tamaño del efecto** encontrado. Para ello pueden usarse los estadísticos eta-cuadrado (η^2), epsilon-cuadrado (ϵ^2) y omega-cuadrado (ω^2). Para calcularlos en *R* tenemos que cargar la librería `effectsize` y usar las funciones `eta_squared`, `epsilon_squared` y `omega_squared`, poniendo en el argumento el objeto donde hayamos guardado los resultados del ANOVA que se obtuvieron mediante la función `aov` (en nuestro ejemplo, dicho objeto se llamaba `fit`). Continuando con el ejemplo anterior, el código para calcular las medidas del tamaño del efecto es:

```
library(effectsize)
eta_squared(fit)
epsilon_squared(fit)
omega_squared(fit)
```

La función `eta_squared` muestra una proporción de varianza explicada de 0,60, así como su intervalo de confianza al 95 %:

```
## Registered S3 method overwritten by 'parameters':
##   method      from
## predict.kmeans statip

##
## Attaching package: 'effectsize'
```

```
## The following objects are masked from 'package:epitools':
##
##     oddsratio, riskratio

## For one-way between subjects designs, partial eta squared is equivalent to eta squared
## Returning eta squared.

## # Effect Size for ANOVA
##
## Parameter | Eta2 |          95% CI
## -----
## fedad     | 0.60 | [0.29, 1.00]
##
## - One-sided CIs: upper bound fixed at (1).
```

Los estadísticos ϵ^2 y ω^2 son correcciones de η^2 que intentan evitar el problema de la sobrestimación de la proporción de varianza explicada asociado a este último estadístico. El resultado para ϵ^2 es 0,55:

```
## For one-way between subjects designs, partial epsilon squared is equivalent to epsilon squared
## Returning epsilon squared.

## # Effect Size for ANOVA
##
## Parameter | Epsilon2 |          95% CI
## -----
## fedad     | 0.55 | [0.23, 1.00]
##
## - One-sided CIs: upper bound fixed at (1).
```

Por último, como puede verse más abajo, el estadístico ω^2 toma el valor 0,54. Para interpretar los valores de ω^2 , Cohen (1988) propuso que un valor de ω^2 entre 0,01 y 0,06 se considera un efecto pequeño; entre 0,06 y 0,14 se considera un efecto medio; y por encima de 0,14, como es el caso en este ejemplo, se considera que el efecto es grande.

```
## For one-way between subjects designs, partial omega squared is equivalent to omega squared
## Returning omega squared.

## # Effect Size for ANOVA
##
## Parameter | Omega2 |          95% CI
## -----
## fedad     | 0.54 | [0.22, 1.00]
##
## - One-sided CIs: upper bound fixed at (1).
```

7.2. Comparaciones múltiples

El rechazo de la hipótesis nula del ANOVA lleva a concluir que existen diferencias entre las medias poblacionales de los grupos comparados; sin embargo, no proporciona información acerca de qué medias difieren entre sí ni del patrón que siguen los resultados. Para obtener esta información necesitamos aplicar los procedimientos de comparaciones múltiples, dentro de los cuales están las comparaciones a-posteriori y a-priori.

Se denominan *comparaciones a-posteriori* a las que se realizan sin tener una expectativa previa acerca de qué medias pueden diferir de otras. Son procedimientos exploratorios que no se basan en una hipótesis previa acerca de qué medias difieren entre sí.

Por su parte, las *comparaciones a-priori* tienen una naturaleza confirmatoria. Parten de una idea previa acerca de qué medias podrían diferir de otras, o de cuál puede ser el patrón de relación entre las medias. Esta idea inicial se traduce en una hipótesis estadística que ha de someterse a contraste.

7.2.1. Comparaciones a-posteriori

La prueba a-posteriori más sencilla es la *prueba de Tukey*, que consiste en comparar “dos a dos” todas las medias de nuestro estudio para buscar aquellas cuyas diferencias sean estadísticamente significativas.

La prueba de Tukey puede aplicarse en *R* mediante la función `TukeyHSD`, a la cual le pasamos como argumento el objeto devuelto por la función `aov`. En nuestro ejemplo, dicho objeto se llamaba `fit`, por lo que haremos la prueba de Tukey con el código:

```
TukeyHSD(fit)
```

La salida de resultados de la prueba de Tukey muestra la diferencia entre cada par de medias (`diff`), el límite inferior y superior del intervalo de confianza para la diferencia (`lwr` y `upr`) y el nivel crítico (`p adj`). El resultado permite concluir, utilizando $\alpha = 0,01$, que hay una diferencia significativa entre las medias de los grupos 1 y 2.

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = horas ~ fedad)
##
## $fedad
##          diff          lwr          upr          p adj
## 2-1  6.842857  3.356837 10.3288768 0.0002849
## 3-1  3.133333 -0.495031  6.7616976 0.0970953
## 3-2 -3.709524 -7.337888 -0.0811595 0.0446719
```

Una segunda forma de hacer comparaciones por pares de medias es mediante el estadístico de contraste T . Para ello utilizamos la función `pairwise.t.test`, a la cual hay que indicarle cuál es la variable dependiente e independiente. En nuestro ejemplo, la conclusión es la misma que con la prueba de Tukey:

```
pairwise.t.test(horas,fedad)

##
## Pairwise comparisons using t tests with pooled SD
##
## data:  horas and fedad
##
##      1      2
## 2 0.00031 -
## 3 0.04068 0.03564
##
## P value adjustment method: holm
```

La diferencia entre el procedimiento `paired.t.test` (comparaciones múltiples entre pares de medias) y `t.test` (prueba T sobre diferencia de medias) está en el cálculo del nivel crítico. Cuando se realizan comparaciones múltiples entre pares de medias se incrementa la probabilidad de cometer *errores de tipo I*, que ocurrirían cuando por causa del error muestral, en alguna de las comparaciones se rechaza la hipótesis nula aun siendo verdadera. En concreto, si realizamos un contraste de medias y H_0 es verdadera, la probabilidad de no cometer un *error de tipo I* es $1 - \alpha$ y la probabilidad de cometerlo es α . Si realizamos tres contrastes y H_0 fuese verdadera en todos ellos, la probabilidad de que en ningún contraste cometamos un *error de tipo I* es $(1 - \alpha)^3$, y la probabilidad de cometerlo en al menos un contraste es $1 - (1 - \alpha)^3$. Por concretar estos cálculos, con tres contrastes y $\alpha = 0,05$, tenemos una probabilidad de 0,14 de cometer un *error de tipo I* en al menos uno de ellos si todas las hipótesis nulas son verdaderas.

Existen métodos para ajustar el nivel crítico con la intención de que la probabilidad de cometer al menos un *error de tipo I* no sea superior al nivel α del contraste. El procedimiento `paired.t.test` incorpora varias de estas correcciones, y en la salida de resultados del ejemplo vemos que el nivel crítico se ha ajustado utilizando el método de Holm.

7.2.2. Comparaciones planeadas o a-priori

Las *comparaciones planeadas* se basan en definir una combinación lineal de medias que represente la comparación que queremos realizar, y contrastar la hipótesis de que dicha combinación lineal es 0 en la población. De modo general, las hipótesis para $J = 3$ son:

$$H_0 : c_1\mu_1 + c_2\mu_2 + c_3\mu_3 = 0$$

$$H_1 : c_1\mu_1 + c_2\mu_2 + c_3\mu_3 \neq 0$$

Donde c_1 , c_2 y c_3 son coeficientes que deben cumplir la propiedad de suma cero:

$$c_1 + c_2 + c_3 = 0$$

Comparaciones de tendencia

Estudian cuál es la forma de la relación entre una variable independiente cuantitativa y la variable dependiente. Con $J = 3$ niveles se puede contrastar la tendencia de tipo *lineal* y *cuadrática*, con $J = 4$ las tendencias *lineal*, *cuadrática* y *cúbica*, etc.

Cada tipo de relación lleva asociada una hipótesis nula expresada como combinación lineal de medias. Continuando con nuestro ejemplo, la variable **edad** forma tres grupos por lo que es posible estudiar $J - 1 = 2$ tipos de relación, la *lineal* y la *cuadrática*. La Figura 7.1 sugiere que la relación de **horas** con **edad** es *cuadrática*, aunque vamos hacer los contrastes oportunos. Las hipótesis nulas para estudiar estos tipos de relación son:

$$H_{0(\text{lineal})} : -\mu_1 + \mu_3 = 0$$

$$H_{0(\text{cuadrática})} : \mu_1 - 2\mu_2 + \mu_3 \neq 0$$

En forma matricial, los coeficientes de las dos hipótesis nulas aparecen en la matriz **C**, donde la primera columna se refiere a la relación *lineal* y la segunda a la *cuadrática*, y las filas se refieren a las tres medias comparadas:

$$\mathbf{C} = \begin{pmatrix} -1 & 1 \\ 0 & -2 \\ 1 & 1 \end{pmatrix}$$

La matriz **C** de coeficientes se obtiene automáticamente en *R* con la función `contr.poly(3)`, donde 3 es el número de grupos de la variable independiente. El código *R* para contrastar estos dos tipos de relación es:

```
contrasts(fedad) <- contr.poly(3) # Crear matriz C y asignarla a edad
fit <- aov(horas~fedad) # Realizar el ANOVA
fit_tendencia <- lm(fit) # Realizar las comparaciones de tendencia
summary(fit_tendencia) # Ver resultados de comparaciones de tendencia
```

El análisis de resultados muestra tres parámetros estimados: el intercepto (media total), el parámetro L y el parámetro Q. Los nombres L y Q vienen de **linear** y **quadratic** (*lineal* y *cuadrático*, en inglés). Estos parámetros toman el valor cero cuando dicha relación no se da en la población. Por tanto, se contrasta la hipótesis de valor cero poblacional, y un nivel crítico significativo lleva a concluir que el parámetro es distinto de cero y que la relación está presente en los datos.

Al realizar las comparaciones de tendencia, R ha llevado a cabo un contraste de significación para cada una de las tres combinaciones lineales de medias:

$$\begin{aligned}\text{Intercept} &= \frac{\mu_1}{3} + \frac{\mu_2}{3} + \frac{\mu_3}{3} \\ \text{fedad.L} &= -\frac{\mu_1}{\sqrt{2}} + \frac{\mu_3}{\sqrt{2}} \\ \text{fedad.Q} &= \frac{\mu_1}{\sqrt{6}} - \frac{2\mu_3}{\sqrt{6}} + \frac{\mu_3}{\sqrt{6}}\end{aligned}$$

La función `contr.poly(3)` utiliza los coeficientes: $(-1/\sqrt{2}, 0, 1/\sqrt{2})$ para la relación lineal, y $(1/\sqrt{6}, -2/\sqrt{6}, 1/\sqrt{6})$ para la relación cuadrática. Estos coeficientes son los mismos que los del planteamiento de las hipótesis, pero se han dividido por el denominador adecuado para conseguir que la suma de los coeficientes al cuadrado sea 1; es decir, $\sum_{j=1}^J c_j^2 = 1$. Esta modificación no afecta al sentido de las hipótesis a contrastar.

La salida de resultados obtenida con `summary(fit_tendencia)` muestra el valor estimado de cada combinación lineal y su prueba de significación con el estadístico T :

```
##
## Call:
## lm(formula = fit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.4429 -1.9000  0.1667  1.9071  4.1000
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.7254      0.5700  32.854 < 2e-16 ***
## fedad.L       2.2156      1.0001   2.215 0.040676 *
## fedad.Q      -4.3080      0.9741  -4.422 0.000373 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.542 on 17 degrees of freedom
## Multiple R-squared:  0.5992, Adjusted R-squared:  0.552
## F-statistic: 12.71 on 2 and 17 DF, p-value: 0.0004219
```

En el ejemplo, mantenemos la hipótesis nula de la relación *lineal* (`fedad.L`) y rechazamos la hipótesis de la *cuadrática* (`fedad.Q`) utilizando $\alpha = 0,01$. Por tanto, concluimos que la relación es cuadrática. La hipótesis nula referida al intercepto también se rechaza, lo que simplemente quiere decir que la media total es distinta de cero.

Comparaciones planeadas ortogonales

Las comparaciones planeadas ortogonales se realizan de un modo similar a las de tendencia, pero la matriz **C** no se obtiene automáticamente mediante `contr.poly` sino que debemos fijarla según las comparaciones de medias que nos interese contrastar. El número de comparaciones ortogonales que pueden realizarse con J grupos es $J - 1$. Por ejemplo, si $J = 3$ podemos hacer las dos siguientes comparaciones ortogonales:

$$H_{0(1)} : c_{11}\mu_1 + c_{12}\mu_2 + c_{13}\mu_3 = 0$$

$$H_{0(2)} : c_{21}\mu_1 + c_{22}\mu_2 + c_{23}\mu_3 \neq 0$$

Además de cumplir la propiedad de suma cero, $\sum_{j=1}^J c_{1j} = 0$ y $\sum_{j=1}^J c_{2j} = 0$, debe cumplirse la propiedad de vectores de coeficientes ortogonales, $\sum_{j=1}^J c_{1j}c_{2j} = 0$.

Continuando con el ejemplo, vamos a contrastar las dos siguientes hipótesis:

$$H_{0(1)} : -\mu_1 - \mu_2 + 2\mu_3 = 0$$

$$H_{0(2)} : \mu_1 - \mu_2 = 0$$

La primera hipótesis nula compara la media μ_3 con las medias μ_1 y μ_2 tomadas juntas. La segunda hipótesis compara las medias μ_1 y μ_2 . En *R* tenemos que crear la matriz **C** con los coeficientes de estas hipótesis a contrastar, donde cada hipótesis se sitúa en una columna. En el ejemplo, la matriz de coeficientes es:

$$\mathbf{C} = \begin{pmatrix} -1 & 1 \\ -1 & -1 \\ 2 & 0 \end{pmatrix}$$

Podemos utilizar la función `cbind` para crear la matriz **C**. En este ejemplo, la función `cbind` toma dos argumentos como entrada, que son los vectores de coeficientes de las hipótesis nulas. El vector de coeficientes de cada hipótesis lo creamos con la función `c()` y después utilizamos `cbind()` para unir los dos vectores en una única matriz. Es decir:

```
C <- cbind(c(-1, -1, 2), c(1, -1, 0))
```

A continuación, tenemos que aplicar la matriz de coeficientes al factor `edad`. Para ello empleamos la función `contrasts` del siguiente modo:

```
contrasts(edad) <- C
```

Finalmente, empleamos `aov` para realizar el ANOVA y guardamos los resultados en el objeto denominado, de forma arbitraria, `fit`. Después pasamos `fit` a la función `lm` para que realice las comparaciones múltiples entre medias:


```
fit <- aov(horas~fedad)
fit_planned <- lm(fit)
summary(fit_planned)
```

El resultado de `lm` puede visualizarse con `summary`. La salida de resultados muestra la fila **Intercept** y otras dos filas denominadas **fedad1** y **fedad2** que se refieren a cada hipótesis nula a contrastar. Estas filas se han calculado del siguiente modo:

$$\begin{aligned}\text{Intercept} &= \frac{\mu_1}{3} + \frac{\mu_2}{3} + \frac{\mu_3}{3} \\ \text{fedad1} &= -\frac{\mu_1}{6} - \frac{2\mu_3}{6} + \frac{2\mu_3}{6} \\ \text{fedad2} &= \frac{\mu_1}{2} - \frac{\mu_3}{2}\end{aligned}$$

R ha calculado los términos **fedad1** y **fedad2** utilizando los coeficientes indicados en **C**, y después ha dividido estos términos por la suma de sus coeficientes al cuadrado.

```
##
## Call:
## lm(formula = fit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.4429 -1.9000  0.1667  1.9071  4.1000
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.72540    0.56996   32.854 < 2e-16 ***
## fedad1       -0.09603    0.41349   -0.232  0.819121
## fedad2       -3.42143    0.67944   -5.036  0.000102 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.542 on 17 degrees of freedom
## Multiple R-squared:  0.5992, Adjusted R-squared:  0.552
## F-statistic: 12.71 on 2 and 17 DF, p-value: 0.0004219
```

En cada fila de la tabla de resultados tenemos el valor estimado de la combinación lineal de medias (**Estimate**), el error típico (**Std. Error**), el estadístico de contraste (**t value**) y el nivel crítico (**Pr(>t)**). La decisión es mantener $H_{0(1)}$ y rechazar $H_{0(2)}$, por lo que se puede concluir que las medias del primer y segundo grupo difieren significativamente.

7.3. Análisis de varianza con medidas repetidas

El ANOVA de medidas repetidas (ANOVA-MR) se aplica en un diseño intra-sujetos, en el que un mismo grupo de sujetos es evaluado en diferentes ocasiones, por ejemplo en distintos momentos temporales (*diseño longitudinal*) o en distintas variables medidas en un mismo momento (*diseño transversal*).

7.3.1. Organización de los datos

Para realizar un ANOVA-MR con *R* hay que organizar los datos en tres columnas, una para el código de identificación del sujeto, otra para la variable independiente y una tercera columna para la variable dependiente (para una amplia revisión de estos modelos véase Ximénez y San Martín, 2000).

Supongamos que queremos contrastar la hipótesis de que las medias poblacionales de **noche1**, **noche2** y **noche3** son iguales. En el conjunto de datos con que estamos trabajando (archivo *terapia.dat* del anexo 5) estas variables ocupan cada una de ellas una columna, por lo que tenemos el siguiente formato:

<i>n</i>	<i>noche1</i>	<i>noche2</i>	<i>noche3</i>
1	4,0	4,5	5,0
2	5,8	6,4	7,0
3	4,1	4,9	5,1
4	4,3	5,0	5,6
5	5,4	6,1	6,3
6	5,7	6,3	6,8
7	4,3	4,8	5,4
8	7,6	7,9	8,6
9	5,7	6,0	6,7
10	3,8	4,6	4,9
11	7,5	8,0	8,5
12	6,4	7,0	7,3
13	4,0	4,5	5,0
14	7,7	8,1	8,5
15	6,4	6,9	7,3
16	5,9	6,6	7,0
17	7,2	7,8	8,1
18	5,6	6,2	6,8
19	5,7	6,1	6,8
20	7,4	7,6	8,2

Para realizar un ANOVA-MR en *R* es necesario colocar los datos de tal modo que **noche1**, **noche2** y **noche3** estén en una misma columna, a la cual vamos a denominar **nhoras**.

También es necesario tener otra columna con los códigos de identificación del sujeto, que en este ejemplo no es más que el número indicado por la columna **n**. A la variable con los códigos de identificación vamos a llamarle **fsujeto**.

Finalmente, definiremos la variable independiente **fnoche** que indica a qué noche se refiere cada uno de los datos de **nhoras**.

En definitiva, vamos a reestructurar los datos para que tengan el siguiente formato (para ahorrar espacio, solamente se muestran los datos de los cinco primeros sujetos):

<i>fsujeto</i>	<i>fnoche</i>	<i>nhoras</i>
1	1	4,0
2	1	5,8
3	1	4,1
4	1	4,3
5	1	5,4
1	2	4,5
2	2	6,4
3	2	4,9
4	2	5,0
5	2	6,1
1	3	5,0
2	3	7,0
3	3	5,1
4	3	5,6
5	3	6,3

El siguiente código reorganiza nuestros datos para colocarlos del modo requerido. Se han añadido comentarios para aclarar el sentido de cada línea:

```
# Creamos una matriz con los datos de la noche 1. Esta matriz
# contiene tres columnas, el número de sujeto (que está en la
# variable n), una columna de valores 1 que indica que estamos
# utilizando los datos de la primera noche y una tercera columna
# con el número de horas dormidas
dat1 <- cbind(n, 1, noche1)

# Creamos matrices similares para la segunda y la tercera noche
dat2 <- cbind(n, 2, noche2)
dat3 <- cbind(n, 3, noche3)

# Creamos una matriz de datos poniendo dat1, dat2 y dat3 uno encima
# del otro
dat <- rbind(dat1,dat2,dat3)
```

```
# Asignamos los nombres de variables a las tres columnas de la
# matriz dat
colnames(dat) <- c("fsujeto", "fnoche", "nhoras")

# Convertimos dat en un objeto de tipo marco de datos
dat <- as.data.frame(dat)

# Convertimos la variable fsujeto en nominal
dat[, "fsujeto"] <- as.factor(dat[, "fsujeto"])

# Convertimos la variable fnoche en nominal
dat[, "fnoche"] <- as.factor(dat[, "fnoche"])
```

7.3.2. Análisis descriptivos

Una vez organizados los datos del modo apropiado, los análisis descriptivos previos al contraste de hipótesis pueden realizarse con las mismas funciones que vimos en el caso del ANOVA inter-sujetos:

```
aggregate(nhoras~fnoche, mean, data=dat)
boxplot(nhoras~fnoche, mean, data=dat)
```

El resultado de `aggregate` muestra la media de las tres noches:

```
##   fnoche nhoras
## 1      1  5.725
## 2      2  6.265
## 3      3  6.745
```

Con el comando `boxplot` obtenemos el *diagrama de cajas y bigotes* que aparece en la Figura 7.2.

7.3.3. Tabla de ANOVA

El ANOVA-MR se realiza con la función `aov`. Para llamar a esta función se utiliza el siguiente código:

```
fit <- aov(nhoras~fnoche + Error(fsujeto/fnoche), data=dat)
```

Los argumentos que le hemos pasado a `aov` son los siguientes:

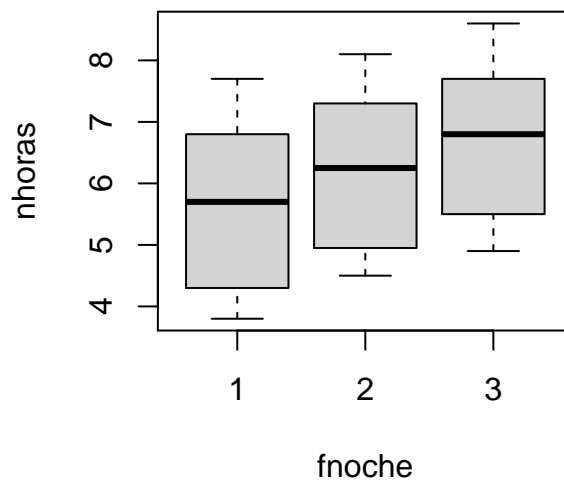


Figura 7.2: Diagrama de cajas y bigotes de las tres noches

- La fórmula `nhoras~fnoche + Error(fsujeto/fnoche)` indica que estamos haciendo un ANOVA-MR y que `nhoras` es la variable dependiente y `fnoche` la independiente.
- Con la expresión `Error(fsujeto/fnoche)` estamos diciendo que existen diferencias aleatorias (*errores*, en terminología estadística) entre los distintos sujetos evaluados una misma noche. En el ANOVA-MR siempre hay que incluir en la ecuación un término de la forma `Error(id/vi)`, donde `id` es la variable con los códigos de identificación del sujeto y `vi` es la variable independiente.
- Con `data=dat` le estamos diciendo a `aov` que las variables mencionadas en la fórmula están almacenadas en el objeto `dat`.

Una vez realizado el ANOVA y guardado el resultado en el objeto `fit`, podemos pasarle este objeto a distintas funciones para que nos muestre un resumen de resultados, los parámetros estimados, y las medias en forma de tabla o gráfico:

```
summary(fit)
coef(fit)
model.tables(fit, "means")
```

La salida de resultados de `summary(fit)` muestra la tabla de ANOVA del factor intra-sujetos, en la que podemos ver un nivel crítico muy próximo a cero que nos lleva a rechazar la hipótesis nula de igualdad de medias.

7 Análisis de varianza de un factor y comparaciones múltiples

```
##
## Error: fsujeto
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 19  91.37    4.809
##
## Error: fsujeto:fnoche
##           Df Sum Sq Mean Sq F value Pr(>F)
## fnoche      2 10.416    5.208   445.7 <2e-16 ***
## Residuals 38  0.444    0.012
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La función `coef(fit)` muestra los parámetros estimados del modelo estadístico subyacente al ANOVA-MR.

```
## (Intercept) :
## (Intercept)
##          6.245
##
## fsujeto :
## numeric(0)
##
## fsujeto:fnoche :
## fnoche2 fnoche3
##      0.54      1.02
```

La función `model.tables(fit)` muestra la tabla de efectos, que son las diferencias entre la media de cada grupo y la media total de toda la muestra.

```
## Tables of effects
##
## fnoche
## fnoche
##      1      2      3
## -0.52  0.02  0.50
```

7.4. Ejercicios propuestos

1. Un psicólogo desea comprobar si existen diferencias en el número medio de horas dormidas por los pacientes con insomnio durante la primera noche en los tres grupos de edad considerados en el archivo *terapia.dat* del anexo 5. A continuación, conteste a las siguientes cuestiones:
 - a) Seleccione el modelo de ANOVA más apropiado, plantee la hipótesis pertinente y tome una decisión con $\alpha = 0,05$
 - b) ¿Existe relación entre la edad y el número de horas dormidas la primera noche? En caso afirmativo, ¿Cuál es la tendencia de dicha relación?
 - c) ¿A qué edad duermen menos horas los pacientes con insomnio? ($\alpha = 0,05$).
 - d) Compruebe si el número de horas de sueño que duermen los sujetos de 20 a 25 años difiere del nivel de los restantes considerados juntos ($\alpha = 0,01$).

2. Una psicóloga clínica desea evaluar la eficacia de un fármaco para reducir la ansiedad. Para ello selecciona al azar 15 pacientes de su consulta que sufren este problema y forma aleatoriamente tres grupos del mismo tamaño. A cada grupo le administra aleatoriamente una dosis del fármaco (10 mg., 20 mg. y 30 mg.). Al cabo de un tiempo mide su nivel de ansiedad. Los resultados obtenidos se muestran en la siguiente tabla:

10mg.	7	8	8	9	8
20mg.	4	4	5	6	6
30mg.	2	3	2	2	1

- a) Introduzca los datos en *R* utilizando un marco de datos.
- b) Seleccione el modelo de ANOVA más apropiado, plantee la hipótesis nula y tome una decisión con $\alpha = 0,05$.
- c) ¿Existe relación entre la dosis del fármaco y el nivel de ansiedad?
- d) En caso afirmativo, interprete gráfica y estadísticamente el tipo de relación.
- e) Cuantifique el tamaño de dicha relación.
- f) ¿Entre qué pares de dosis existen diferencias?
- g) Compruebe si el nivel de ansiedad con la dosis de 10 mg. difiere del nivel de las restantes consideradas juntas.

3. Supongamos que hemos vuelto a medir el número de horas dormidas por los sujetos transcurrido un mes (los resultados aparecen en el Ejercicio 1 de los ejercicios propuestos en el capítulo 6). Según esto, ¿Existen diferencias significativas entre la media de `noche4` y la media de cada una de las otras tres noches con $\alpha = 0,05$?
4. Un profesor de matemáticas de una facultad evalúa su asignatura a partir de tres controles que se realizan a lo largo del curso y hacen media con la nota del examen final. El profesor desea saber si el rendimiento de los alumnos ha ido aumentando en cada uno de los controles. Para ello selecciona aleatoriamente una muestra de 5 alumnos. Sus calificaciones obtenidas en los tres controles se muestran en la siguiente tabla:

Control	1	5	4	5	3	1
Control	2	6	5	6	4	3
Control	3	7	6	8	5	4

- a) Introduzca los datos en *R* utilizando un marco de datos.
- b) Realice el ANOVA y tome una decisión acerca de la hipótesis nula con $\alpha = 0,05$.
- c) ¿Entre qué pares de medias existen diferencias significativas?

Nota: **Para responder a estos ejercicios quizá te ayude consultar la tabla del anexo 6.4**, que incluye un resumen de los comandos que se han visto en este capítulo para llevar a cabo un ANOVA de un factor.

8 Análisis de varianza de dos factores

El ANOVA de dos factores se caracteriza porque se estudia la relación entre una variable dependiente cuantitativa y dos variables independientes (factores) que se tratan como variables cualitativas. En este tipo de análisis se realizan contrastes de hipótesis sobre los efectos de cada uno de los factores (efecto principal) y el efecto de interacción. Lo específico de esta técnica es la posibilidad de estudiar el *efecto de interacción*, dado que los efectos principales puede analizarse haciendo ANOVAs de un factor por separado para cada uno de ellos.

Se denomina *casilla* a cada una de las combinaciones que pueden darse entre los niveles de las variables independientes. Por ejemplo, si estudiamos el efecto que tienen los factores **sexo**(hombre/mujer) y **edad** (joven/intermedio/adulto) sobre la variable **noche3**, tenemos un diseño con seis casillas. Las personas son diferentes en cada una de las casillas, por lo que este diseño es de tipo inter-sujetos.

Al tener dos factores, pueden darse otros dos diseños, además del inter-sujetos. En un diseño intra-sujetos, las personas son las mismas en todas las casillas. Por ejemplo, si estudiamos dos trastornos (ansiedad/depresión) medidos sobre las mismas personas en dos momentos temporales (antes/después de una terapia), tenemos un diseño con cuatro casillas. Las mismas personas se repiten en las cuatro casillas (aunque sus puntuaciones cambien entre los dos trastornos evaluados o los dos momentos temporales), por lo que este diseño es de medidas repetidas o intra-sujetos.

En los contextos clínicos se da con mucha frecuencia el diseño mixto. Supongamos que tenemos dos grupos de personas (reciben tratamiento/grupo control) evaluados en dos momentos temporales (al comienzo de una terapia/después de un mes). Uno de los factores, el grupo, es inter-sujetos porque son personas distintas las que reciben tratamiento y las que no. El otro factor, el momento temporal, es intra-sujetos porque cada persona es evaluada dos veces. Se trata entonces de un diseño mixto, o que combina factores de distinto tipo, al cual se le denomina también *split-plot*.

8.1. Diseño inter-sujetos

Continuando con el ejemplo del fichero *terapia.dat* del anexo 5, vamos a realizar ahora el ANOVA de **horas** en función de las variables **edad** y **sexo**.

8.1.1. Preparación de los datos

En primer lugar creamos el factor `fedad` y el factor `fsexo` a partir de las variables `edad` y `sexo`. Estos factores no son más que variables que contienen los mismos datos que `edad` y `sexo`, pero *R* interpreta que son variables medidas en una escala nominal:¹

```
fedad <- factor(edad)
fsexo <- factor(sexo)
```

8.1.2. Análisis descriptivos

En cualquier contraste de hipótesis, y mucho más en diseños más complejos, como el del ANOVA de dos factores, es muy conveniente comenzar analizando los datos a nivel descriptivo para obtener una primera impresión. Podemos calcular las medias de los grupos mediante el comando `aggregate`, que muestra la media de `horas` para cada combinación de valores de las variables independientes:

```
aggregate(horas, by=list(fedad,fsexo), mean)
```

```
##   Group.1 Group.2      x
## 1      1      0 15.02500
## 2      2      0 21.63333
## 3      3      0 17.70000
## 4      1      1 15.90000
## 5      2      1 22.70000
## 6      3      1 19.36667
```

8.1.3. Tabla de ANOVA

Al igual que el ANOVA de un factor, el ANOVA de dos factores se realiza mediante la función `aov`. Simplemente tenemos que especificar cuáles son los dos factores en la fórmula que le pasamos a `aov` como argumento de entrada:

```
fit <- aov(horas~fedad*fsexo)
summary(fit)
```

¹Recordemos que ya habíamos utilizado `fedad` para realizar contrastes polinómicos en las comparaciones de tendencia, por lo que si ya lo tenemos definido, no se necesita crearlo de nuevo

La sintaxis `horas~fedad*fsexo` significa que `horas` es la variable dependiente y los factores son `fedad` y `fsexo`. Con el asterisco, `fedad*fsexo`, estamos diciéndole a `aov` que realice el contraste sobre la interacción entre los dos factores. Si hubiésemos estimado el modelo utilizando el signo `+`, es decir:

```
fitNI <- aov(horas~fedad+fsexo)
summary(fitNI)
```

la función `aov` no habría realizado el contraste sobre la interacción. En el resto de este apartado asumimos que el modelo se ha estimado incluyendo la interacción.

El comando `summary()` nos muestra la tabla de ANOVA con los tres posibles efectos en un diseño de dos factores: el efecto principal de `fedad`, el efecto principal de `fsexo` y el efecto de la interacción entre ambos factores. El nivel crítico muestra que únicamente resulta estadísticamente significativo el efecto principal de `fedad`.

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## fedad      2 164.24   82.12   11.223 0.00123 **
## fsexo      1   6.89    6.89    0.942 0.34823
## fedad:fsexo 2   0.54    0.27    0.037 0.96408
## Residuals 14 102.44    7.32
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Existen algunos comandos que nos permiten hacer un inspección más detallada de los datos. Por ejemplo, `coef(fit)` muestra los parámetros estimados del modelo ANOVA:

```
## The following objects are masked from datoss (pos = 3):
##
##      edad, horas, sexo

##      (Intercept)          fedad2          fedad3          fsexo1 fedad2:fsexo1
##      15.0250000      6.6083333      2.6750000      0.8750000      0.1916667
## fedad3:fsexo1
##      0.7916667
```

Utilizando `model.tables(fit,"means")`, obtenemos la tabla de medias de nuestro análisis, donde podemos ver la media total de la muestra (denominada **Grand mean**). Después aparecen las medias marginales de `fedad` y `fsexo`, las medias de las casillas y el tamaño de cada grupo (`rep`).

```
## The following objects are masked from datoss (pos = 3):
##
##      edad, horas, sexo

## The following objects are masked from datoss (pos = 4):
##
##      edad, horas, sexo

## Tables of means
## Grand mean
##
## 18.735
##
## fedad
##      1      2      3
##    15.4 22.2 18.5
## rep  7.0  7.0  6.0
##
## fsexo
##      0      1
##    18.2 19.3
## rep 10.0 10.0
##
## fedad:fsexo
##      fsexo
## fedad 0      1
##   1   15.03 15.90
##   rep  4.00  3.00
##   2   21.63 22.70
##   rep  3.00  4.00
##   3   17.70 19.37
##   rep  3.00  3.00
```

8.1.4. Medidas del tamaño del efecto

El estadístico η^2 tiene dos versiones. En su versión más sencilla, el eta-cuadrado es la proporción de variabilidad debida a un efecto con respecto a la variabilidad total. Por ejemplo, el eta-cuadrado debido al factor **fedad** es:

$$\eta^2 = \frac{SS_{fedad}}{SS_{Total}}.$$

En la medida eta-cuadrado parcial se estima la proporción de varianza debida a un factor después de eliminar la variabilidad debida a los demás factores. Esto se traduce en el cálculo:

$$\eta_{\text{parcial}}^2 = \frac{SS_{\text{fedad}}}{SS_{\text{fedad}} + SS_{\text{Error}}}.$$

Tal y como vimos en el capítulo anterior, para obtener las medidas del tamaño del efecto puede usarse la función `eta_squared` de la librería `effectSize`. Por defecto, esta función calcula el eta-cuadrado parcial, por lo que si queremos el eta-cuadrado debemos incluir el argumento `partial=FALSE`:

```
library(effectsize)
eta_squared(fit,partial=FALSE)

## # Effect Size for ANOVA (Type I)
##
## Parameter      |      Eta2 |      95% CI
## -----
## fedad          |      0.60 | [0.25, 1.00]
## fsexo          |      0.03 | [0.00, 1.00]
## fedad:fsexo    | 1.96e-03 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at (1).
```

Para obtener el eta-cuadrado parcial, basta con llamar a `eta_squared` sin incluir ningún argumento:

```
eta_squared(fit)

## # Effect Size for ANOVA (Type I)
##
## Parameter      | Eta2 (partial) |      95% CI
## -----
## fedad          |      0.62 | [0.27, 1.00]
## fsexo          |      0.06 | [0.00, 1.00]
## fedad:fsexo    | 5.21e-03 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at (1).
```

Las medidas de tamaño del efecto ϵ^2 y ω^2 pueden obtenerse con las funciones `epsilon_squared` y `omega_squared`. También puede obtenerse la versión parcial de estos estadísticos del mismo modo que con η^2 :

```
epsilon_squared(fit,partial=FALSE) # epsilon-cuadrado
epsilon_squared(fit)               # epsilon-cuadrado parcial
omega_squared(fit,partial=FALSE)  # omega-cuadrado
omeg_squared(fit)                 # omega-cuadrado parcial
```

8.1.5. Representación gráfica de la interacción

Por último, veamos una sintaxis para elaborar un gráfico de líneas que muestre el efecto de `fedad`, de `fsexo` y de la interacción entre `fedad` y `fsexo`:

```
interaction.plot(fedad, fsexo, horas,
                 legend=FALSE,
                 type="b",           # Mostrar puntos y líneas
                 ylab="Media de horas", # Etiqueta del eje y
                 col=c("firebrick","navyblue"), # Colores de las líneas
                 lwd=2,              # Ancho de línea
                 pch=c(17,19))      # Forma de los marcadores

legend(x=2.7,y=22.5,
       title="Sexo",
       legend=c("Mujer","Varón"), # poner etiqueta a fsexo
       pch=c(17,19),
       lwd=2,
       col=c("firebrick","navyblue"))
```

Como puede verse en la Figura 8.1, el gráfico de medias presenta unas líneas casi paralelas, que indican que la interacción no ha resultado significativa.

8.1.6. Comparaciones múltiples

En el ANOVA de dos factores podemos aplicar la *prueba de Tukey* para realizar todas las comparaciones por pares de medias, tanto para los efectos principales como los de la interacción. Para ello utilizamos la siguiente función:

```
TukeyHSD(fit)
```

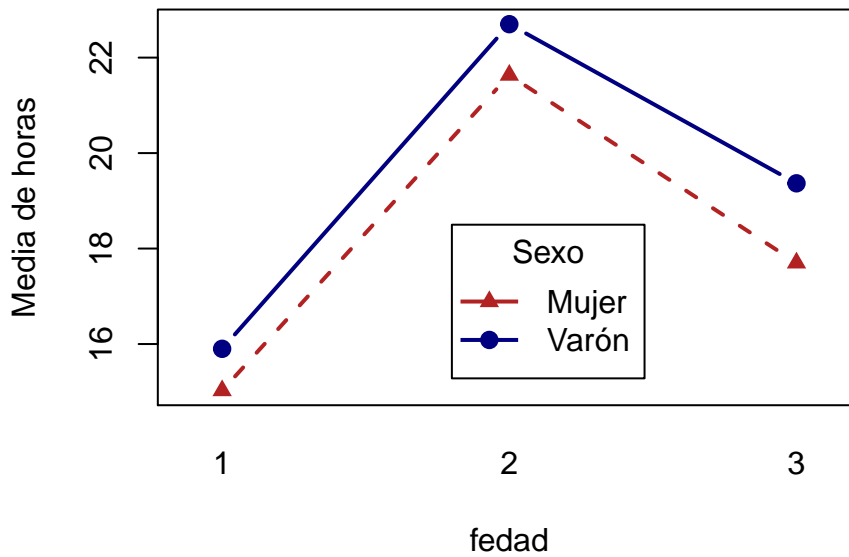


Figura 8.1: Gráfico de medias en el ANOVA inter-sujetos

En el ejemplo que estamos viendo, únicamente resulta significativo el efecto debido al factor `fsexo`. Por ello, no tendría sentido aplicar Tukey sobre `fsexo` ni sobre la interacción. Para decirle a *R* que aplique Tukey solamente sobre `fedad`, llamamos a la siguiente función:

```
TukeyHSD(fit, which = "fedad")
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = horas ~ fedad * fsexo)
##
## $fedad
##          diff          lwr          upr      p adj
## 2-1  6.842857  3.0585291 10.6271852 0.0008764
## 3-1  3.133333 -0.8055202  7.0721869 0.1296555
## 3-2 -3.709524 -7.6483774  0.2293297 0.0661219
```

8.2. Diseño intra-sujetos o de medidas repetidas

El ANOVA de dos factores con medidas repetidas en ambos vamos a verlo con un ejemplo nuevo, dado que los datos del fichero *terapia.dat* no contienen variables adecuadas para este análisis.

8.2.1. Preparación de los datos

Supongamos que, en un estudio sobre psicología de la memoria, se registra el número de errores cometidos por seis sujetos en dos condiciones: recuerdo y reconocimiento (factor **condición**), y en dos momentos temporales: después de una hora y después de un día (factor **momento**). Al ser las mismas personas las que pasan por todas las condiciones experimentales, se trata de un diseño con dos factores y medidas repetidas en ambos.

Para realizar el análisis en *R* debemos organizar los datos en una tabla con el siguiente formato:

<i>sujeto</i>	<i>condicion</i>	<i>momento</i>	<i>errores</i>
1	1	1	4
2	1	1	6
3	1	1	1
4	1	1	2
5	1	1	5
6	1	1	1
1	1	2	5
2	1	2	8
3	1	2	6
4	1	2	10
5	1	2	10
6	1	2	7
1	2	1	1
2	2	1	3
3	2	1	3
4	2	1	1
5	2	1	5
6	2	1	2
1	2	2	4
2	2	2	6
3	2	2	5
4	2	2	4
5	2	2	6
6	2	2	8

Podemos advertir que los resultados de un mismo sujeto aparecen en más de una fila de la matriz. En concreto, cada sujeto aparece una vez por cada una de las condiciones experimentales por las que ha pasado. Por este motivo, necesitamos definir la columna **sujeto** con el código de identificación de cada participante que permita a *R* saber cuáles son las filas de la tabla de datos que se refieren a un mismo sujeto.

Para introducir estos datos en *R* podemos crear una matriz mediante **matrix**, luego la convertimos en un *marco de datos* con **data.frame** y asignamos los nombres de las variables con **colnames**. Por último, utilizamos **factor** para convertir las tres primeras columnas de la matriz en variables de tipo nominal (*factores*). Los siguientes comandos realizan estas operaciones:

```
memoria <- matrix(c(
1, 1, 1, 4,
2, 1, 1, 6,
3, 1, 1, 1,
4, 1, 1, 2,
5, 1, 1, 5,
6, 1, 1, 1,
1, 1, 2, 5,
2, 1, 2, 8,
3, 1, 2, 6,
4, 1, 2, 10,
5, 1, 2, 10,
6, 1, 2, 7,
1, 2, 1, 1,
2, 2, 1, 3,
3, 2, 1, 3,
4, 2, 1, 1,
5, 2, 1, 5,
6, 2, 1, 2,
1, 2, 2, 4,
2, 2, 2, 6,
3, 2, 2, 5,
4, 2, 2, 4,
5, 2, 2, 6,
6, 2, 2, 8),
nrow=24,byrow=T)

memoria <- as.data.frame(memoria)
colnames(memoria) <- c('sujeto','condicion','momento','errores')
memoria[,1] <- factor(memoria[,1])
memoria[,2] <- factor(memoria[,2])
memoria[,3] <- factor(memoria[,3])
```

8.2.2. Tabla de ANOVA

Una vez introducidos los datos en el formato adecuado, basta utilizar `aov` para realizar el ANOVA. La llamada a la función es:

```
fit <-aov(errores~condicion*momento + Error(sujeto/(condicion*momento)),
        data=memoria)
```

La función `aov` recibe como argumento de entrada una fórmula que indica cuál es el papel de cada variable en el diseño. Esta fórmula es: `errores~condicion*momento + Error(sujeto/(condicion*momento))`, y significa lo siguiente:

- `errores` es la variable dependiente, por lo que aparece a la izquierda de la fórmula.
- A la derecha del símbolo `~` ponemos las variables independientes. Al escribir `condicion*momento` estamos diciendo que estas dos son las variables independientes, y con el símbolo `*` decimos que también hay que analizar el efecto de interacción entre ellas.²
- El término `Error(sujeto/(condicion*momento))` indica que en cada una de las condiciones definidas por los valores de `condición` y `momento`, hay un grupo de sujetos que difieren entre sí al azar. De modo general, en un ANOVA con medidas repetidas tenemos que incluir en la fórmula un término error con el formato `Error(id/factores)`, donde `id` es la variable de identificación de los sujetos, y `factores` se refiere a la/s variable/s independiente/s intra-sujetos.
- Por último, con `data=memoria` estamos diciendo a *R* que las variables mencionadas en la fórmula están contenidas en el marco de datos `memoria`.

Los resultados de la función `aov` los hemos guardado en el objeto `fit`, y podemos visualizarlos con:

```
summary(fit)
```

En la salida de resultados aparece, en primer lugar, una tabla de ANOVA para el efecto principal de `condicion`, en la que podemos ver que no es significativo porque el nivel crítico es mayor que cualquiera de los niveles de significación habituales. Después tenemos la tabla para el efecto principal de `momento`, en la que sí se aprecia un efecto significativo. Finalmente aparece la tabla de la interacción, en la que no hay un efecto significativo.

²Si hubiésemos escrito `errores~condicion+momento`, es decir con el signo `+` en lugar de `*`, *R* habría entendido que no hay que analizar el efecto de interacción entre los factores.

```
##
## Error: sujeto
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5  27.71   5.542
##
## Error: sujeto:condicion
##           Df Sum Sq Mean Sq F value Pr(>F)
## condicion  1  12.04  12.042   3.833  0.108
## Residuals  5  15.71   3.142
##
## Error: sujeto:momento
##           Df Sum Sq Mean Sq F value Pr(>F)
## momento   1  84.38   84.38  31.54 0.00248 **
## Residuals  5  13.38    2.68
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: sujeto:condicion:momento
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  6  13.75    2.292
```

Por último, podemos utilizar la función `eta_squared` para calcular la medida del tamaño del efecto:

```
library(effectsize)
eta_squared(fit,partial=FALSE)
```

```
## # Effect Size for ANOVA (Type I)
##
## Group          | Parameter | Eta2 |      95% CI
## -----
## sujeto:condicion | condicion | 0.07 | [0.00, 1.00]
## sujeto:momento   | momento  | 0.51 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at (1).
```

8.3. Diseño mixto

Un ANOVA de dos factores con un diseño mixto, o *split-plot*, es aquel en que un factor es inter-sujetos y otro intra-sujetos.

8.3.1. Preparación de los datos

Para realizar cualquier tipo de ANOVA en *R*, es necesario disponer de un marco de datos en el que la variable dependiente ocupe una única columna. La/s variable/s independientes se codifican también como columnas. Al igual que ocurre en el análisis de medidas repetidas, en un diseño mixto es necesario tener también una columna que indique el código de identificación del sujeto y programar una fórmula con el término `Error(id/factores)`, siendo `id` la variable de identificación y `factores` el/los factor/es de medidas repetidas.

En el apartado 7.3 de ANOVA de varianza con medidas repetidas del capítulo anterior vimos el análisis de la variable `nhoras` en función del factor intra-sujetos `fnoche`. En este apartado vamos a ver el análisis de `nhoras` en función de `fnoche` y `fedad`, siendo este último un factor inter-sujetos.

De forma similar al modelo ANOVA-MR, el primer paso es organizar los datos en un marco de datos en el cual tengamos los datos dispuestos por columnas, una con las etiquetas de identificación del sujeto, otra para la variable dependiente, y una columna para cada variable independiente. Aunque en los datos del ejemplo tenemos 20 sujetos, se muestran aquí únicamente los datos de los cinco primeros sujetos, para ahorrar espacio:

<i>fsujeto</i>	<i>fnoche</i>	<i>nhoras</i>	<i>fedad</i>
1	1	4,0	1
2	1	5,8	2
3	1	4,1	3
4	1	4,3	1
5	1	5,4	2
1	2	4,5	1
2	2	6,4	2
3	2	4,9	3
4	2	5,0	1
5	2	6,1	2
1	3	5,0	1
2	3	7,0	2
3	3	5,1	3
4	3	5,6	1
5	3	6,3	2

Siguiendo una lógica similar a la del apartado 7.3, creamos el marco de datos `dat` con los datos organizados de este modo. La única novedad en este apartado es que se ha incluido `fedad` en el objeto `dat`:

```
# Creamos matrices con los datos de cada noche
dat1 <- cbind(n, 1, noche1, fedad)
dat2 <- cbind(n, 2, noche2, fedad)
dat3 <- cbind(n, 3, noche3, fedad)

# Creamos una matriz de datos poniendo dat1, dat2 y dat3 uno encima
# del otro
dat <- rbind(dat1,dat2,dat3)

# Asignamos los nombres de variables a las columnas de la
# matriz dat
colnames(dat) <- c("fsujeto", "fnoche", "nhoras", "fedad")

# Convertimos dat en un objeto de tipo marco de datos
dat <- as.data.frame(dat)

# Convertimos la variable fsujeto en nominal
dat[, "fsujeto"] <- as.factor(dat[, "fsujeto"])

# Convertimos la variable fnoche en nominal
dat[, "fnoche"] <- as.factor(dat[, "fnoche"])

# Convertimos la variable fedad en nominal
dat[, "fedad"] <- as.factor(dat[, "fedad"])
```

8.3.2. Tabla de ANOVA

Una vez creado el marco de datos, `dat`, podemos realizar el ANOVA para obtener la prueba de significación. El análisis se realiza con la función `aov`, la cual recibe como argumento de entrada una fórmula que especifica el análisis a realizar. En el ejemplo, esta fórmula es `nhoras~(fnoche*fedad) + Error(fsujeto/fnoche)`, que significa que `nhoras` es la variable dependiente y los factores son `fnoche` y `fedad`. El código `fnoche*fedad` significa que también hay que analizar si la interacción es significativa. Si en su lugar hubiésemos incluido en la fórmula el término `fnoche+fedad`, el comando `aov` no analizaría la interacción.

También hemos incluido en la fórmula el término `Error(fsujeto/fnoche)`, que significa que `fsujeto` es la variable con el código de identificación del sujeto y `fnoche` es el factor

intra-sujetos. Estadísticamente, este término especifica que existe variabilidad error de los sujetos que son analizados un mismo día.

Con los siguientes comandos realizamos el análisis mediante la función `aov` y vemos el resultado con `summary`:

```
fit <- aov(nhoras~fnoche*fedad + Error(fsujeto/fnoche), dat)
summary(fit)
```

La salida de resultados muestra dos tablas de ANOVA separadas, una para los análisis que solo incluyen factores inter-sujetos (`fedad`, en el ejemplo), y otra para los análisis en las que está incluido el factor intra-sujetos (`fnoche`). El nivel crítico está especificado en la columna `Pr(>F)`, y podemos ver que el efecto principal de los dos factores es significativo pero el efecto de interacción no alcanza la significación estadística.

```
##
## Error: fsujeto
##           Df Sum Sq Mean Sq F value    Pr(>F)
## fedad      2  54.75   27.373    12.71 0.000422 ***
## Residuals 17   36.62    2.154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: fsujeto:fnoche
##           Df Sum Sq Mean Sq F value    Pr(>F)
## fnoche      2  10.416    5.208  450.365 <2e-16 ***
## fnoche:fedad 4   0.051    0.013    1.099  0.373
## Residuals   34   0.393    0.012
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

8.3.3. Representación gráfica de la interacción

Podemos utilizar la función `interaction.plot`, vista en el apartado 8.1.5, para representar el gráfico de medias como ayuda a la interpretación. En la llamada a esta función, especificamos que el rango de `nhoras` va de 0 a 8 porque 0 es el valor mínimo que podría tomar esta variable y su media más alta en los grupos comparados está un poco por debajo de 8. La leyenda del gráfico la hemos obtenido por separado utilizando la función `legend`.

```

interaction.plot(dat$fnoche,dat$fedad,dat$nhoras,
                legend=FALSE,
                ylim=c(0,8),           # Ajustar el rango del eje y
                type="b",              # Mostrar puntos y lineas
                xlab="Noche",          # Etiqueta del eje x
                ylab="Media de horas", # Etiqueta del eje y
                col=c("coral3","seashell4","aquamarine3"), # Colores de las lineas
                lwd=3,                 # Ancho de linea
                pch=c(17,19,23))      # Forma de los marcadores

legend(x=2.7,y=4,
      title="Edad",
      legend=c("1","2", "3"),        # poner etiquetas a fedad
      pch=c(17,19,23),
      lwd=2,
      col=c("coral3","seashell4","aquamarine3"))

```

En el gráfico de la Figura 8.2 podemos ver que las líneas son casi paralelas, lo que indica que la interacción no ha resultado significativa. Los efectos encontrados en estos datos han sido los de *fedad* y *fnoche*.

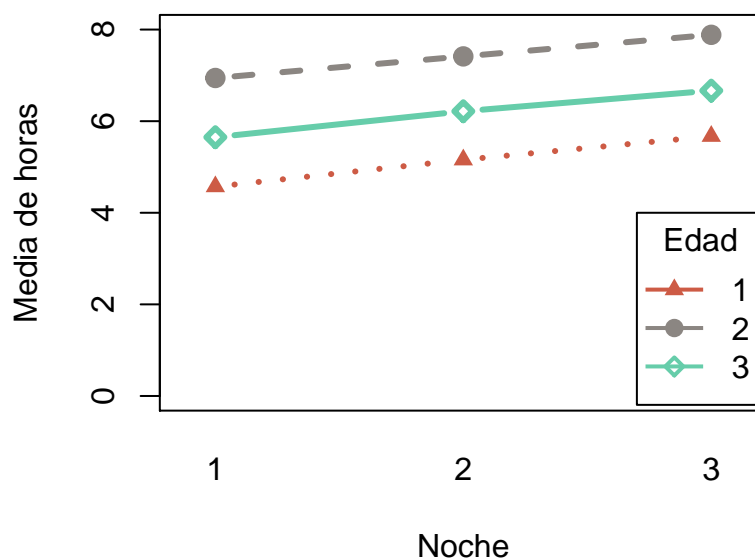


Figura 8.2: Gráfico de medias en el ANOVA split-plot

8.4. Análisis de covarianza

En el análisis de covarianza (ANCOVA) se estudia la relación de una variable dependiente cuantitativa con dos variables independientes. Una de las variables independientes es cualitativa (o factor) y por tanto crea grupos de sujetos. La segunda variable independiente, también denominada *covariable*, es cuantitativa. Por tanto, el análisis de covarianza combina elementos del ANOVA y del análisis de regresión. Por una parte analiza si el intercepto de la regresión cambia entre los grupos definidos por el factor. Por otra parte, se estima la pendiente de la regresión de la variable dependiente sobre la covariable, asumiendo además que dicha pendiente es igual en todos los grupos.

El comando `aov` también se utiliza para realizar el ANCOVA con *R*. Desde un punto de vista estadístico, tanto el ANOVA como la regresión forman parte del modelo lineal general, en el que se pueden incluir variables independientes cuantitativas o cualitativas. Por tanto, todos estos análisis se realizan en *R* de un modo similar, simplemente especificando cuál es el nivel de medida de cada variable independiente.

Según hemos visto en apartados anteriores, es necesario pasarle una fórmula a `aov` que le indique cuáles son las variables independientes y la dependiente. Para realizar el ANCOVA debemos introducir dos variables independientes en la fórmula en un orden determinado, en primer lugar hay que indicar la covariable y después la variable independiente cualitativa.

Ilustraremos esta técnica con un ejemplo nuevo. En una empresa se ha analizado el salario de sus empleados (miles de euros mensuales) en función de los años de experiencia en la empresa y del nivel educativo (1: primaria, 2: secundaria, 3: universidad). El siguiente código carga los datos en *R*:

```
experiencia <- c(4, 10, 6, 8, 5, 1, 7, 3, 2, 9, 5, 8)
salario     <- c(2, 8, 4, 5, 3, 1, 5, 1, 2, 4, 2, 4)
educacion   <- c(2, 1, 2, 1, 2, 3, 1, 3, 3, 1, 2, 1)
```

El propósito del análisis es investigar la relación entre el salario y el nivel educativo de los empleados. Para ello, calculamos las medias de cada grupo y realizamos un ANOVA:

```
feduccion<-factor(educacion)           # crear el factor feduccion
aggregate(salario,list(feduccion),mean) # Tabla de medias
fit1 <- aov(salario~feduccion)          # ANOVA
summary(fit1)                          # Prueba de significación del ANOVA
```

El ANOVA muestra un resultado sorprendente, el salario es menor a mayor nivel educativo.


```
##      Group.1      x
## 1         1 5.200000
## 2         2 2.750000
## 3         3 1.333333

##              Df Sum Sq Mean Sq F value Pr(>F)
## feduccion    2  30.70   15.35   9.717 0.00565 **
## Residuals    9   14.22    1.58
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Realizaremos un ANCOVA para investigar más a fondo sobre estos datos. El ANCOVA se basa en calcular regresiones lineales por separado en cada uno de los grupos formados por la variable independiente cualitativa. En este ejemplo las regresiones son

$$\begin{aligned} \text{salario} &= \alpha_1 + \beta \times \text{experiencia} + \text{error} && (\text{grupo de educación primaria}) \\ \text{salario} &= \alpha_1 + \alpha_2 + \beta \times \text{experiencia} + \text{error} && (\text{grupo de educación secundaria}) \\ \text{salario} &= \alpha_1 + \alpha_3 + \beta \times \text{experiencia} + \text{error} && (\text{grupo de educación universidad}) \end{aligned}$$

Este modelo incluye tres parámetros para representar la constante de la regresión: el intercepto (α_1), el efecto de la educación secundaria (α_2) y el efecto de la educación universitaria (α_3). Si α_2 fuese igual a cero no habría efecto de la educación secundaria en el salario, y si $\alpha_3 = 0$ no habría efecto de la educación universitaria.

El otro parámetro del modelo es la pendiente de la regresión, β . El ANCOVA asume que la pendiente es igual en todos los grupos analizados, y representa el efecto de la experiencia en el salario. Finalmente, el modelo incluye un término error con media 0 y cuya varianza es la varianza error del análisis.

Para realizar el ANCOVA, llamamos a la función `aov` y le pasamos la fórmula que indica que `salario` es la variable dependiente, `experiencia` es la covariable y `feduccion` es el factor. Es importante poner las variables en este orden, de modo que la covariable se escribe antes que el factor.

```
fit2 <- aov(salario~experiencia+feduccion) # ANCOVA
```

La salida de resultados muestra que no existe un efecto significativo de la educación sino de la experiencia. Es decir, lo que está ocurriendo en estos datos es que los empleados con distinto nivel educativo varían también en los años de experiencia en la empresa. Al controlar el efecto de la variable `experiencia`, el efecto de `feduccion` desaparece.

```
summary(fit2)      # Ver la Prueba de significación del ANCOVA

##      Group.1      x
## 1          1 5.200000
## 2          2 2.750000
## 3          3 1.333333

##              Df Sum Sq Mean Sq F value    Pr(>F)
## experiencia   1   34.95    34.95   29.487 0.000623 ***
## feduccion     2    0.49     0.24    0.205 0.818726
## Residuals     8    9.48     1.19
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La función `coef` muestra los parámetros estimados. La tabla de la prueba de significación del ANCOVA, vista anteriormente, indicó que el parámetro β es significativamente distinto de cero, pero no así α_2 ni α_3 .

```
coef(fit2)

## (Intercept) experiencia feduccion2 feduccion3
## -0.82608696  0.71739130 -0.01086957  0.72463768
```

El diagrama de dispersión de la Figura 8.3 proporciona información adicional acerca de lo que ocurre en estos datos. Los empleados de mayor experiencia son los que tienen menor nivel educativo, y los universitarios son los que menos tiempo llevan en la empresa. La relación entre el salario y la experiencia en la empresa enmascara el efecto del nivel educativo. Una vez que hemos controlado la variable `experiencia`, el ANCOVA no encuentra un efecto significativo de la educación.

```
plot(experiencia,salario,
     col=c("chartreuse3","chocolate3","cadetblue3")[educacion],
     pch=19,cex=1.5)
legend(1,8,
     title = "Educación",
     legend = c("Primaria","Secundaria","Universitaria"),
     pch=19,
     col=c("chartreuse3","chocolate3","cadetblue3"))
```

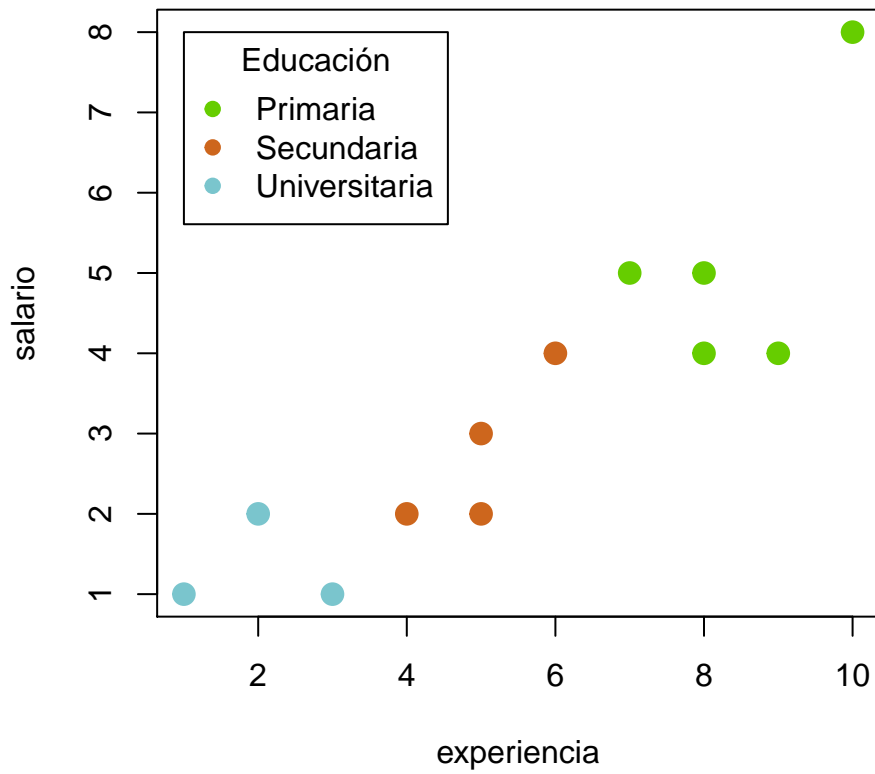


Figura 8.3: Gráfico de dispersión en un ANCOVA

8.5. Ejercicios propuestos

- Supongamos que se dispone de los datos sobre si los sujetos considerados son fumadores (1) o no lo son (0). Estos datos aparecen en el ejercicio 5 del apartado 5.6 del capítulo 5. Según esto, contrastar el efecto de las variables `edad` y `fumar` sobre la variable `horas` con $\alpha = 0,05$.
- Un gabinete de psicología clínica pretende estudiar la eficacia de cuatro terapias (psicoanalítica, conductista, cognitiva y gestáltica) en el tratamiento de los trastornos del sueño. Para ello asigna aleatoriamente a un grupo de 24 pacientes (mitad varones, mitad mujeres) a cada terapia y mide las horas que duermen transcurrido un mes después de la terapia. Los resultados se muestran en la siguiente tabla:

	Varones			Mujeres		
Psicoanalítica	4	5	3	4	3	4
Conductista	8	7	7	6	7	5
Cognitivista	8	9	8	7	8	6
Gestáltica	6	5	5	4	5	3

- ¿Qué puede concluirse con $\alpha = 0,01$?
 - ¿Qué terapia recomendarías a un paciente que acudiera a tu consulta con insomnio?
 - Representa gráficamente el efecto de la interacción e interpreta el resultado
- Con los datos del fichero *terapia.dat*, contraste el efecto de las variables `noche` y `sexo` sobre el número de horas dormidas con $\alpha = 0,05$.

Nota: Todos los ejercicios (excepto el segundo) hacen referencia al fichero *terapia.dat* descrito en el anexo 5. **Para responder a estos ejercicios quizá te ayude consultar la tabla del anexo 6.4**, que incluye un resumen de los comandos que se han visto en este capítulo para llevar a cabo un ANOVA de dos factores.

9 Contrastes no paramétricos

Los contrastes no paramétricos son un conjunto de pruebas estadísticas que se caracterizan por no especificar completamente cual es la función de distribución de la que proceden los datos. Estas técnicas se aplican con distintos propósitos, y no existe una base teórica común a todas ellas. Una de las situaciones más habituales es emplearlas como alternativa a los contrastes tradicionales con los estadísticos T y F cuando no se cumple la normalidad de las variables. Existen contrastes no paramétricos, como la prueba de *Kolmogorov-Smirnov*, que sirven para este propósito. En este capítulo revisaremos brevemente los contrastes no paramétricos más comúnmente empleados.

9.1. Prueba de Kolmogorov-Smirnov

El contraste de Kolmogorov-Smirnov permite contrastar la hipótesis de que la función de distribución de una variable sigue una distribución teórica (normal, Poisson, gamma, etc.). A modo de ejemplo, supongamos que vamos a analizar la normalidad de la variable *horas*. Podemos comenzar haciendo un histograma de frecuencias para obtener una primera impresión visual.

El siguiente código realiza el histograma de la Figura 9.1, superpone la curva normal que más se aproxima al histograma (línea continua) y añade la función de densidad, no necesariamente normal, que más se aproxima al histograma (línea discontinua).

```
histograma <- hist(horas,col="lightblue",
                  main="histograma de horas",ylab="frecuencia")
coeficiente <- histograma$counts / histograma$density
densidad <- density(horas)
densidad$y <- densidad$y * coeficiente[1]

curve(dnorm(x,mean(horas),sd(horas))*coeficiente[1],add=T,
      lwd=2,lty="solid",col="firebrick")
lines(densidad,lwd=2,lty="dashed",col="navyblue")
```

La Figura 9.1 muestra que hay ciertas discrepancias entre la distribución de *horas* y la curva normal, aunque para valorar su significación es necesario realizar el test de Kolmogorov-Smirnov. La hipótesis nula es $H_0 : \text{horas} \sim \text{normal}$, y la contrastamos

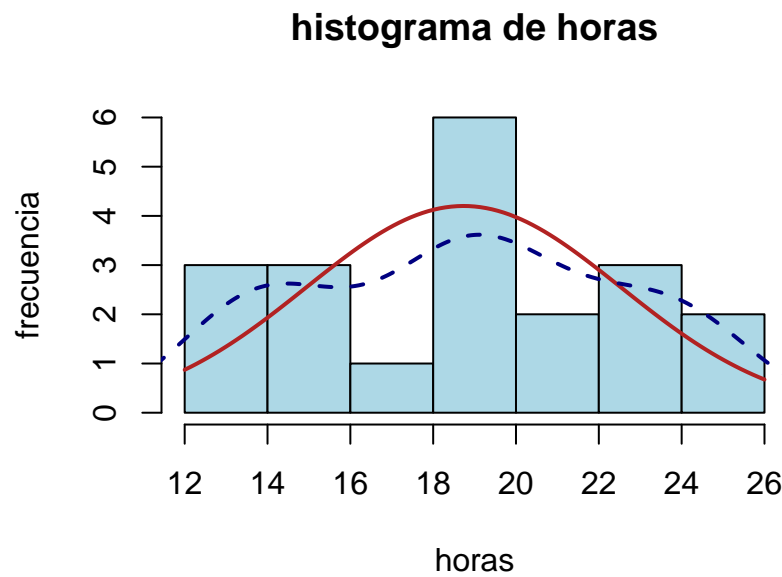


Figura 9.1: Histograma para `horas` con curva normal

con la función `ks.test`, a la cual pasamos como argumentos la variable a analizar y la distribución que queremos evaluar:

```
ks.test(horas, "pnorm")
```

El resultado muestra el valor del estadístico de contraste (D) y el nivel crítico (p -value), que resulta ser significativo, y por tanto, rechazamos la hipótesis nula de normalidad.

```
## Warning in ks.test(horas, "pnorm"): ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
## One-sample Kolmogorov-Smirnov test
##
## data: horas
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

9.2. Prueba de los signos (binomial)

La prueba de los signos se emplea para contrastar hipótesis acerca de los cuantiles de una variable. En el caso más sencillo, se utiliza como alternativa a la prueba T sobre una

media en caso de que no se cumpla el supuesto de normalidad. La prueba de los signos suele utilizarse para contrastar la hipótesis de que la mediana toma un determinado valor; sin embargo, también puede contrastarse una hipótesis nula referida a cualquier otro centil.

9.2.1. Contraste sobre una mediana

Supongamos que queremos contrastar la hipótesis de que la mediana de `noche2` es 7,5. Para ello, en primer lugar, creamos una variable dicotómica llamada `noche2D` que indique cuales de los valores de `noche2` son superiores a 7,5. Después aplicamos una prueba binomial sobre `noche2D` en la cual la hipótesis nula es $H_0 : \pi = 0,50$, lo que significa que en `noche2D` existen tantos valores mayores de 7,5 como valores menores de este valor, como ocurriría si 7,5 fuese la mediana.

El código *R* para este ejemplo es:

```
noche2D <- noche2 > 7.5
binom.test(x=sum(noch2D),
           n=length(noch2D),
           p=0.5)
```

Los resultados de `binom.test` muestran un nivel crítico mayor que los niveles de significación habituales, por lo que no podemos descartar que la mediana sea 7,5.

```
##
## Exact binomial test
##
## data: sum(noch2D) and length(noch2D)
## number of successes = 5, number of trials = 20, p-value = 0.04139
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.08657147 0.49104587
## sample estimates:
## probability of success
##                0.25
```

9.2.2. Comparación de la mediana de dos variables

La prueba de los signos también puede utilizarse como alternativa a la prueba *T* sobre dos medias relacionadas. Supongamos que queremos investigar si el promedio de `noche2` es superior al de `noche1`. La comparación entre ambas se basa en la diferencia $D = \text{noche1}$

- `noche2`, al igual que cuando se realiza una prueba T sobre medias relacionadas. Si la hipótesis nula fuese cierta, cabe esperar que la variable D tenga tantos valores positivos como negativos. Por tanto, podemos calcular el número de valores positivos, y utilizar la distribución binomial ($n; \pi = 0,50$) para calcular su nivel crítico. El código *R* es:

```
D <- noche1 - noche2
 exitos <- sum(D > 0)
 n <- length(noche1)
 binom.test(exitos,n,alternative = "less")

##
## Exact binomial test
##
## data: exitos and n
## number of successes = 0, number of trials = 20, p-value = 9.537e-07
## alternative hypothesis: true probability of success is less than 0.5
## 95 percent confidence interval:
##  0.0000000 0.1391083
## sample estimates:
## probability of success
##                                0
```

Podemos ver que el nivel crítico es prácticamente cero, por lo que concluimos que el promedio de `noche1` es menor que el de `noche2`.

9.3. Prueba de las rachas

La prueba de las rachas examina la aleatoriedad de una variable mediante el estudio de las rachas de valores consecutivos que están por encima o por debajo de la mediana. Una racha es una secuencia de valores consecutivos del mismo tipo. Esta prueba se basa en contar el número de rachas que aparecen en la muestra, y comparar dicho valor con la distribución teórica de valores que se obtiene bajo la hipótesis nula de que la muestra es aleatoria.

Supongamos que queremos contrastar la hipótesis de que la muestra de valores de `horas` es aleatoria. Podemos aplicar la prueba de las rachas con las librerías `snpar` y `randtests`, aunque aquí utilizaremos esa última. Simplemente tenemos que cargar la librería y llamar a la función `runs.test`. Como puede verse, también utilizamos el argumento `plot` para pedir el gráfico de rachas.


```
library(randtests)
runs.test(horas,plot=T)
```

La salida de resultados muestra que hay 12 rachas en la muestra. El nivel crítico es no significativo, por lo que mantenemos la hipótesis de muestra aleatoria. Además, el gráfico de la Figura 9.2 permite visualizar estas doce rachas, los puntos negros son datos por encima de la mediana y los puntos rojos son datos por debajo. Las líneas verticales permiten identificar las 12 rachas.

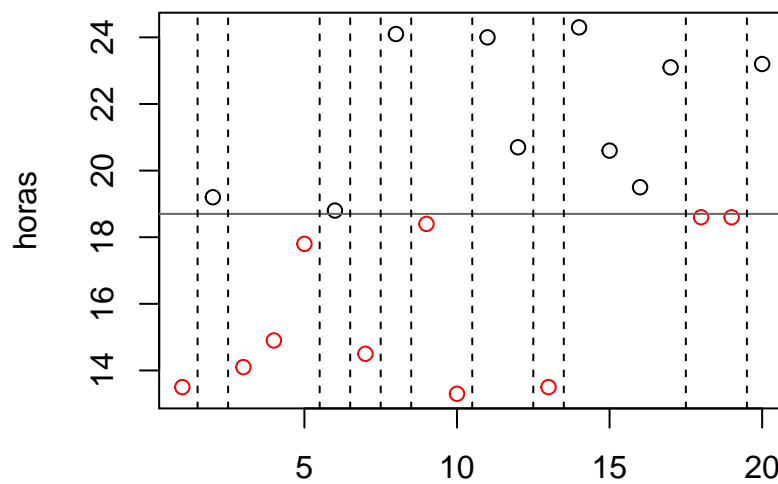


Figura 9.2: Prueba de las rachas

```
##
## Runs Test
##
## data: horas
## statistic = 0.45947, runs = 12, n1 = 10, n2 = 10, n = 20, p-value =
## 0.6459
## alternative hypothesis: nonrandomness
```

9.4. Prueba de Mann-Whitney

La prueba de Mann-Whitney permite contrastar la hipótesis de que los promedios de dos variables son iguales, y constituye una alternativa a la prueba T sobre dos medias cuando no se cumple el supuesto de normalidad.

La forma de realizar el contraste depende de si son dos muestras independientes o relacionadas. Supongamos que deseamos contrastar si el promedio de `noche2` es igual para los sujetos que reciben la terapia 1 y los que no. Se trata de un diseño inter-sujetos con dos muestras independientes. La forma de programarlo en *R* es utilizar la función `wilcox.test` y pasarle la fórmula que representa este diseño:

```
wilcox.test(noche2 ~ terapia1)
```

La salida de resultados muestra el estadístico de contraste W y un nivel de significación menor de 0,01, por lo que descartamos la igualdad de promedios entre ambos grupos.

```
##
## Wilcoxon rank sum test with continuity correction
##
## data:  noche2 by terapia1
## W = 5.5, p-value = 0.001184
## alternative hypothesis: true location shift is not equal to 0
```

Si se trata de dos muestras relacionadas introducimos las dos variables en la función `wilcox.test` e indicamos que se trata de un contraste para muestras relacionadas mediante el argumento `paired=T`. Por ejemplo, contrastamos la hipótesis de igualdad de promedios entre `noche1` y `noche2` mediante el código:

```
wilcox.test(noche1, noche2, paired=T)
```

Podemos ver que el nivel de significación, al ser menor de 0,01, nos permite concluir que los promedios difieren.

```
##
## Wilcoxon signed rank test with continuity correction
##
## data:  noche1 and noche2
## V = 0, p-value = 9.212e-05
## alternative hypothesis: true location shift is not equal to 0
```

9.5. Prueba de Kruskal-Wallis

En el caso de que queramos contrastar la hipótesis de igualdad entre los promedios de más de dos muestras independientes, podemos recurrir a la prueba de Kruskal-Wallis. Por ejemplo, podemos contrastar que el promedio de `noche3` es igual en los tres grupos de edad con el código:

```
kruskal.test(noche3 ~ edad)
```

La salida de resultados muestra un estadístico de contraste chi-cuadrado con dos grados de libertad, y un nivel crítico que lleva a rechazar la hipótesis nula de igualdad de promedios entre los tres grupos de edad.

```
##
##  Kruskal-Wallis rank sum test
##
## data:  noche3 by edad
## Kruskal-Wallis chi-squared = 10.088, df = 2, p-value = 0.006448
```

9.6. Prueba de Friedman

Cuando tenemos más de dos muestras relacionadas, podemos contrastar la hipótesis de igualdad de sus promedios mediante la prueba de Friedman. Al igual que vimos en el apartado de ANOVA con medidas repetidas, para hacer este análisis intra-sujetos es necesario crear un nuevo objeto de datos reorganizando los datos originales. A este nuevo objeto de datos lo llamaremos `n123`, y todos los datos referidos a número de horas dormidas deben aparecer en una única columna a la que denominaremos `nhoras`. También necesitamos una columna con los códigos de identificación del sujeto, a la que denominaremos `n`. Por último, la tercera columna indicará a qué noche se refieren los datos mediante los números 1 a 3. El código para reorganizar los datos es:

```
n1 <- cbind(noche1, n, 1)
n2 <- cbind(noche2, n, 2)
n3 <- cbind(noche3, n, 3)
n123 <- rbind(n1, n2, n3)
colnames(n123) <- c("nhoras", "n", "noche")
```

Una vez tengamos los datos en el formato adecuado, basta llamar a `friedman.test` para realizar este contraste. A esta función debemos pasarle tres variables como argumentos de entrada: la variable dependiente del análisis, la variable independiente (es decir, la que indica a qué noche se refiere cada dato), y la variable con los códigos de identificación del sujeto. Es decir:

```
friedman.test(n123[, "nhoras"], n123[, "noche"], n123[, "n"])
```

A continuación se muestran los resultados de la prueba de Friedman, que obtienen un nivel crítico menor de 0,01; por tanto, descartamos la hipótesis de igualdad de promedios.

```
##  
## Friedman rank sum test  
##  
## data:  n123[, "nhoras"], n123[, "noche"] and n123[, "n"]  
## Friedman chi-squared = 40, df = 2, p-value = 2.061e-09
```

9.7. Ejercicios propuestos

1. En un centro escolar se ha registrado el número de niños y niñas hiperactivos en cinco clases diferentes. Contraste la hipótesis de que en promedio hay dos niños hiperactivos por clase utilizando un nivel de confianza del 95 %. Los datos son:

3 5 6 4 2

2. Se está elaborando un cartel para una campaña publicitaria. Para ello se han seleccionado cuatro fotografías diferentes. Se muestra cada una a cuatro sujetos, quienes deben evaluar de 0 a 10 su atractivo estético. ¿Puede concluirse con $\alpha = 0,05$ que existen diferencias entre los promedios de las fotografías?

Fotografía 1	3	2	1	2
Fotografía 2	4	6	8	6
Fotografía 3	7	4	7	5
Fotografía 4	3	2	3	1

3. Un investigador realiza un experimento para determinar si la ingestión de alcohol afecta a la atención auditiva. Selecciona 18 sujetos a los que divide en tres grupos y les administra diferentes dosis de alcohol (baja, media o alta). A continuación, los sujetos pasan por una prueba de audición, cuyos resultados aparecen en la tabla inferior. Según esto, contraste si hay efecto con $\alpha = 0,01$.

	Dosis		
	Baja	Media	Alta
85		60	60
83		58	48
76		76	38
64		52	47
75		63	50
81			49
78			

Nota: todos los ejercicios incluyen datos que tendrás que introducir en *R* antes de resolverlos. **Para responder a estos ejercicios quizá te ayude consultar la tabla del anexo 6.5**, que incluye un resumen de los comandos que se han visto en este capítulo para llevar a cabo contrastes no paramétricos.

Referencias

- Bergmann, J. y Sams, A. (2012). *Flip your classroom: Reach every student in every class every day*. International Society for Technology in Education, USA.
- Botella, J., Suero, M. y Ximénez, C. (2012). *Análisis de Datos en Psicología I*. Madrid: Pirámide.
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Erlbaum.
- Pardo, A. y San Martín, R. (2015). *Análisis de datos en ciencias sociales y de la salud* (vol II, 2ª ed). Madrid: Síntesis.
- R Development Core Team. (2019). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. <http://www.r-project.org/index.html>
- Revuelta, J. y Ponsoda, V. (2003). *Simulación de modelos estadísticos en ciencias sociales*. Madrid y Salamanca: La Muralla y Hespérides.
- Revuelta, J. y Ponsoda, V. (2005). *Fundamentos de estadística*. 2ª edición. Madrid: UNED ediciones.
- Ximénez, C. y Revuelta, J. (2011). *Cuaderno de prácticas de análisis de datos con SPSS*. Madrid: ediciones UAM.
- Ximénez, C. y San Martín, R. (2000). *Análisis de varianza con medidas repetidas*. Madrid y Salamanca: Editorial la Muralla S.A. y Hespérides.
- Ximénez, C. y San Martín, R. (2004). *Fundamentos de las técnicas multivariantes*. Madrid: UNED ediciones.

ANEXOS

Anexo 1. Archivo de datos *practica.sav*

	Clave	genero	edad	peso	estatura	prov	idprov	rama
1	1	Mujer	34	46	1.63	MADRID	46	Ciencias Sociales y Juridicas
2	2	Mujer	34	51	1.72	BURGOS	22	Otros/Varios
3	3	Varon	25	45	1.53	VALLADOLID	28	Ciencias Sociales y Juridicas
4	4	Varon	26	64	1.75	PALENCIA	24	Ciencias Sociales y Juridicas
5	5	Varon	27	58	1.68	MADRID	46	Enseñanzas tecnicas
6	6	Mujer	29	45	1.64	BURGOS	22	Humanidades
7	7	Varon	28	60	1.70	MADRID	46	Enseñanzas tecnicas
8	8	Mujer	27	45	1.64	BURGOS	22	Ciencias Sociales y Juridicas
9	9	Varon	33	60	1.70	VALLADOLID	28	Humanidades
10	10	Varon	28	54	1.63	BURGOS	22	Ciencias Sociales y Juridicas
11	11	Varon	28	54	1.63	LEON	23	Ciencias Experimentales y de la Salud
12	12	Varon	32	54	1.63	BURGOS	22	Enseñanzas tecnicas
13	13	Varon	35	61	1.71	PALENCIA	24	Ciencias Sociales y Juridicas
14	14	Varon	28	62	1.72	ZAMORA	29	Ciencias Experimentales y de la Salud
15	15	Varon	30	56	1.65	ASTURIAS	12	Humanidades
16	16	Mujer	35	45	1.64	ALAVA	50	Ciencias Sociales y Juridicas

Descripción de variables en el fichero *practica.sav*:¹

genero: variable nominal, donde: (1): varón; (0): mujer.

edad: variable cuantitativa (medida en años)

peso: variable cuantitativa (medida en kilogramos)

estatura: variable cuantitativa (medida en metros)

prov: variable nominal, indicadora de la provincia de residencia

idprov: variable nominal que sirve de identificador numérico de la provincia de residencia, donde:

1 = Almería	14 = Santa Cruz de Tenerife	27 = Soria	40 = La Coruña
2 = Cádiz	15 = Cantabria	28 = Valladolid	41 = Lugo
3 = Córdoba	16 = Albacete	29 = Zamora	42 = Ourense
4 = Granada	17 = Ciudad Real	30 = Barcelona	43 = Pontevedra
5 = Huelva	18 = Cuenca	31 = Girona	44 = Baleares
6 = Jaén	19 = Guadalajara	32 = Lleida	45 = La Rioja
7 = Málaga	20 = Toledo	33 = Tarragona	46 = Madrid
8 = Sevilla	21 = Ávila	34 = Ceuta	47 = Melilla
9 = Huesca	22 = Burgos	35 = Alicante	48 = Murcia
10 = Teruel	23 = León	36 = Castellón	49 = Navarra
11 = Zaragoza	24 = Palencia	37 = Valencia	50 = Arava
12 = Asturias	25 = Salamanca	38 = Badajoz	51 = Guipuzkoa
13 = Las Palmas	26 = Segovia	39 = Cáceres	52 = Bizkaia

rama: variable nominal que indica el área de conocimiento: (0): ciencias experimentales y de la salud, (1): ciencias sociales y jurídicas, (2): enseñanzas técnicas, (3): humanidades y (4) otros/variados.

licen: variable nominal que indica la titulación del sujeto (licenciado en derecho, económicas, etc.).

inteli: inteligencia general (o capacidad cognitiva para resolver problemas en general).

compren: comprensión verbal (o capacidad para comprender los mensajes transmitidos verbalmente).

orient: orientación espacial (o capacidad para situarse en el espacio respecto a alguna referencia).

extra: extraversión (valora la cantidad e intensidad de las relaciones personales).

respon: responsabilidad (grado de organización y motivación de la conducta del sujeto).

emocio: estabilidad emocional (refleja el nivel de ajuste emocional de la persona).

sincer: sinceridad (o grado en que responde con honestidad).

fumar: variable nominal que indica si el sujeto fuma (1) o no fuma (0).

¹ El archivo puede descargarse desde: <https://www.psicologiauam.es/carmen.ximenez/Practicas1.html>

Anexo 2. Tabla resumen comandos R (estadística univariada)

Concepto	Comando
Crear objeto x	<code>x <- 10</code>
Definir datos	<code>(X <- c(1,2,3,4,5,6))</code> <code>1:10</code>
Leer ficheros SPSS	Cargar librería <i>foreign</i> <code>library(foreign)</code> <code>practica = read.spss("practica.sav", to.data.frame=TRUE)</code>
Calcular media de X	<code>mean(X)</code>
Calcular varianza de X	<code>var(X)</code>
Calcular desviación típica de X	<code>sd(X)</code>
Calcular Coefficiente de Variación para <i>edad</i>	<code>CV_edad = (sd(practica\$edad)/mean(practica\$edad)) * 100</code>
Representaciones gráficas Diagrama de barras Histograma Diagrama de sectores Diagrama de cajas Diagrama de cajas (por grupos)	<code>barplot(X)</code> <code>hist(X)</code> <code>pie(X)</code> <code>boxplot(X)</code> <code>boxplot(practica\$estatura ~ practica\$fumar)</code>
Resumir estadísticos	<code>summary(X)</code>
Analizar solo una variable (media para <i>edad</i>)	<code>mean(practica\$edad)</code>
Estadísticos solo para una variable (<i>edad</i>)	<code>summary(practica\$edad)</code>
Distribuciones de frecuencias para <i>edad</i> Elaborar tablas con frecuencias absolutas Elaborar tablas con frecuencias relativas Colocar t1 y t2 en una sola tabla Elaborar tablas con frecuencias absolutas acumuladas Elaborar tablas con frecuencias relativas acumuladas Elaborar Tabla de frecuencias completa	<code>t1 = table(practica\$edad)</code> <code>t2 = prop.table(t1)</code> <code>data.frame(t2)</code> <code>t3 = cumsum(t1)</code> <code>t4 = cumsum(t2)</code> <code>data.frame(t1, t3, t2, t4)</code>
Centiles (C_{10} , C_{25} , C_{50} , C_{75} y C_{90} para <i>edad</i>)	<code>quantile(practica\$edad, c(.10, .25, .50, .75, .90))</code>
Hacer análisis para varones y mujeres por separado (similar a “segmentar archivo” en SPSS)	<code>aggregate(practica\$edad, by=list(practica\$genero), mean)</code> <code>aggregate(practica\$edad, by=list(practica\$genero), sd)</code>
Asimetría y Curtosis	Cargar librería <i>moments</i> <code>library(moments)</code> <code>skewness(practica\$edad)</code> <code>kurtosis(practica\$edad)</code>
Seleccionar casos “solo varones” (similar a “filtrar” casos en SPSS)	<code>practica2 <- subset.data.frame(practica, genero=="Varon")</code>
Puntuaciones típicas $z_i = \frac{X_i - \bar{X}}{S_x}$	<code>zedad <- (practica\$edad - mean(practica\$edad)) /</code> <code>sd(practica\$edad)</code> <code>print(zedad)</code>
Escalas derivadas $T_i = a \cdot z_i + b$ <i>Ejemplo para T = 10 * z + 50 (para edad)</i>	<code>Tedad <- zedad * 10 + 50</code> <code>print(Tedad)</code>

Nota: todos los ejemplos de la tabla hacen referencia al archivo *practica.sav* del anexo 1.

Anexo 3. Tabla resumen comandos R (estadística bivariada)

Concepto	Comando
Correlación de Pearson	<code>cor(notas\$nota09, notas\$nota10)</code>
Covarianza	<code>cov(notas\$nota09, notas\$nota10)</code>
Diagrama de dispersión	<code>plot(notas\$nota09, notas\$nota10)</code>
Matriz de correlaciones, R	<code>cor(practica[, c("extra", "respon", "emocio", "sincer")])</code>
Matriz de Covarianzas, S	<code>cov(practica[, c("extra", "respon", "emocio", "sincer")])</code>
Matriz R solo para mujeres	<code>cor(subset(practica, genero == "Mujer")[, c("extra", "respon", "emocio", "sincer")])</code>
Diagrama de dispersión conjunto	<code>plot(practica[, c("extra", "respon", "emocio", "sincer")])</code>
Combinaciones lineales de 3 variables (y sus propiedades: media y varianza)	<code>T <- X + Y + Z</code> <code>mean(T)</code> <code>var(T)</code>
Regresión lineal simple $Y_i = A + B \cdot X_i$	<code>regresion <- lm(peso ~ estatura, data = practica)</code> <code>summary(regresion)</code> # VD es peso y VI estatura
<i>Gráfico de regresión</i> (con línea superpuesta)	<code>plot(practica\$estatura, practica\$peso, main = "Regresión Peso sobre Estatura", xlab = "estatura", ylab = "peso")</code> <code>abline(regresion)</code>
Predicciones del modelo de regresión (Y'_i)	<code>pre <- predict(regresion, data.frame(practica))</code>
Residuos del modelo ($Y_i - Y'_i$)	<code>res <- resid(regresion)</code>
Descomposición de la varianza del criterio $S^2_Y = S^2_{Y'} + S^2_{Y \cdot X}$	<code>var(practica\$peso)</code> # varianza de la VD <code>var(pre)</code> # varianza explicada <code>var(res)</code> # varianza no explicada
Tablas de contingencia (frecuencias absolutas)	<code>tc <- ftable(practica\$genero, practica\$fumar)</code>
Añadir marginales	<code>addmargins(tc)</code>
Tablas de contingencia (frecuencias relativas conjuntas)	<code>prop.table(tc)</code>
Tablas de contingencia (frecuencias relativas condicionales por filas)	<code>prop.table(tc, 1)</code>
Tablas de contingencia (frecuencias relativas condicionales por columnas)	<code>prop.table(tc, 2)</code>
Gráfico de barras para Tabla de Contingencia	<code>barplot(tc, main = "Grafica Tabaquismo y Genero", xlab = "Mujer: negro, Varon: gris", ylab = "frecuencias")</code>

Nota: todos los ejemplos de la tabla hacen referencia al archivo *practica.sav* del anexo 1.

Anexo 4. Tabla resumen comandos R (Probabilidad)

Concepto	Comando
Definir Binomial Ejemplo $X \sim B(12; 0,20)$	<code>Xb <- seq(0, 12, by = 1)</code> <code>fb <- dbinom(Xb, 12, .20)</code> <code>barplot(fb)</code>
Función de probabilidad para una Binomial	<code>(dbinom(0, 12, .20))</code> # f(0) en una B(12; 0,20)
Función de distribución para una Binomial	<code>(pbinom(5, 12, .20))</code> # F(5) en una B(12; 0,20)
Área derecha para una Binomial	<code>(pbinom(5, 12, .20, lower.tail=FALSE))</code> # 1-F(5)
Área comprendida entre 2 valores en una Binomial	<code>sum(dbinom(2:4, 12, .20))</code>
Valor de x para una Binomial conocida $F(x) = p$	<code>(qbinom(p, 12, .20))</code> # Valor de X que corresponde al $F(x) = p$ en una B(12; 0,20)
Definir Normal Función de densidad y representación gráfica de la Campana de Gauss	<code>x <- seq(-3, 3, by=0.01)</code> <code>f <- dnorm(x)</code> <code>plot(x, f, type="l", lwd=1.5)</code>
Función de densidad de probabilidad para una $N(100; 15)$	<code>x <- seq(70, 130, by=0.1)</code> # Generar valores entre 70 y 130 en intervalos de 0.1 puntos <code>f1 <- dnorm(x, mean=100, sd=15)</code> # Función de densidad
Función de distribución para una z de una Normal	<code>pnorm(z)</code> # requiere definir antes la z También puede hacerse con: <code>pnorm(X, μ, σ)</code> # se pone directamente la X_i , la μ y la σ
Área derecha para una z de una Normal	<code>pnorm(z, lower.tail=FALSE)</code>
Valor de la z según $F(x) = p$ en una Normal	<code>qnorm(p)</code> # z para $F(x) = p$ <code>qnorm(p, lower.tail=FALSE)</code> # z para $1 - F(x) = p$
Definir una Chi-cuadrado con k grados de libertad Función de densidad y representación gráfica para una χ^2_{10}	<code>x2 <- seq(0, 60, by=0.1)</code> <code>f1c <- dchisq(x2, 10)</code> <code>plot(x2, f1c, type="l", lwd=1.5, ylim=c(0, 0.10), ylab="densidad")</code>
Función de distribución para $\chi^2_{10} = 3,94$	<code>pchisq(3.94, 10)</code>
Área derecha para $\chi^2_{10} = 3,94$	<code>pchisq(3.94, 10, lower.tail=FALSE)</code>
Valor de la χ^2_k según una $F(x) = p$	<code>qchisq(p, k)</code>
Definir una t de Student con k grados de libertad Función de densidad y representación gráfica para una t_{10}	<code>T <- seq(-3, 3, by=0.1)</code> <code>f1t <- dt(T, 10)</code> <code>plot(T, f1t, type="l", lwd=1.5, ylim=c(0, 0.40), ylab="densidad")</code>
Función de distribución para $t_{10} = 2$	<code>pt(2, 10)</code>
Área derecha para $t_{10} = 2$	<code>pt(2, 10, lower.tail=FALSE)</code>
Valor de la t_k según una $F(x) = p$	<code>qt(p, k)</code>
Definir F de Snedecor con m y n grados de libertad Función de densidad y representación gráfica para una $F_{5,15}$	<code>F <- seq(0, 5, by=0.1)</code> <code>f1f <- df(F, 5, 15)</code> <code>plot(F, f1f, type="l", lwd=1.5, ylim=c(0, 0.99), ylab="densidad")</code>
Función de distribución para $F_{10,15} = 2,544$	<code>pf(2.544, 10, 15)</code>
Área derecha para $F_{10,15} = 2,544$	<code>pf(2.544, 10, 15, lower.tail=FALSE)</code>
Valor de la $F_{m,n}$ según una $F(x) = p$	<code>qf(p, 10, 15)</code>

Anexo 5. Archivo de datos *terapia.dat*

La siguiente tabla muestra el contenido del archivo de datos *terapia.dat*, utilizado en distintos capítulos de libro para ejemplificar los análisis inferenciales:²

<i>n</i>	noche 1	noche 2	noche 3	sexo	edad	terapia1	terapia2	terapia3	horas
1	4.0	4.5	5.0	0	1	0	1	0	13.5
2	5.8	6.4	7.0	1	2	0	1	0	19.2
3	4.1	4.9	5.1	0	3	0	0	0	14.1
4	4.3	5.0	5.6	1	1	0	1	1	14.9
5	5.4	6.1	6.3	0	2	0	0	1	17.8
6	5.7	6.3	6.8	1	3	1	0	0	18.8
7	4.3	4.8	5.4	0	1	0	1	1	14.5
8	7.6	7.9	8.6	1	2	1	0	1	24.1
9	5.7	6.0	6.7	0	3	1	1	0	18.4
10	3.8	4.6	4.9	1	1	0	1	1	13.3
11	7.5	8.0	8.5	0	2	1	0	0	24.0
12	6.4	7.0	7.3	1	3	1	1	0	20.7
13	4.0	4.5	5.0	0	1	0	1	1	13.5
14	7.7	8.1	8.5	1	2	1	1	1	24.3
15	6.4	6.9	7.3	0	3	1	0	1	20.6
16	5.9	6.6	7.0	1	1	1	0	1	19.5
17	7.2	7.8	8.1	0	2	1	1	1	23.1
18	5.6	6.2	6.8	1	3	1	1	0	18.6
19	5.7	6.1	6.8	0	1	1	0	0	18.6
20	7.4	7.6	8.2	1	2	1	1	1	23.2

La primera línea del archivo contiene los nombres de las variables. Los valores de las variables están de la segunda línea en adelante. El archivo *terapia.dat* incluye los datos de 20 sujetos en 10 variables relacionadas con el insomnio. Más concretamente:

- Tres variables relacionadas con las horas dormidas en tres noches consecutivas (*noche1*, *noche2* y *noche3*).
- Tres variables relacionadas con las terapias que ha recibido el sujeto (*terapia1*, que se refiere a la terapia contra el *insomnio*, *terapia2*, que indica si el sujeto ha recibido terapia contra estados de *ansiedad* generalizada, y *terapia3*, que indica si ha recibido terapia contra algún tipo de *fobia*) que toman los valores 1, si el sujeto ha recibido la terapia, y 0 en caso contrario.
- La variable *horas* que se calcula como la suma de *noche1*, *noche2* y *noche3*.
- La variable *sexo*, que toma los valores 0 para las mujeres y 1 para los hombres.
- La variable *edad*, con los valores 1 para los sujetos menores de 20 años, 2 para los sujetos entre 20 y 25 años, y 3 para los sujetos con más de 25 años.

² El archivo *terapia.dat* puede bajarse desde: <https://www.psicologiauam.es/carmen.ximenez/Practicas1.html>

El archivo *terapia.dat* puede leerse utilizando el comando:

```
datos <- read.table("terapia.dat", header=TRUE)
```

Hemos convertido el fichero *terapia.dat* en el objeto **datos**, porque *R* trabaja con objetos.

Alternativamente, pueden introducirse los datos en *R* ejecutando el siguiente comando:

```
datos <- data.frame(  
  "n" = 1:20,  
  "noche1" = c(4.0,5.8,4.1,4.3,5.4,5.7,4.3,7.6,5.7,3.8,7.5,6.4,4.0,7.7,  
               6.4,5.9,7.2,5.6,5.7,7.4),  
  "noche2" = c(4.5,6.4,4.9,5.0,6.1,6.3,4.8,7.9,6.0,4.6,8.0,7.0,4.5,8.1,  
               6.9,6.6,7.8,6.2,6.1,7.6),  
  "noche3" = c(5.0,7.0,5.1,5.6,6.3,6.8,5.4,8.6,6.7,4.9,8.5,7.3,5.0,8.5,  
               7.3,7.0,8.1,6.8,6.8,8.2),  
  "sexo" = c(0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1),  
  "edad" = c(1,2,3,1,2,3,1,2,3,1,2,3,1,2,3,1,2,3,1,2),  
  "terapia1" = c(0,0,0,0,0,1,0,1,1,0,1,1,0,1,1,1,1,1,1,1),  
  "terapia2" = c(1,1,0,1,0,0,1,0,1,1,0,1,1,1,0,0,1,1,0,1),  
  "terapia3" = c(0,0,0,1,1,0,1,1,0,1,0,0,1,1,1,1,1,0,0,1),  
  "horas" = c(13.5,19.2,14.1,14.9,17.8,18.8,14.5,24.1,18.4,13.3,24.0,20.7,  
              13.5,24.3,20.6,19.5,23.1,18.6,18.6,23.2)  
)  
attach(datos)
```


Anexo 6. Tabla resumen comandos R (inferencia estadística)

1. Contrastes sobre medias

Concepto	Comando
Contraste sobre una media. Prueba Z con $\sigma = 2$ Contraste $H_0: \mu \geq 7$ frente a $H_1: \mu < 7$ utilizando el nivel crítico (p)	<code>m0 <- 7</code> <code>media <- mean(datos\$noche1)</code> <code>n <- sum(!is.na(datos\$terapia1))</code> <code>sigma <- 2</code> <code>z <- sqrt(n)*(media - m0)/sigma</code> <code>p <- pnorm(z)</code>
Contraste sobre una media. Prueba T Bilateral Prueba T Unilateral izquierdo Prueba T Unilateral derecho	<code>t.test(datos\$noche1, mu=6)</code> <code>t.test(datos\$noche1, mu=6, conf.level=0.99)</code> <code>t.test(datos\$noche1, mu=6, alternative="less")</code> <code>t.test(datos\$noche1, mu=6, alternative="greater")</code>
Contraste sobre dos varianzas. Prueba F	<code>var.test(datos\$noche1~datos\$sexo)</code>
Contraste sobre dos medias independientes (asumiendo varianzas iguales)	<code>t.test(datos\$noche1~datos\$sexo, var.equal=TRUE)</code>
Contraste sobre dos medias relacionadas Prueba T Bilateral Prueba T Unilateral izquierdo Prueba T Unilateral derecho	<code>t.test(datos\$noche1, datos\$noche2, paired=T)</code> <code>t.test(datos\$noche1, datos\$noche2, paired=T, alternative="less")</code> <code>t.test(datos\$noche1, datos\$noche2, paired=T, alternative="greater")</code>

2. Contrastes sobre proporciones y tablas de contingencia

Concepto	Comando
Prueba binomial ($\pi = 0,50$)	<code>b2 <- datos\$terapia2</code> <code>binom.test(sum(b2), length(b2), 0.50)</code>
Aproximación normal a la binomial	<code>prop.test(sum(b2), length(b2), 0.50)</code>
Bondad de ajuste	<code>tab <- table(datos\$edad)</code> <code>gof <- chisq.test(tab)</code>
Contraste de independencia	<code>chisq.test(datos\$terapia1, datos\$terapia2)</code>
Homogeneidad marginal (McNemar)	<code>mcnemar.test(datos\$terapia1, datos\$terapia2)</code>

3. Contrastes sobre correlación y regresión lineal

Concepto	Comando
Correlación	<code>cor(datos\$terapia1, datos\$terapia2, use="complete.obs")</code>
Contraste de correlación	<code>cor.test(datos\$terapia1, datos\$terapia2)</code>
Regresión lineal simple	<code>reg <- lm(datos\$noche1~datos\$edad)</code> <code>summary(reg)</code> <code>plot(datos\$edad, datos\$noche1)</code> <code>abline(reg)</code>
Regresión lineal múltiple	<code>regm <- lm(datos\$noche1~datos\$edad+datos\$sexo)</code> <code>summary(regm)</code> <code>anova(regm)</code>

4. ANOVA

Concepto	Comando
ANOVA de un factor (diseño inter-sujetos)	<pre> edad <- factor(datos\$edad_rec) boxplot(datos\$terapia2~edad) anova <- aov(datos\$terapia2~edad) summary(anova) coef(anova) plot(anova) model.tables(anova,"means") </pre>
Tamaño del efecto, η^2	<pre> library(lsr) etaSquared(anova) </pre>
Comparaciones múltiples Tukey Planeadas o a-priori $H_{0(1)}: -\mu_1 - \mu_2 + 2\mu_3 = 0$ $H_{0(2)}: \mu_1 - \mu_2 = 0$	<pre> TukeyHSD(anova) H <- cbind(c(-1, -1, 2), c(1, -1, 0)) contrasts(edad) <- H anova <- aov(horas~edad) fit_planned <- lm(anova) summary(anova_planned) </pre>
ANOVA de dos factores	<pre> fsexo <- factor(datos\$fsexo) anova2 <- aov(datos\$terapia2~edad*fsexo) model.tables(anova2,"means") summary(anova2) </pre>
Tukey, ANOVA de dos factores	<pre> TukeyHSD(anova2) </pre>
ANCOVA	<pre> ancova <- aov(horas~fsexo+edad) summary(ancova) </pre>

5. Contrastes no paramétricos

Concepto	Comando
Test Kolmogorov-Smirnov	<pre> ks.test(horas, "pnorm") </pre>
Prueba de los signos (binomial)	<pre> noche2D <- noche2 > 7.5 binom.test(x=sum(noches2D), n=length(noches2D), p=0.5) </pre>
Prueba de las rachas	<pre> library(randtests) runs.test(horas,plot=T) </pre>
Prueba de Mann-Whitney	<pre> wilcox.test(noches2 ~ terapia1) muestras independientes wilcox.test(noches1, noche2, paired=T) muestras relacionadas </pre>
Test de Kruskal-Wallis	<pre> kruskal.test(noches3 ~ edad) </pre>
Prueba de Friedman	<pre> friedman.test(n123[, "nhoras"], n123[, "noche"], n123[, "n"]) </pre>

Nota: todos los ejemplos de las tablas del anexo 6 hacen referencia al archivo *terapia.dat* (ver anexo 5).

Anexo 7. Librerías de R más usadas en el Análisis de datos en Psicología

Librería	Descripción	Enlace
corrplot	Elabora gráficos de correlaciones	https://cran.r-project.org/web/packages/corrplot/index.html
DescTools	Realiza análisis descriptivos y calcula los índices de asociación C y V para tablas de contingencia	https://cran.r-project.org/web/packages/DescTools/index.html
effectsize	Calcula las medidas de tamaño del efecto (η^2 , ε^2 y ω^2)	https://cran.r-project.org/web/packages/effectsize/index.html
epitools	Realiza análisis epidemiológicos (aquí la hemos usado para calcular el Índice de riesgo y Razón de ventajas)	https://cran.r-project.org/web/packages/epitools/index.html
foreign	Lee archivos SPSS (.sav)	https://cran.r-project.org/web/packages/foreign/index.html
haven	Lee archivos SPSS (.sav) y de texto (.dat)	https://cran.r-project.org/web/packages/haven/index.html
lavaan	No lo hemos visto pero se usa mucho en análisis multivariante (por ejemplo, análisis factorial y modelos SEM)	https://cran.r-project.org/web/packages/lavaan/index.html
mirt	Se aprende a usar en la asignatura de tercer curso <i>Introducción a la Psicometría</i> , sirve para trabajar con modelos de TRI	https://cran.r-project.org/web/packages/mirt/index.html
modeest	Calcula la moda	https://cran.r-project.org/web/packages/modeest/index.html
moments	Calcula los estadísticos de asimetría y curtosis	https://cran.r-project.org/web/packages/moments/index.html
PerformanceAnalytics	Es útil para econometría. Aquí lo hemos usado para elaborar gráficos de dispersión y correlaciones juntos (ver capítulo 6)	https://cran.r-project.org/web/packages/PerformanceAnalytics/index.html
Psych	Se verá en la asignatura de <i>Introducción a la Psicometría</i> , es una librería básica para llevar a cabo análisis psicométricos (fiabilidad, validez, AFE, etc.)	https://cran.r-project.org/web/packages/psych/index.html
randtests	Realiza la <i>Prueba de las Rachas</i> (ver capítulo 9)	https://cran.r-project.org/web/packages/randtests/index.html
xlsx	Lee archivos Excel	https://cran.r-project.org/web/packages/xlsx/index.html

