

NICHOLAS J. HORTON, DANIEL KAPLAN, AND
RANDALL PRUIM

UNA GUÍA DE ESTUDIAN- TES PARA R

La traducción al español de este libro fue realizada por Francisco Javier Jara Ávila. En caso de encontrar errores en la misma, siéntase libre de escribir a fjaraavila@gmail.com

Copyright (c) 2015 by Nicholas J. Horton, Randall Pruim, & Daniel Kaplan.

Edition 1.2, November 2015

This material is copyrighted by the authors under a Creative Commons Attribution 3.0 Unported License. You are free to *Share* (to copy, distribute and transmit the work) and to *Remix* (to adapt the work) if you attribute our work. More detailed information about the licensing is available at this web page: <http://www.mosaic-web.org/go/teachingRlicense.html>.

Cover Photo: Maya Hanna.

Índice general

<i>1</i>	<i>Introducción</i>	<i>13</i>
<i>2</i>	<i>Empezando con RStudio</i>	<i>15</i>
<i>3</i>	<i>Una variable cuantitativa</i>	<i>27</i>
<i>4</i>	<i>Una variable categóricas</i>	<i>39</i>
<i>5</i>	<i>Dos variables cuantitativas</i>	<i>45</i>
<i>6</i>	<i>Dos variables categóricas</i>	<i>55</i>
<i>7</i>	<i>Respuesta cuantitativa, predictor categórico</i>	<i>61</i>
<i>8</i>	<i>Respuesta categórica, predicto cuantitativo</i>	<i>71</i>
<i>9</i>	<i>Resulados de análisis de supervivencia</i>	<i>75</i>

<i>10</i>	<i>Más de dos variables</i>	79
<i>11</i>	<i>Distribuciones de probabilidad & variables aleatorias</i>	87
<i>12</i>	<i>Cálculos de Potencia</i>	95
<i>13</i>	<i>Manejo de datos</i>	99
<i>14</i>	<i>Health Evaluation (HELP) Study</i>	111
<i>15</i>	<i>Ejercicios y problemas</i>	115
<i>16</i>	<i>Bibliografía</i>	119
<i>17</i>	<i>Índice alfabético</i>	121

Acerca de estas notas

Aquí presentamos un abordaje a la enseñanza introductoria e intermedia en cursos de estadística, la cual está acoplada importantemente con la computación general, y con R y RStudio en particular. Estas actividades y ejemplos tienen la intención de resaltar el abordaje moderno de la enseñanza estadística que se concentra en el modelaje, la inferencia basada en el re-muestreo y las técnicas de gráficos multivariados. Una meta secundaria es facilitar los cálculos con datos mediante el uso de pequeños estudios de simulaciones y el flujo de trabajo de análisis estadístico apropiado. Esto sigue la filosofía descrita por Nolan y Temple Lang¹. La importancia de la computación moderna en la educación estadística consiste en que esta es un componente principal de los lineamientos de currículo recientemente adoptados de la Asociación Americana de Estadística².

En este libro (y en sus volúmenes acompañantes), introduciremos múltiples actividades, algunas apropiadas para un curso introductorio, otras maleables para niveles superiores, que demuestran conceptos claves de estadística y modelaje, mientras también ofrecen ayuda al material de base de cursos más tradicionales.

Un trabajo en progreso

Estos materiales fueron contruidos para un taller llamado *Teaching Statistics Using R* (Enseñando estadística con R) anterior a la United States Conference on Teaching Statistics y fue corregido para USCOTS 2011, USCOTS 2013, eCOTS 2014, ICOTS 9, and USCOTS 2015.

Organizamos todos estos talleres para ayudar a los instructores a integrar R (así como otras tecnologías relacionadas) en los cursos de estadística de todos los niveles.

Recibimos una gran retroalimentación y bastantes buenas

¹ D. Nolan and D. Temple Lang. Computing in the statistics curriculum. *The American Statistician*, 64(2):97–107, 2010

² ASA Undergraduate Guidelines Workgroup. 2014 curriculum guidelines for undergraduate programs in statistical science. Technical report, American Statistical Association, November 2014. <http://www.amstat.org/education/curriculumguidelines.cfm>

PRECAUCIÓN!

A pesar de nuestros mejores esfuerzos, VA a encontrar fallas en este documento y en nuestro código. Por favor háganos saber cuándo las encuentre para poder resolverlas..

ideas por parte de participantes y por parte de aquellos con los que compartimos este material en los talleres.

Considere las notas como un trabajo en progreso. Apreciaremos cualquier retroalimentación que esté dispuesto a compartírnos, conforme continuamos el trabajo en estos materiales y el acompañante del material, el paquete `mosaic`. Envíenos un email a pis@mosaic-web.org con cualquier comentario, sugerencia, corrección, etc

Versiones actualizadas del paquete serán publicadas en <http://mosaic-web.org>.

Dos audiencias

Inicialmente desarrollamos estos materiales para instructores y profesores de estadística a nivel universitario. Otra audiencia para la cual desarrollamos el material son los estudiantes a quién estos instructores o profesores enseñan. Algunas de las secciones, algunos de los ejemplos y algunos ejercicios están pensados para una u otra de estas audiencias, con una mayor claridad en el énfasis que se le da a la audiencia. Esto significa que:

1. Algunos materiales son utilizados esencialmente para los estudiantes.
2. Algunos materiales apuntan a equipar al instructor a explotar sus propias capacidades en R and RStudio para desarrollar su propio material de enseñanza.

Aunque la distinción puede volverse confusa, y lo que funciona ^{así}.en una condición puede no funcionar ^{así}.en otras, intentaremos indicarle cuales partes se ajustan a cual audiencia conforme vayamos por ella.

R, RStudio y los paquetes de R

R puede ser obtenido <http://cran.r-project.org/>. La descarga e instalación es abierta y sencilla para máquinas con MAC, PC o linux

RStudio es una interfaz de desarrollo integrada (IDE por sus siglas en inglés) que facilita el uso de R tanto para usuarios sin experiencia como usuarios expertos. Lo hemos adaptado como nuestra interfaz standard de enseñanza puesto que simplifica drásticamente el uso de R para instructores y estudiantes.

MÁS INFO

Algunas cosas sólo pueden ser ejecutadas en RStudio, como la función `manipulate()` y la herramienta de RStudio para la investigación reproducible.

RStudio está disponible en <http://www.rstudio.org/>.

RStudio . RStudio puede instalarse como aplicación de escritorio o como una aplicación en un servidor accesible a los usuarios vía Internet.

Además de R y RStudio haremos uso de algunos paquetes que necesitan ser instalados y cargados separadamente. El paquete `mosaic` (y sus dependencias) serán utilizados en todo momento. Otros paquetes aparecerán esporádicamente.

Notas marginales

Las notas marginales aparecerán por aquí y por allá. En ocasiones estas son comentarios extra que deseábamos hacer, pero no queríamos interrumpir el flujo de la lectura y mencionarlos en el texto principal. Otras proveen consejos para la enseñanza o precauciones acerca de trampas, caídas y 'jugadas'.

Lo que es nuestro, es suyo, hasta cierto punto

Este material se encuentra en copyright por los autores, bajo 'Creative Commons Attribution 3.0 Unported License'. Está libre para *compartirlo* (copiarlo, distribuirlo y transferir el trabajo) y *mezclarlo* (adaptar el trabajo) si nos atribuye el trabajo realizado. Para información más detallada sobre esto, está disponible en el siguiente sitio web: <http://www.mosaic-web.org/go/teachingRlicense.html>.

Creación del documento

El documento original fue creado el 12 de Agosto de 2016, utilizando: knitr, versión 1.13.2 mosaic, versión 0.14.9000 *R version 3.3.0 (2016-05-03)

Inevitablemente, cada una de estas va ser actualizada en ciertas ocasiones. Si usted encuentra cosas que se ven diferente en su computadora, asegúrese que su versión de R y sus paquetes estén actualizados y revise versiones más nuevas de este documento

Honores y agradecimientos a Joseph Cappelleri por sus comentarios útiles en los primeros borradores de este material

chapter*Proyecto MOSAIC Este libro es producto del Proyecto MOSAIC, una comunidad de educadores trabajan-

La versión en un servidor web de RStudio funciona bien con estudiantes que están iniciando. Lo único que se necesita es un navegador web, evadiendo todos los potenciales problemas con particularidades de la computadora personal de cada estudiante.

¿Tiene alguna sugerencia para una nota marginal? Compártala con nosotros.

CAVANDO HONDO

Si usted conoce LaTeX tan bien como R, entonces el paquete knitr ofrece una buena solución para mezclar ambos. Nosotros utilizamos este sistema para producir este libro. También lo utilizamos para nuestra propia investigación y para introducir a estudiantes de nivel superior a métodos de análisis reproducible. Para principiantes, introducimos knitr con RMarkdown, el cual produce archivos en formato PDF, HTML o Word utilizando sintaxis sencilla

do para desarrollar nuevas formas de introducir las matemáticas, la estadística, la computación y el modelaje de datos a estudiantes universitarios.

La meta del proyecto MOSAIC es ayudar a compartir ideas y recursos para mejorar la enseñanza, y desarrollar una estructura curricular y valorativa para dar apoyo a la diseminación y evaluación de estos esfuerzos. Nuestra meta es proveer un amplio acercamiento a los estudios cuantitativos que brindan una mejor herramienta al trabajo en la ciencia y tecnología. El proyecto resalta e integra diversos aspectos del trabajo cuantitativo que los estudiantes de ciencia, tecnología e ingeniería necesitarán en sus vidas profesionales, pero que actualmente es usual que sean enseñados de forma aislada, si es que acaso se enseñan. En particular, nos concentramos en:

En particular nos enfocamos en:

Modelaje La habilidad de crear, manipular e investigar representaciones matemáticas útiles e informativas de situaciones del mundo real.

Estadística El análisis de variabilidad que se esboza de nuestra habilidad de cuantificar la incertidumbre y generar inferencias lógicas de observación y experimentación.

Computación La capacidad de pensar algorítmicamente, de manejar datos en gran escala, visualizar e interactuar con modelos, y automatizar tareas para la eficiencia, precisión y reproducibilidad.

Cálculo El punto de entrada tradicional para estudiantes universitarios y un tema que todavía hoy tiene la capacidad de dotar de nociones importantes a los estudiantes

Tomando el apoyo de la Fundación Nacional de Ciencia de Estados Unidos (NSF DUE-0920350), Proyecto MOSAIC apoya algunas iniciativas para ayudar a lograr estas metas, incluyendo:

Desarrollo de facultades y oportunidad de entrenamiento así como en USCOTS 2011, USCOTS 2013, eCOTS 2014, e ICOTS9 en los talleres *Teaching Statistics Using R y RStudio*, en 2010 nuestro Proyecto MOSAIC inició talleres en el Instituto de Matemática y sus aplicaciones, y

nuestro taller *Modeling: Early and Often in Undergraduate Calculus* AMS PREP fueron ofrecidos en 2012, 2013, and 2015.

M-casts, una serie de seminarios web programados regularmente, dados vía Internet, en el que se creó foro para que los instructores puedan compartir sus observaciones e innovaciones y desarrollar colaboraciones para refinar y desarrollar las mismas. Las grabaciones de los M-casts están disponibles en el sitio web del Proyecto MOSAIC, <http://mosaic-web.org>.

La construcción de un programa y materiales para cursos que enseña MOSAIC con tópicos integrados de la mejor manera. Estos cursos y materiales podrían ser completamente nuevas construcciones, o podrían ser modificaciones de recursos existentes que se aproximen a las conexiones entre los tópicos de MOSAIC.

Más detalles pueden ser encontrados en <http://www.mosaic-web.org>. Le damos la bienvenida e incitamos su participación en todas estas iniciativas.

Estadística computacional

Hay al menos dos formas en las cuales se puede introducir un software estadístico en un curso de estadística. En el primer abordaje, el curso debe ser enseñado esencialmente como se hacía anterior a la utilización de los software estadísticos, pero utilizando la computadora para hacer más rápidos algunos cálculos y generar gráficos de alta calidad displays. Probablemente el tamaño de los conjuntos de datos también deba ser incrementado. Nos referiremos a este enfoque como **computación estadística** puesto que la computadora funciona primariamente como una herramienta de cálculo para reemplazar los cálculos en lápiz y papel, además del dibujo manual de los gráficos.

En el segundo abordaje, cambios más importantes en el curso resultan de la introducción de la computadora. Algunos temas nuevos son cubiertos, algunos temas anteriores son omitidos. Algunos temas anteriores son tratados en formas muy diferentes, y quizás en diferentes etapas en el curso. A este abordaje nos referiremos como **estadística computacional** porque la disponibilidad de la computación está moldeando como la estadística es construida y enseñada. La estadística computacional es un componente clave de la **ciencia de los datos**, definida como la habilidad de usar datos para responder preguntas y comunicar resultados.

En la práctica, la mayoría de los cursos van a incorporar elementos de ambas, la estadística computacional y la computación estadística, pero las proporciones relativas pueden diferir drásticamente de curso a curso. En qué parte del espectro se encuentra el curso va depender de muchos factores, incluyendo las metas del curso, la disponibilidad de tecnología para el uso de los estudiantes, la perspectiva del libro utilizado y el nivel de 'comfort' que sienta el instructor con la estadística y la computación.

Los estudiantes necesitan ver los aspectos de la computación y la ciencia de los datos de manera pronta y cotidiana para desarrollar habilidades profundas. Establecer bases en cursos introductorios ayuda a que inicien de mejor manera.

Entre los varios paquetes de software estadístico disponible, R está incrementando su popularidad. La reciente adición de RStudio ha hecho R más poderoso y accesible. Como R y RStudio son gratis, ambos se han vuelto ampliamente utilizados en la investigación y la industria. Entrenar en R y RStudio es normalmente visto como una habilidad adicional importante que un curso de estadística puede desarrollar. Por esto, un alto número de instructores están utilizando R para su propio trabajo estadístico; por lo que es natural que inicien a integrarlo en la enseñanza también. Al mismo tiempo, el desarrollo de R y RStudio (una interfaz opcional e integrada al desarrollo de un entorno para R) están haciendo cada vez más fácil iniciar la utilización de R.

Desarrollamos el paquete de R `mosaic` (disponible en CRAN), para hacer ciertos aspectos de la computación estadística y la estadística computacional más simple para inexpertos, sin limitar su habilidad a utilizar características del lenguaje. El paquete `mosaic` incluye un acercamiento al modelaje, que utiliza la misma sintaxis general para calcular estadísticas descriptivas, crear gráficos y ajustar modelos lineales.

La información sobre el paquete `mosaic`, incluyendo ejemplos de características y material suplementario (así como este libro) puede encontrarla en <https://cran.r-project.org/web/packages/mosaic>.

1

Introducción

En este libro de referencia, resumidamente repasamos comandos y funciones necesarias para analizar datos desde cursos introductorios hasta secundarios de estadística . Esto con la intención de complementar los libros *Empiece a enseñar con R* and *Start Modeling with R* books.

La mayoría de nuestros ejemplos van a utilizar datos del estudio HELP (Health Evaluation and Linkage to Primary Care): un ensayo clínico aleatorizado buscando conectar pacientes en riesgo con el cuidado primario de salud. Más información del conjunto de datos se puede encontrar en el capítulo 14.

Como la selección y el orden de los temas pueden variar de libro a libro y de instructor a instructor, hemos decidido organizar este material por tipo de datos que están siendo analizados. Esto debe hacerse dirigido a encontrar lo que se está buscando. Algunas habilidades en manejo de datos pueden ser necesitadas por los estudiantes¹. Una introducción básica a expresiones clave se da en el capítulo 13. 13.

Este proyecto se aventura de las iniciativas tomadas por el Proyecto MOSAIC (<http://www.mosaic-web.org>), un esfuerzo financiado por la NSF para mejorar la enseñanza de la estadística, el cálculo, la ciencia y la computación en el currículo de los programas de grado. En particular, utilizamos el paquete `mosaic`, el cual fue escrito para simplificar el uso de R para los cursos introductorios de estadística, y el paquete `mosaicData` que incluye algunos archivos de datos. Un pequeño resumen de los comandos de R necesarios para la enseñanza de estadística introductoria puede ser encontrado en el siguiente sitio: <https://cran.r-project.org/web/packages/mosaic>.

Otros recursos relacionados del Proyecto MOSAIC pueden ser de ayuda, incluyendo un conjunto de ejemplos ano-

¹ N.J. Horton, B.S. Baumer, and H. Wickham. Setting the stage for data science: integration of data management skills in introductory and second courses in statistics (<http://arxiv.org/abs/1401.3269>). *CHANCE*, 28(2):40–50, 2015

tados de la quinta edición de Moore, McCabe y Craig *Introduction to the Practice of Statistics*² (ver <http://www.amherst.edu/~nhorton/ips6e>), la segunda y tercera edición de *Statistical Sleuth*³ (ver <http://www.amherst.edu/~nhorton/sleuth>), and *Statistics: Unlocking the Power of Data* por Lock et al (see <https://github.com/rpruim/Lock5withR>).

Para usar el paquete dentro de R, este debe ser instalado (una vez), y cargado (cada sesión). El paquete `mosaic` puede ser instalado utilizando el siguiente comando:

```
> install.packages("mosaic") # Observe las comillas
```

El `#` es un comentario en R, y todo el texto después de ese símbolo es ignorado a la hora de ejecutar el código.

O Una vez que el paquete es instalado (una única vez), puede ser cargado ejecutando el siguiente comando:

```
> require(mosaic)
```

El sistema de RMarkdown aporta una adición en el lenguaje y produce documentos en PDF, Word o HTML. Esto permite a los estudiantes tomar sus análisis utilizando un flujo de trabajo que facilita la reproducibilidadz evita copiar y pegar errores.

Usualmente introducimos a los estudiantes a RMarkdown de forma muy pronta, pidiéndoles que lo utilicen para tareas y reportes⁴.

² D. S. Moore and G. P. McCabe. *Introduction to the Practice of Statistics*. W.H.Freeman and Company, 6th edition, 2007

³ F. Ramsey and D. Schafer. *Statistical Sleuth: A Course in Methods of Data Analysis*. Cengage, 2nd edition, 2002

RStudio dispone forma más simplificada de instalación de paquetes (en el panel de arriba a la derecha).

El sistema `knitr`/ \LaTeX permite combinar R y \LaTeX en el mismo documento. La recompensa por aprender este sistema más complicado es un control más fino sobre el formato del documento de salida. Pero RMarkdown es más sencillo de aprender y adecuar incluso en el trabajo a nivel profesional.

Usar Markdown o `knitr`/ \LaTeX requiere que el paquete `markdown` esté instalado.

⁴ B.S. Baumer, M. Çetinkaya Rundel, A. Bray, L. Loi, and N. J. Horton. R Markdown: Integrating a reproducible analysis tool into introductory statistics. *Technology Innovations in Statistics Education*, 8(1):281–283, 2014

2

Empezando con RStudio

RStudio es un entorno integrado de desarrollo (IDE por sus siglas en inglés) para R que proporciona una alternativa de interface de R, que tiene algunas ventajas sobre la interfaz predeterminada de R:

- RStudio es ejecutable en máquinas Mac, PC o Linux y brinda una interfaz simplificada que *se ve y se siente idéntico en todas ellas*. La interface predeterminada de R es un poco diferente en las distintas plataformas. Esto es un distractor para los estudiantes y agrega una responsabilidad extra de conocimiento al instructor

- RStudio puede ejecutarse desde un navegador web.

Además de la versión para escritorio, RStudio puede ser utilizado y configurado como una aplicación en un servidor al que se tiene acceso vía internet

La interfaz web es prácticamente idéntica a la versión de escritorio. Como con otros servicios web, el usuario debe hacer un inicio de sesión para acceder a su cuenta. Si los estudiantes cierran sesión y abren sesión después, incluso en una máquina diferente, la sesión es restaurada y pueden continuar sus análisis justo donde los dejaron. Con un poco de ajustes avanzados, los instructores pueden salvar el registro de su utilización de R classroom R use and students en clase y pueden cargar estos archivos de registro en su propio entorno.

- RStudio brinda apoyo para la investigación reproducible.. RStudio facilita incluir texto, análisis estadístico (código de R y salidas de R), y gráficos, todo en un mismo documento. El sistema de RMarkdown proporciona una adición en el lenguaje y produce documentos en

Una serie de videos para iniciar con R está disponible en: <http://www.amherst.edu/~nhorton/rstudio>.

PRECAUCIÓN!

La versión de escritorio y la versión del servidor de RStudio son tan similares tendrá que prestar mucha atención y asegurarse de que este trabajando en la que tenía intención de trabajar.

NOTA

Usar RStudio en un navegador es como Facebook para estadística. Cada vez que el usuario vuelve, la sesión previa es restaurada y puede continuar el trabajo donde lo dejó. Los usuarios pueden iniciar sesión desde cualquier dispositivo con acceso a internet.

HTML. El sistema `knitr`/ \LaTeX permite a los usuarios combinar R y \LaTeX en el mismo documento. La recompensa por aprender este sistema más complicado es un control más fino sobre el formato del documento de salida. Dependiendo del nivel del curso, los estudiantes pueden usar esto para tareas y proyectos.

- RStudio ofrece una opción integrada para editar y ejecutar código de R y documentos.
- RStudio provee una interfaz funcional de gráficos para el usuario.

RStudio no es un GUI para R pero este sí brinda una GUI que simplifica cosas como instalar y actualizar los paquetes; monitorear, guardar y cargar entornos; importar y exportar datos; navegar y exportar gráficos; y navegar por archivos y documentación.

- RStudio permite el acceso al paquete `manipulate`

El paquete `manipulate` brinda una forma de crear aplicaciones gráficas interactivas de forma rápida y sencilla.

Realmente se puede usar R sin usar RStudio, sin embargo, RStudio hace una cantidad considerable de cosas más sencillas, recomendamos vigorosamente la utilización de RStudio. Además puesto que RStudio está en desarrollo activo, tenemos la expectativa de más características útiles en el futuro.

Primordialmente utilizamos una versión online de RStudio. RStudio es una interfaz innovadora y poderosa para R que se ejecuta en un navegador web o en su computadora. Ejecutarlo en el navegador tiene la ventaja de que no se necesita instalar o configurar nada. Solo inicia sesión y está listo. Entonces, RStudio va recordar lo que estuvo haciendo cada vez que inicie sesión (incluso en una máquina diferente), usted puede elegir iniciar justo donde usted lo dejó. Esto es "R en la nube" funciona un poco como GoogleDocs o Facebook para R.

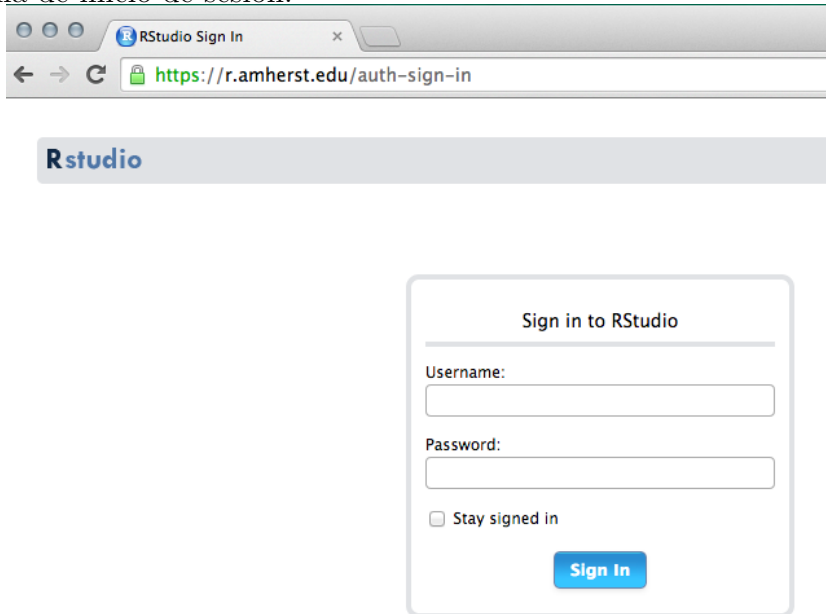
R se puede obtener también de <http://cran.r-project.org/>. La descarga y la instalación son dirigidas a máquinas Mac, PC o Linux RStudio está disponible en <http://www.rstudio.org/>.

Para usar Markdown o `knitr`/ \LaTeX se requiere que el paquete `knitr` esté instalado en su sistema.

2.1 Conectándose al servidor de RStudio

Los servidores de RStudio han sido ajustados en algunas escuelas para facilitar la computación basada en la nube.

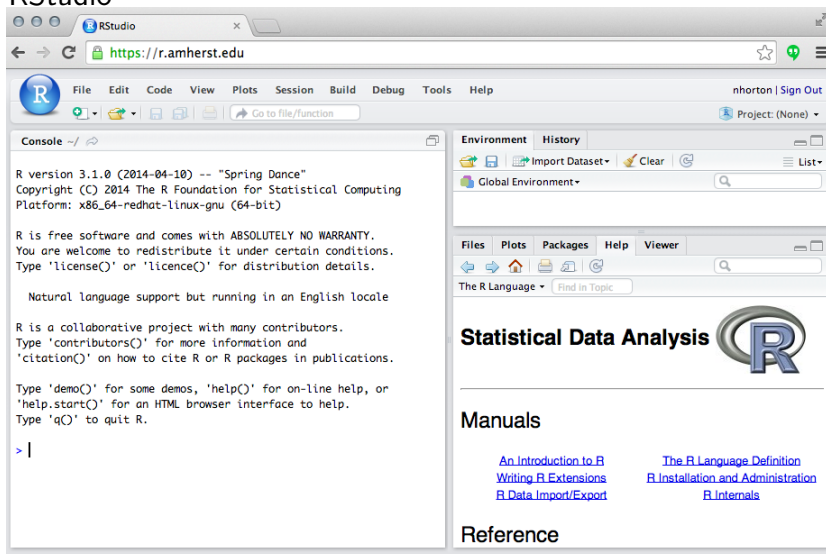
Una vez que se conecte al servidor, usted va ver una pantalla de inicio de sesión:



Los servidores de RStudio han sido instalados en muchas instituciones. Más detalles acerca de licencias académicas (gratuitas) para 'RStudio Server Pro' como también instrucciones de ajuste se pueden encontrar en <http://www.rstudio.com/resources/faqs> en la viñeta Academic

El servidor de RStudio no suele funcionar bien con Internet Explorer

Una vez que se identifique, usted debería ver la interfaz de RStudio



Observe que RStudio divide su universo en 4 paneles. Algunos de estos paneles después son subdivididos en viñe-

tas múltiples. Qué viñetas aparecen en los paneles pueden ser personalizados por el usuario. R puede hacer muchísimo más que una simple calculadora, y vamos a introducir características adicionales en el debido tiempo. Sin embargo, hacer cálculos sencillos en R es una buena forma de iniciar el conocimiento de las características de RStudio.

Los comandos ingresados en la ventana de la *Consola* son inmediatamente ejecutados por R. Una buena forma de familiarizarse con la consola es hacer algunos cálculos sencillos. La mayoría de este trabajo lo podría esperar de una calculadora típica. Intente escribir los siguientes comandos en el panel de la consola:

```
> 5 + 3
[1] 8
> 15.3 * 23.4
[1] 358.02
> sqrt(16)                # raíz cuadrada
[1] 4
```

Este último ejemplo demuestra como las funciones son llamadas dentro de R como también el uso de comentarios. Los comentarios son anticipados por el símbolo `#`. Los comentarios pueden ser de mucha ayuda cuando se escriben scripts con múltiples comandos o para anotar código de ejemplo para sus estudiantes.

Usted puede guardar valores en variables nombradas para reutilización posterior.

```
> product = 15.3 * 23.4    # Guardar el resultado
> product                  # Desplegar el resultado
[1] 358.02
> product <- 15.3 * 23.4   # <- Puede ser utilizado
>                          # en lugar de =
> product
[1] 358.02
```

Una vez que las variables son definidas, pueden ser referenciadas en otras operaciones y funciones.

Es probablemente mejor establecer el uso de uno u otro operador de asignación en lugar de estar cambiándolo. Personalmente preferimos el operador de la flecha, porque representa visualmente lo que está pasando en la asignación además porque deja clara la distinción entre el operador de asignación, el uso de `=` en este caso es para dar valores a los argumentos de funciones, y el uso de `==` para probar la igualdad.

```

> 0.5 * product                # La mitad del producto
[1] 179.01

> log(product)                 # log (natural) del producto
[1] 5.880589

> log10(product)               # log base 10 del producto
[1] 2.553907

> log2(product)                # log base 2 del producto
[1] 8.483896

> log(product, base=2)         # log base 2 del producto, de otra forma
[1] 8.483896

```

El punto y coma puede ser utilizado para ubicar comandos múltiples en una sola línea. Un uso frecuente de esto puede ser para salvar y después imprimir un valor, en una sola línea.

```

> product <- 15.3 * 23.4; product    # Salva el resultado y lo muestra
[1] 358.02

```

2.1.1 Información de la versión

En algunos momentos, puede ser útil revisar qué versión del paquete `mosaic`, `Ry RStudio` está usando. Ejecutando `sessionInfo()` se va desplegar información de R, los paquetes que están cargados y `RStudio.Version()` le proveerá información sobre la versión de `RStudio`.

```

> sessionInfo()

R version 3.4.1 (2017-06-30)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 15063)

Matrix products: default

locale:
[1] LC_COLLATE=Spanish_Costa Rica.1252
[2] LC_CTYPE=Spanish_Costa Rica.1252

```

```
[3] LC_MONETARY=Spanish_Costa Rica.1252
[4] LC_NUMERIC=C
[5] LC_TIME=Spanish_Costa Rica.1252
```

attached base packages:

```
[1] grid      stats      graphics  grDevices  utils      datasets
[7] methods   base
```

other attached packages:

```
[1] knitr_1.16      fastR_0.10.2      mosaic_0.14.4
[4] Matrix_1.2-10   mosaicData_0.14.0 ggplot2_2.2.1
[7] dplyr_0.7.1      lattice_0.20-35    MASS_7.3-47
```

loaded via a namespace (and not attached):

```
[1] Rcpp_0.12.11     bindr_0.1          magrittr_1.5
[4] splines_3.4.1    munsell_0.4.3      colorspace_1.3-2
[7] R6_2.2.2          gg dendro_0.1-20    rlang_0.1.1
[10] stringr_1.2.0     plyr_1.8.4          tools_3.4.1
[13] gtable_0.2.0      lazyeval_0.2.0     assertthat_0.2.0
[16] tibble_1.3.3      bindrcpp_0.2        gridExtra_2.2.1
[19] tidyr_0.6.3       evaluate_0.10.1     glue_1.1.1
[22] stringi_1.1.5     compiler_3.4.1      scales_0.4.1
[25] pkgconfig_2.0.1
```

2.2 *Trabajando con archivos*

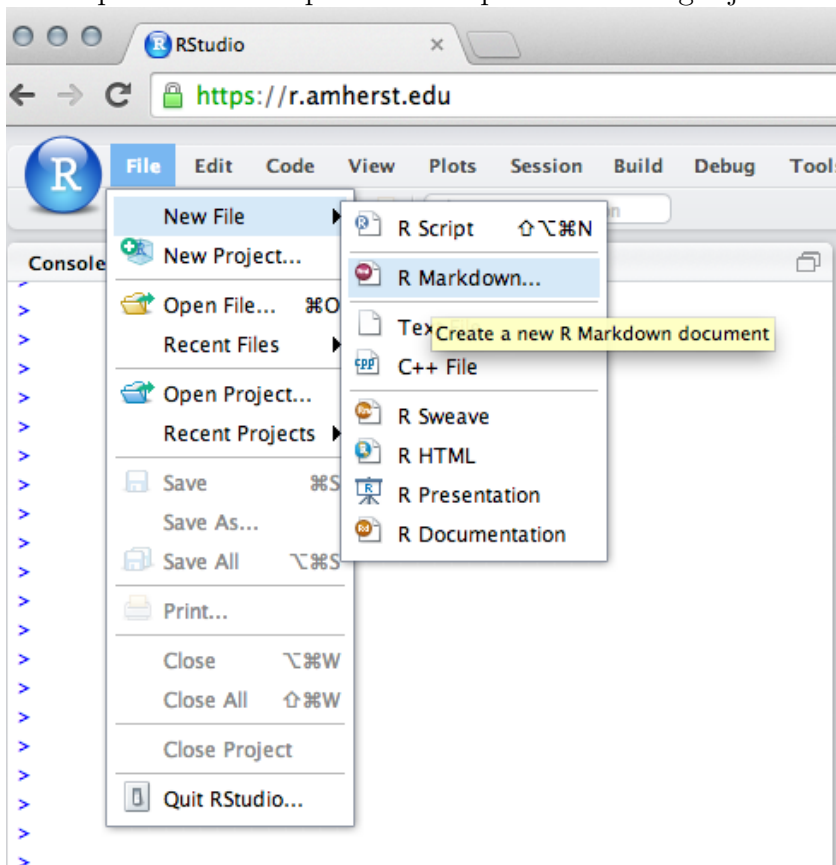
2.2.1 *Trabajando con archivos R script*

Como alternativa, los comandos de R pueden ser almacenados en un archivo. RStudio ofrece un editor integrado para editar este tipo de documentos y facilita ejecutar algunos o todos los comandos. Para crear un archivo, seleccione en el menú de RStudio **File**, después **New File**, después **R Script**. Una ventana del editor de archivos se va abrir en el panel de **Source**. El código de R puede ser escrito ahí, y botones e ítems de menú son provistos para correr todo el código (llamado abastecer el documento) o ejecutar el código en una sola línea o en una sección seleccionada del documento.

2.2.2 Trabajando con RMarkdown, y knitr/L^AT_EX

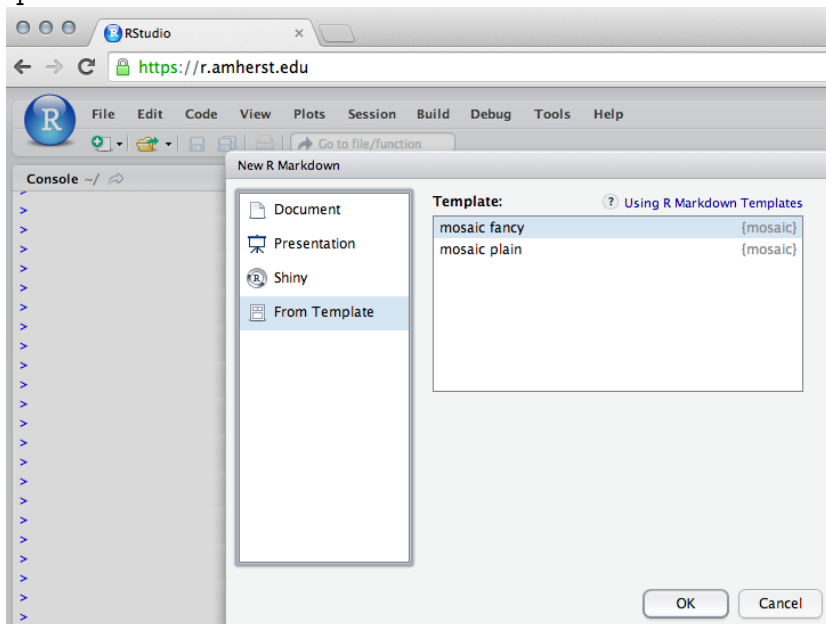
Una tercera alternativa es tomar ventaja del apoyo que brinda RStudio para la investigación reproducible. Si usted ya conoce L^AT_EX, va a querer investigar las capacidades de knitr/L^AT_EX. Para aquellos que no conocen L^AT_EX todavía, el sistema más sencillo de RMarkdown proporciona una entrada fácil en el mundo de los métodos de la investigación reproducible. Además brinda una buena estructura para que los estudiantes creen sus tareas y reportes que incluyen texto, código de R, salidas de R y gráficos.

Para crear un nuevo archivo de RMarkdown, seleccione la pestaña de File, después New File, después RMarkdown. El archivo va a ser abierto con un pequeño documento de plantilla que ilustra la especificación que tiene el lenguaje

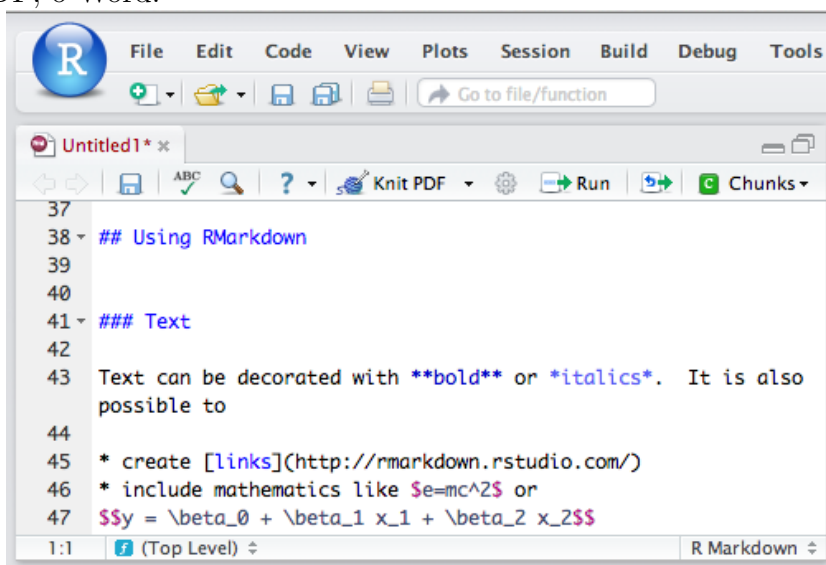


El paquete mosaic incluye dos plantillas útiles de RMarkdown para empezar: **fancy** que es fina y exquisita (y esa era la intención, para dar una observación de las características) mientras que **plain** es útil como un punto de inicio para un nuevo análisis. A estos se puede acceder usando la opción de

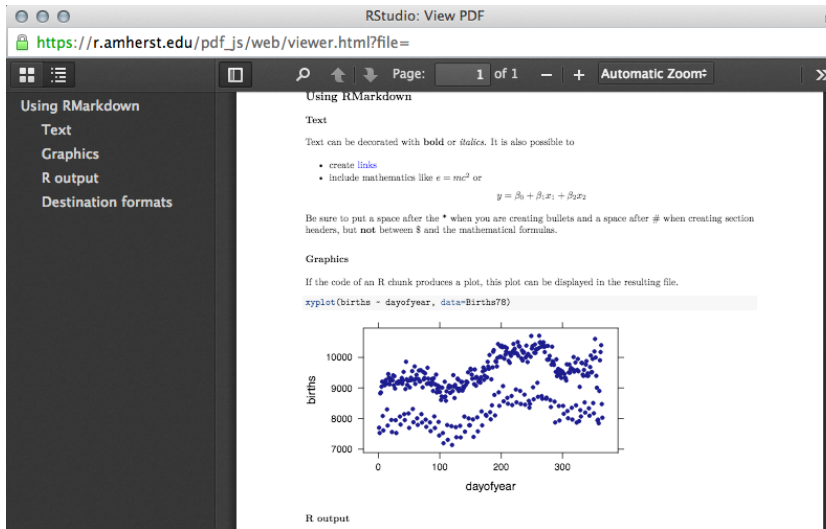
Template cuando se crea un nuevo archivo de RMarkdown.



Presione el botón de Knit para convertir a archivo HTML, PDF, o Word.



Esto va generar una versión con formato del documento.



Hay un botón (marcado con un signo de pregunta) que provee una pequeña descripción de las especificaciones del comando especificado. El sitio web de RStudio incluye tutoriales más extensivos sobre la utilización de RMarkdown.

Es importante recordar que contrario a los scripts de R que son ejecutados en la consola y tiene acceso al entorno de la consola, los archivos de RMarkdown y **knitr**/**L^AT_EX** no tienen acceso al entorno de la consola. Esto es una buena característica puesto que obliga a los archivos a ser auto-contenidos, lo cual los hace transferibles y respeta las buenas prácticas de la investigación reproducible. Pero en el caso de los principiantes, especialmente si adoptan una estrategia de probar cosas en la consola, y copiar y pegar código de la consola en su documento, crean archivos que son incompletos y por esto no compilan correctamente.

PRECAUCIÓN!

Los archivos RMarkdown, and **knitr**/**L^AT_EX** no tienen acceso al entorno de la consola, entonces el código debe ser autocontenido.

2.3 Otros paneles y viñetas

2.3.1 La viñeta de History(historial)

Como los comandos son ingresados en la consola, así aparecen en la viñeta de **History**. Estas "historias" o historiales pueden ser guardados y cargados, hay una opción de búsqueda para encontrar comandos previos, y líneas individuales previas o secciones pueden ser transferidas de vuelta a la consola. Mantener la viñeta de **History** abierta le permite ir atrás y ver varios comandos anteriores. Esto puede ser especialmente útil cuando ciertos comandos producen

una salida grande entonces la pantalla se mueve hacia abajo rápidamente.

2.3.2 *Comunicación entre viñetas*

RStudio brinda varias formas de mover código de R entre las viñetas. Presionar el botón de Run en el panel de edición para un script de R o RMarkdown o algún otro documento va copiar líneas de código en la consola y ejecutarlas.

2.3.3 *La viñeta de files(archivos)*

La viñeta de Files provee un manejador de datos simples. En esta se puede navegar de forma amigable y puede ser utilizada para abrir, renombrar y borrar archivos. En la versión de navegador web de RStudio, la pestaña de archivos también ofrece una utilidad para subir los archivos de la máquina local al servidor. En archivos de RMarkdown y **knitr** también se puede ejecutar código en algún “chunk” del archivo o en todos los “chunks” del archivo. Cada una de estas características simplifica el intentar escribir código “en vivo” mientras se crea un documento que deja un historial del código.

En la dirección inversa, el código del historial puede ser copiado de nuevo en la consola para ejecutarlo de nuevo (después de un poco de edición) o en una de las viñetas de edición, para la inclusión en un archivo.

2.3.4 *La viñeta de Help(ayuda)*

La viñeta de ayuda (**Help**) es donde RStudio despliega los archivos de ayuda de R. Se puede abrir un archivo de ayuda usando el operador `?` en la consola. Por ejemplo, el siguiente comando va mostrarnos el archivo de ayuda para la función de logaritmo.

```
> ?log
```

2.3.5 *La viñeta de Environment(entorno)*

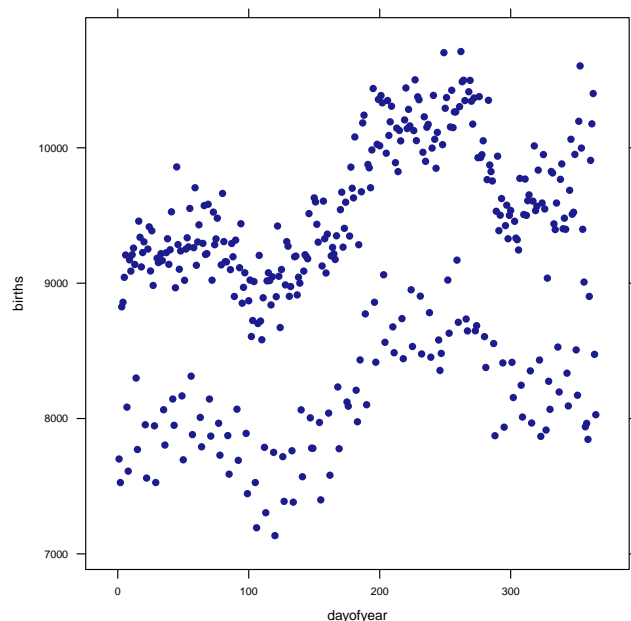
La viñeta de entorno (**Environment**) muestra los objetos disponibles para la consola. Estos están subdivididos en datos, valores (no son objetos de tipo conjunto de datos,

ni objetos de tipo función) y funciones. El ícono de la escoba puede ser utilizado para remover todos los objetos del entorno, y es bueno hacer esto de vez en cuando, especialmente cuando se está ejecutando RStudio en el servidor o si decide salvar el entorno cuando cierra RStudio, puesto que en estos casos los objetos se pueden quedar en el entorno indefinidamente.

2.3.6 La viñeta de *Plots*(Gráficos)

Los gráficos creados en la consola son desplegados en la viñeta de gráficos (*Plots*). Por ejemplo, los siguientes comandos despliegan el número de nacimientos en Estados Unidos cada día en 1978.

```
> # Esto va hacer los gráficos de lattice disponibles en la sesión
> # Así como el set de datos Births78
> require(mosaic)
> xyplot(births ~ dayofyear, data=Births78)
```



En la viñe-

ta de gráficos(*Plots*), se puede navegar a gráficos anteriores y también exportar gráficos en varios formatos, después de interactivamente manipular las dimensiones.

2.3.7 *La viñeta de Packages(paquetes)*

Mucha de la funcionalidad de R está localizada en paquetes, muchos de los cuales se pueden obtener de la casa matriz llamada CRAN (Comprehensive R Archive Network). La viñeta de **Packages** facilita instalar y cargar paquetes. También le permite buscar paquetes que han sido actualizados desde que usted los instaló.

3

Una variable cuantitativa

3.1 Resúmenes numéricos

R incluye comandos para resumir variables numéricamente. Estos incluyen la capacidad de calcular la media, la desviación estándar, la varianza, mediana, el mínimo, máximo el primer cuartil, el tercer cuartil, el rango intercuartil así como cuartiles arbitrarios. Esto lo ilustraremos usando la medida de síntomas depresivos (que toma valores entre 0 y 60, con notas mayores indicando mayores síntomas depresivos) del CESD (Center for Epidemiologic Studies–Depression).

Para mejorar la legibilidad de las salidas, también vamos a elegir una cantidad de números predeterminados para desplegar el resultado de forma más razonable (vea `?options()` para un mayor número de configuraciones posibles).

```
> require(mosaic)
> require(mosaicData)
> options(digits=4)
> mean(~ cesd, data=HELPrct)
```

```
[1] 32.85
```

Note que `mean()`, dentro del paquete `mosaic` utiliza la notación de fórmula de los gráficos lattice comunes y los modelos lineales (e.g `lm()`). El paquete `mosaic` permite que muchas otras funciones utilicen la misma notación, la cual vamos a estar utilizando a lo largo de este documento.

La misma salida puede ser creada utilizando los siguientes comandos (sin embargo, nosotros utilizaremos la version de MOSAIC cuando sea posible)

CAVANDO HONDO

Si usted no ha visto la notación de fórmula antes de este libro, *Empiece a enseñar con R* lo presenta una detalladamente. *Start Modeling with R*, otro libro, detalla la relación entre el proceso de modelaje y la notación de la fórmula.

```
> with(HELPrct, mean(cesd))
```

```
[1] 32.85
```

```
> mean(HELPrct$cesd)
```

```
[1] 32.85
```

La misma funcionalidad existe para otras estadísticas de resumen

```
> sd(~ cesd, data=HELPrct)
```

```
[1] 12.51
```

```
> sd(~ cesd, data=HELPrct)^2
```

```
[1] 156.6
```

```
> var(~ cesd, data=HELPrct)
```

```
[1] 156.6
```

Es también muy sencillo calcular cuartiles de la distribución

```
> median(~ cesd, data=HELPrct)
```

```
[1] 34
```

Como predeterminado, la función `quantile()` muestra los cuartiles, pero se le puede dar un vector de cuantiles a desplegar.

```
> with(HELPrct, quantile(cesd))
```

```
0%  25%  50%  75% 100%
 1    25   34   41   60
```

```
> with(HELPrct, quantile(cesd, c(.025, .975)))
```

```
2.5% 97.5%
6.3  55.0
```

Finalmente, la función `favstats()` en el paquete `mosaic`, muestra un resumen conciso de estadísticas descriptivas útiles.

```
> favstats(~ cesd, data=HELPrct)
```

```
min Q1 median Q3 max  mean    sd    n missing
 1  25     34  41  60 32.85 12.51 453      0
```

PRECAUCIÓN!

No todos los comandos han sido actualizados para utilizar la notación de fórmula. Para estas variables se debe acceder a los conjuntos de datos con la función `with()`

3.2 Resúmenes gráficos

La función `histogram()` es utilizada para crear un histograma. Aquí utilizamos la notación de fórmula (como se discutió en el libro *Start Modeling with R*) para especificar que queremos un histograma de los puntajes del CESD

```
> histogram(~ cesd, data=HELPrct)
```

Podemos usar las opciones de `width()` y `center()` para controlar la ubicación y tamaño de las barras.

```
> histogram(~ cesd, width=5, center=2.5, data=HELPrct)
```

En el conjunto de datos `HELPrct`, aproximadamente un cuarto de los sujetos son mujeres.

```
> tally(~ sex, data=HELPrct)
```

```
sex
female  male
    107    346
```

```
> tally(~ sex, format="percent", data=HELPrct)
```

```
sex
female  male
 23.62  76.38
```

Directamente vamos a restringir nuestra atención a únicamente los sujetos femeninos. Si vamos a hacer una gran cantidad de cosas con una parte de nuestro conjunto de datos, lo más sencillo puede ser hacer un nuevo conjunto de datos conteniendo sólo los casos en los que estamos interesados. La función `filter()` en el paquete `dplyr` puede ser utilizada para generar una nueva tabla de datos que contenga solamente los hombres o las mujeres (observar la sección 13.5). Una vez que esto sea creado, la función `stem()` es utilizada para crear un diagrama de tallo y hojas.

```
> female <- filter(HELPrct, sex=='female')
> male <- filter(HELPrct, sex=='male')
> with(female, stem(cesd))
```

PRECAUCIÓN!

Note que el operador para igualdad utiliza *dos* signos de igual.

The decimal point is 1 digit(s) to the right of the |

0 | 3

```

0 | 567
1 | 3
1 | 555589999
2 | 123344
2 | 66889999
3 | 0000233334444
3 | 5556666777888899999
4 | 00011112222334
4 | 555666777889
5 | 011122222333444
5 | 67788
6 | 0

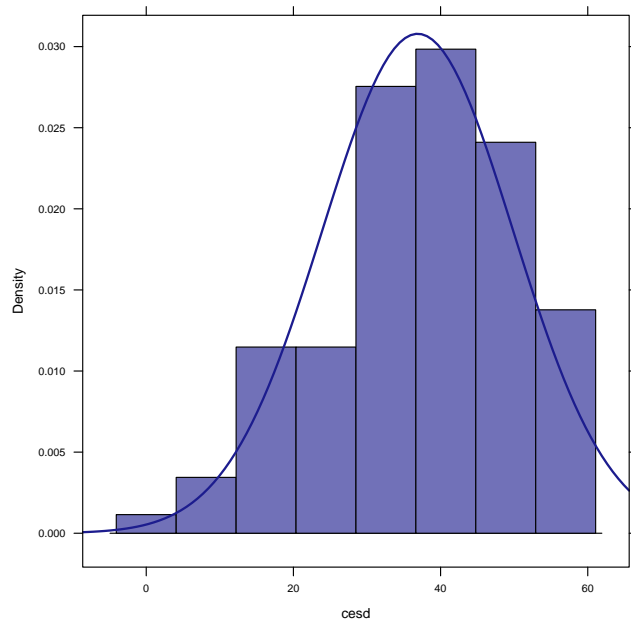
```

Los subconjuntos de datos también pueden ser generadas y utilizadas *.a medio camino*”(esta vez lo veremos traslapando la densidad normal):

```

> histogram(~ cesd, fit="normal",
  data=filter(HELPrct, sex=='female'))

```

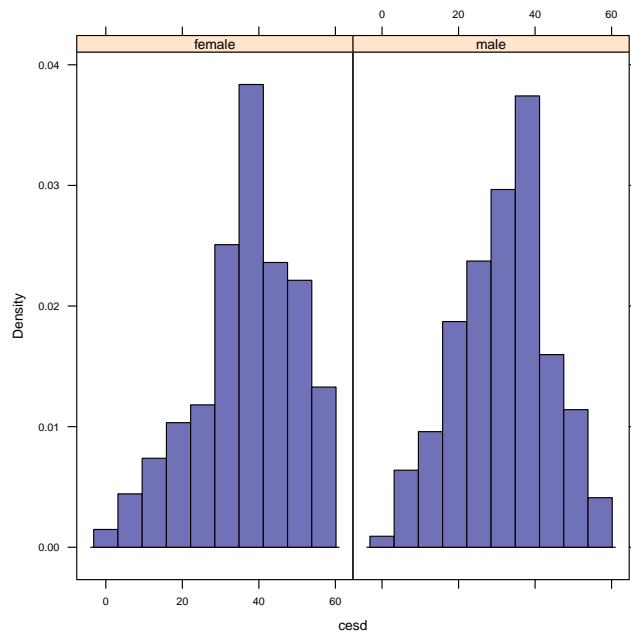


Alternativamente podemos hacer gráficos uno al lado del otro para comparar múltiples subconjuntos:

```

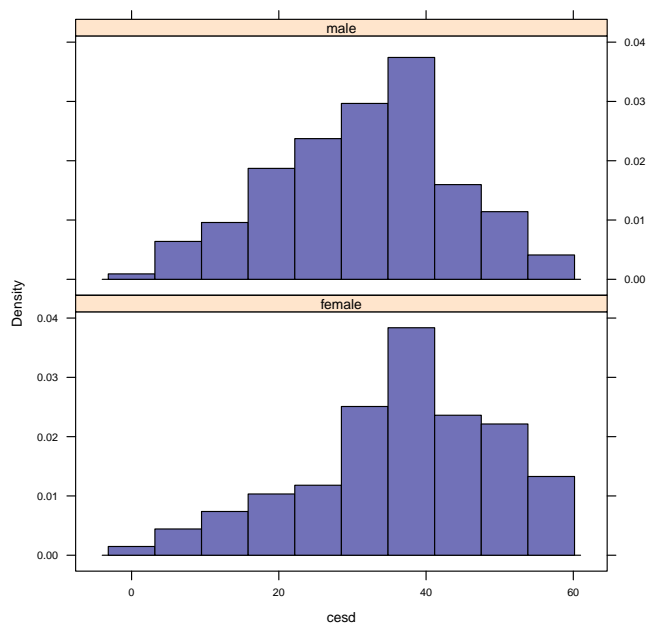
> histogram(~ cesd | sex, data=HELPrct)

```



La estructura puede ser reacomodada

```
> histogram(~ cesd | sex, layout=c(1, 2), data=HELPrct)
```

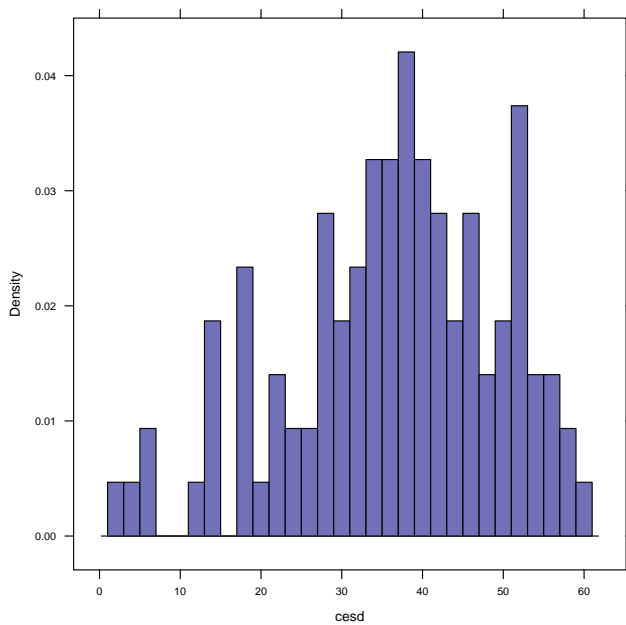


Podemos controlar el número de barras de varias formas.
Esto puede ser especificado mediante el número total

```
> histogram(~ cesd, nint=20, data=female)
```

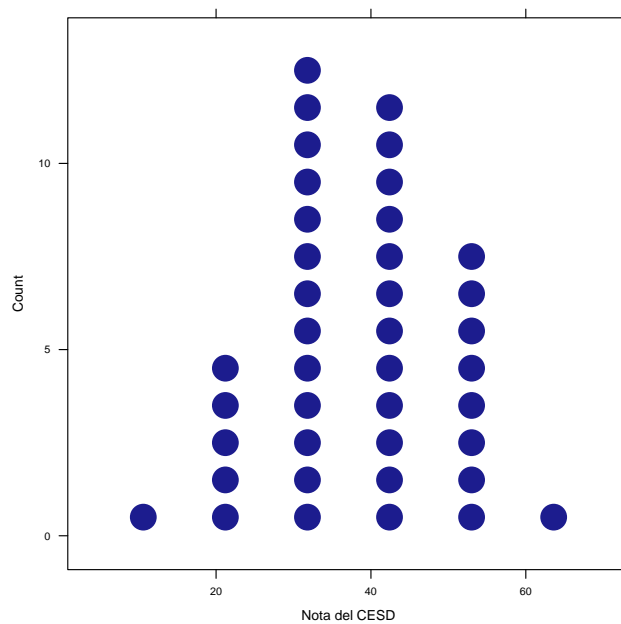
El ancho de las barras puede ser especificado

```
> histogram(~ cesd, width=2, data=female)
```



La función `dotPlot()` es utilizada para crear un gráfico de puntos de pequeños grupos de sujetos (mujeres indigentes). También demostramos como cambiamos el nombre del eje X.

```
> dotPlot(~ cesd, xlab="Nota del CESD",
  data=filter(HELPrct, (sex=="female") & (homeless=="homeless")))
```

3.3 Curvas de densidad

Una desventaja de los histogramas es que pueden ser sensibles a la elección del número de barras. Otro tipo de gráficos a considerar es la curva de densidad.

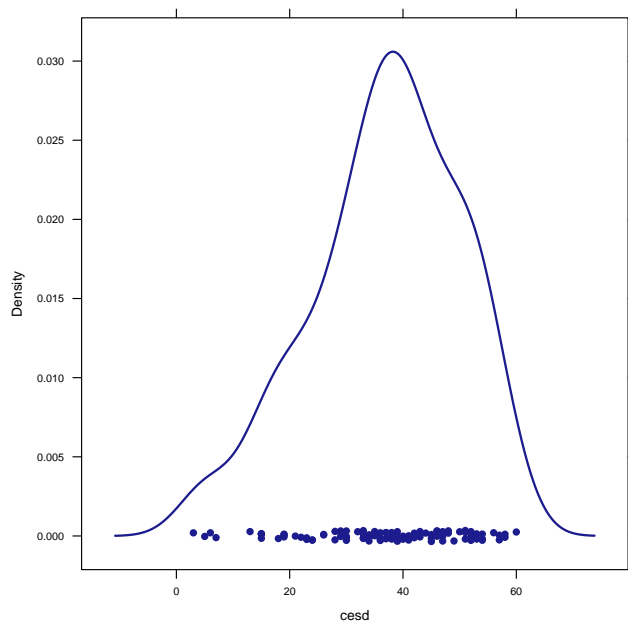
Aquí adornamos el gráfico de densidad con algunas adiciones para demostrar cómo se construye un gráfico para propósitos pedagógicos. Le agregamos algún texto, una función de densidad normal superimpuesta así como una línea vertical. Una variedad de tipos de líneas y colores pueden ser especificados, así como anchos.

Los gráficos de densidad también son sensibles a ciertas opciones. Si su gráfico de densidad es demasiado irregular o demasiado suavizado, intente cambiar el argumento `adjust` : más largo que 1 para gráficos demasiado suavizados, menos de 1 para gráficos sumamente irregulares.

CAVANDO HONDO

La función `plotFun()` puede también utilizarse para anotar gráficos (ver sección 10.2.1)

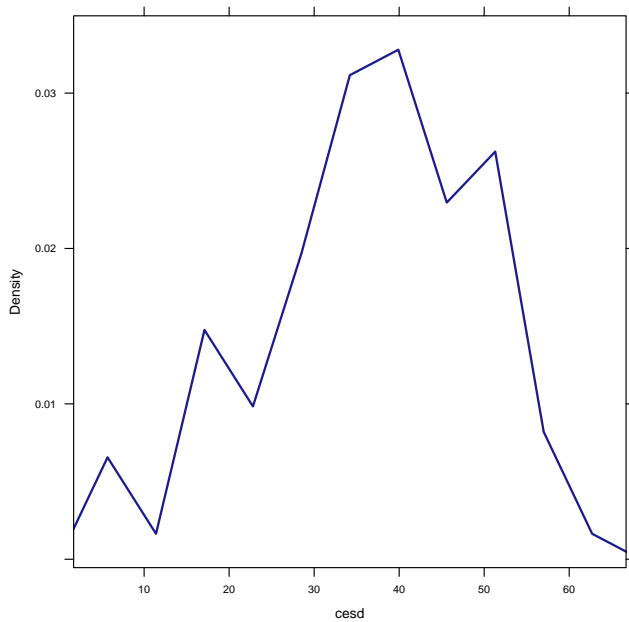
```
> densityplot(~cesd, data=female);ladd(grid.text(x=0.2, y=0.8, 'solo mujeres'));ladd(p
```



3.4 *Polígonos de frecuencia*

Una tercera opción es un polígono de frecuencias, donde el gráfico es creado uniendo los puntos medios en la parte superior de las barras de un histograma.

```
> freqpolygon(~ cesd, data=female)
```



3.5 Distribuciones normales

La curva de densidad más famosa es la distribución normal. La función `xpnorm()` despliega la probabilidad de que una variable aleatoria sea menor que el primer argumento, para una distribución normal con una media dada en el segundo argumento y con la desviación estándar de tercer argumento. Más información sobre las distribuciones de probabilidad se puede encontrar en la sección 11.

`x` es de eXtra.

```
> xpnorm(1.96, mean=0, sd=1)
```

If $X \sim N(0, 1)$, then

$$P(X \leq 1.96) = P(Z \leq 1.96) = 0.975$$

$$P(X > 1.96) = P(Z > 1.96) = 0.025$$

```
[1] 0.975
```

3.6 Inferencia para una sola muestra

Podemos calcular un intervalo de confianza para el 95 % de confianza para la media de puntuación del CESD para mujeres

usando la prueba t

```
> t.test(~ cesd, data=female)

One Sample t-test

data:  female$cesd
t = 29, df = 110, p-value <2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 34.39 39.38
sample estimates:
mean of x
 36.89

> confint(t.test(~ cesd, data=female))

mean of x lower upper level
1      36.89 34.39 39.38  0.95
```

Sin embargo, esto también puede observarse usando bootstrap. El estadístico que queremos remuestrear es la media

```
> mean(~ cesd, data=female)

[1] 36.89
```

One resampling trial can be carried out:

```
> mean(~ cesd, data=resample(female))

[1] 37.7
```

Otro generará un resultado diferente:

```
> mean(~ cesd, data=resample(female))

[1] 34.93
```

Ahora haremos 1000 remuestreos, guardando el resultado en un objeto llamado trials: called `trials`:

```
> trials <- do(1000) * mean(~ cesd, data=resample(female))
> head(trials, 3)

mean
1 36.68
2 35.49
3 35.90
```

CAVANDO HONDO

More details and examples can be found in the `mosaic` package Resampling Vignette.

Aquí se muestrea con remplazo de la tabla de datos original, creando una tabla de datos remuestreada con el mismo número de filas.

Aunque un único intento es de poco uso, es inteligente hacer a los estudiantes calcular para mostrarle que están (usualmente) consiguiendo un resultado diferente que el que se obtiene sin remuestreo

```
> qdata(~ mean, c(.025, .975), data=trials)
```

	quantile	p
2.5%	34.27	0.025
97.5%	39.38	0.975

4

Una variable categóricas

4.1 Resúmenes numéricos

La función `tally()` puede ser utilizada para calcular conteos, porcentajes y proporciones para variables categóricas.

```
> tally(~ homeless, data=HELPrct)
```

```
homeless
homeless  housed
      209      244
```

```
> tally(~ homeless, margins=TRUE, data=HELPrct)
```

```
homeless
homeless  housed  Total
      209      244    453
```

```
> tally(~ homeless, format="percent", data=HELPrct)
```

```
homeless
homeless  housed
     46.14     53.86
```

```
> tally(~ homeless, format="proportion", data=HELPrct)
```

```
homeless
homeless  housed
     0.4614     0.5386
```

4.2 La prueba binomial

Un intervalo de confianza exacto para la proporción (así como una prueba de hipótesis para que la proporción sea

CAVANDO HONDO

El libro *Empiece a enseñar con R* introduce la notación de fórmula utilizada alrededor de este libro. Vea también *Empiece a enseñar con R* para conexiones al modelaje estadístico..

igual a un valor particular [predeterminado 0.5]) puede ser calculado usando la función `binom.test()`. El estándar de `binom.test()` requiere que tabulemos

```
> binom.test(209, 209 + 244)

data: 209 out of 209 + 244
number of successes = 210, number of trials = 450, p-value =
0.1
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.4147 0.5085
sample estimates:
probability of success
      0.4614
```

El paquete `mosaic` utiliza la notación de fórmula que evita la necesidad de pre-tabular los datos.

```
> result <- binom.test(~ (homeless=="homeless"), data=HELPrct)
> result

data: HELPrct$(homeless == "homeless") [with success = TRUE]
number of successes = 210, number of trials = 450, p-value =
0.1
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.4147 0.5085
sample estimates:
probability of success
      0.4614
```

Como es generalmente con este tipo de comandos, hay una cantidad de resultados disponibles del objeto devuelto por la función.

```
> names(result)

[1] "statistic"  "parameter"  "p.value"    "conf.int"
[5] "estimate"   "null.value" "alternative" "data.name"
```

Estos pueden ser extraídos usando el operador `$` o un extractor de la función. Por ejemplo, el usuario puede extraer el intervalo de confianza o el p-value.

```
> result$statistic
```



```

number of successes
      209

> confint(result)

probability of success lower upper level
1          0.4614 0.4147 0.5085  0.95

> pval(result)

p.value
0.1101

```

4.3 La prueba de la proporción

Un intervalo y prueba similar pueden ser calculados usando la función `prop.test()`. Aquí está una cuenta del número de personas a cada uno de los dos niveles de la variable `homeless`.

```

> tally(~ homeless, data=HELPrct)

homeless
homeless  housed
      209     244

```

La función de `prop.test()` va a llevar a cabo los cálculos de la prueba de la proporción y reporta el resultado.

```

> prop.test(~ (homeless=="homeless"), correct=FALSE, data=HELPrct)

1-sample proportions test without continuity correction

data:  HELPrct$(homeless == "homeless") [with success = TRUE]
X-squared = 2.7, df = 1, p-value = 0.1
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.4160 0.5074
sample estimates:
      p
0.4614

```

En esta declaración, `prop.test()` está examinando la variable `homeless` en la misma forma que `tally()` lo haría.

`prop.test()` también puede trabajar directamente con conteos numéricos, de la forma que `binom.test()` lo hace.

CAVANDO HONDO

La mayoría de objetos en R tiene un método de impresión o `print()`. Entonces cuando conseguimos el resultado, lo que hacemos es ver desplegado en la consola el `print(resultado)`. Por lo que puede haber una cantidad buena de información escondiéndose en el objeto mismo. En algunas situaciones, así como los gráficos, los objetos son devueltos *invisiblemente*, entonces nada se imprime. Eso evita que tenga que ver una salida no deseada para consumo humano. Usted de todos modos puede asignar el objeto devuelto a una variable y procesarlo más tarde, incluso si nada aparece en la pantalla. Esto en ocasiones es útil para hacer imágenes gráficas de lattice.

MÁS INFO

Escribimos `homeless=="homeless"` para definir de forma no ambigua que proporción estamos considerando. Podríamos haber escrito también `homeless=="housed"`.

`prop.test()` calcula un estadístico Ji-cuadrado. La mayoría de textos introductorios usan un

```
> prop.test(209, 209 + 244, correct=FALSE)

1-sample proportions test without continuity correction

data: 209 out of 209 + 244
X-squared = 2.7, df = 1, p-value = 0.1
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.4160 0.5074
sample estimates:
      p
0.4614
```

4.4 Bondad de ajuste

Una variedad de pruebas de bondad de ajuste pueden ser calculadas para una distribución dada. Para los datos de HELP, podemos probar la hipótesis nula de que la proporción de sujetos en cada grupo de abuso de sustancias es igual en la población original.

```
> tally(~ substance, format="percent", data=HELPrct)
```

```
substance
alcohol cocaine heroin
 39.07   33.55   27.37
```

```
> observed <- tally(~ substance, data=HELPrct)
> observed
```

```
substance
alcohol cocaine heroin
   177    152    124
```

```
> p <- c(1/3, 1/3, 1/3) # equivalente a rep(1/3, 3)
> chisq.test(observed, p=p)
```

```
Chi-squared test for given probabilities
```

```
data: observed
X-squared = 9.3, df = 2, p-value = 0.01

> total <- sum(observed); total

[1] 453
```

PRECAUCIÓN!
Además de las opción de `format`, hay una opción de incluir los marginales, para incluir el marginal de los totales en la tabla. El predeterminado en `tally()` es `margins=FALSE`. ¡Inténtelo!

```
> expected <- total*p; expected
```

```
[1] 151 151 151
```

También podemos calcular el estadístico χ^2 como la función observada y los valores esperados.

```
> chisq <- sum((observed - expected)^2/(expected)); chisq
```

```
[1] 9.311
```

```
> 1 - pchisq(chisq, df=2)
```

```
[1] 0.009508
```

Puede ser útil consultar un gráfico del estadístico, donde el área sombreada representa el valor de la derecha del valor observado.

```
> plotDist("chisq", df=2, groups = x > 9.31, type="h")
```

Alternativamente, el paquete `mosaic` ofrece una versión del `chisq.test()` con una salida "más rellena"

```
> xchisq.test(observed, p=p)
```

```
Chi-squared test for given probabilities
```

```
data: x
```

```
X-squared = 9.3, df = 2, p-value = 0.01
```

```
      177      152      124
(151.00) (151.00) (151.00)
[4.4768] [0.0066] [4.8278]
< 2.116> < 0.081> <-2.197>
```

```
key:
```

```
  observed
  (expected)
[contribution to X-squared]
<Pearson residual>
```

La función `pchisq()` calcula la probabilidad de que una variable aleatoria χ^2 con `df()` grados de libertad es menor o igual al valor dado. Aquí calculamos el complemento para encontrar el área al derecho del valor observado del estadístico Chi-cuadrado.

`x` en `xchisq.test()` significa eXtra

```
> # limpiar las variables que ya no son necesarias
> rm(observed, p, total, chisq)
```

Objetos en el espacio de trabajo son listados en la viñeta de ENTORNO (ENVIRONMENT) en RStudio. Si usted quiere limpiar un poco la lista, remueva los objetos que ya no necesita con `((rm()))`.

5

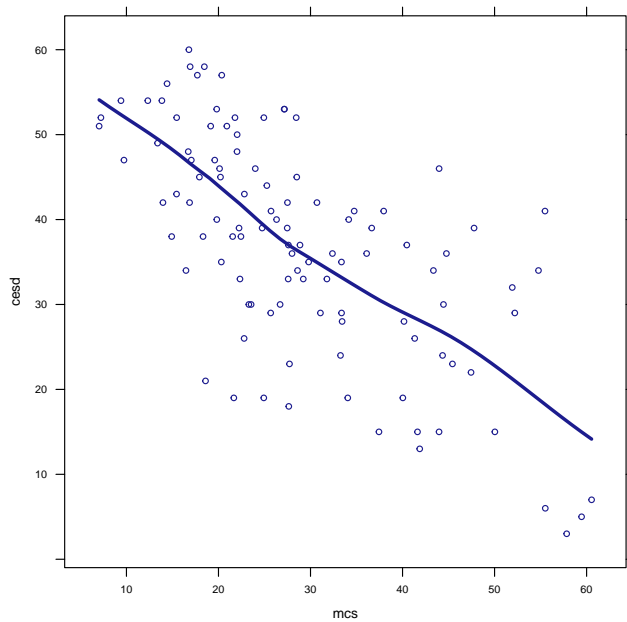
Dos variables cuantitativas

5.1 Gráficos de dispersión

Siempre motivamos a nuestros estudiantes a iniciar cualquier análisis mediante la graficación de sus datos. Aquí presentamos un gráfico de dispersión de la CESD (una medida de síntomas depresivos, mayores puntuaciones indican mayor cantidad de síntomas) y el MCS (Puntuación del componente mental del SF-36, donde las puntuaciones mayores indican mejor funcionamiento) para los sujetos femeninos con una línea lowess (loess en español, una línea de regresión local), usando círculos como la figura para los puntos y una línea un poco más gruesa.

La línea lowess puede ayudar a evaluar la linealidad de la relación. Esto es especificando en la función ambos puntos (usando 'p') y el suavizador lowess.

```
> require(mosaicData)
> females <- filter(HELPrct, female==1)
> xyplot(cesd ~ mcs, type=c("p","smooth"), pch=1,
  cex=0.6, lwd=3, data=females)
```

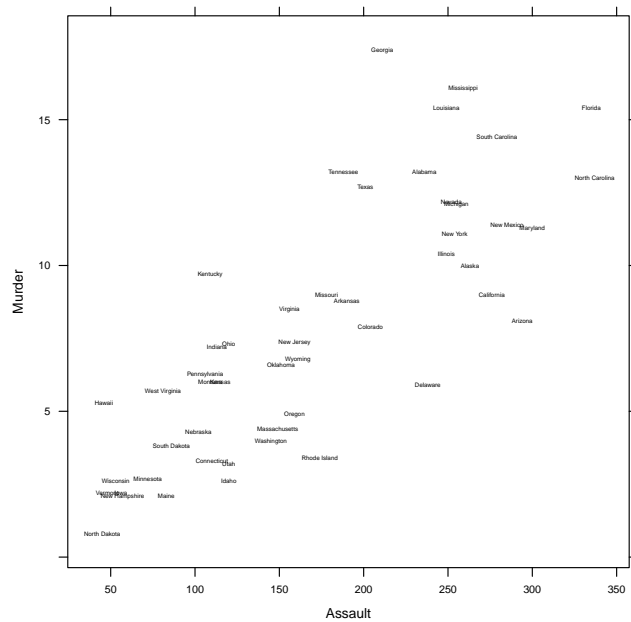


Con esto es sencillo poder agregar algo más que símbolos en el gráfico de dispersión. En este ejemplo, el conjunto de datos de `USArrest` puede ser utilizado para graficar la asociación entre los índices de asalto y asesinato, con el nombre del estado. Esto requiere una función de panel.

CAVANDO HONDO

El libro *Start Modelling with R* da a mayor profundidad como se utiliza el lenguaje para modelar. *Empiece a enseñar con R* también brinda una guía útil para entenderlo

```
> panel.labels <- function(x, y, labels='x',...) {
  panel.text(x, y, labels, cex=0.4, ...)
}
> xyplot(Murder ~ Assault, panel=panel.labels,
  labels=rownames(USArrests), data=USArrests)
```



5.2 Correlación

Las correlaciones pueden ser calculadas para un par de variables o para una matriz de variables.

```
> cor(females$cesd, females$mcs)
```

```
[1] -0.6738
```

```
> smallHELP <- select(females, cesd, mcs, pcs)
> cor(smallHELP)
```

	cesd	mcs	pcs
cesd	1.0000	-0.6738	-0.3685
mcs	-0.6738	1.0000	0.2664
pcs	-0.3685	0.2664	1.0000

Como predeterminado, la correlación de Pearson es la que se muestra. Otras variables (ejemplo, Spearman) pueden ser especificadas usando la opción de `method` `method` option.

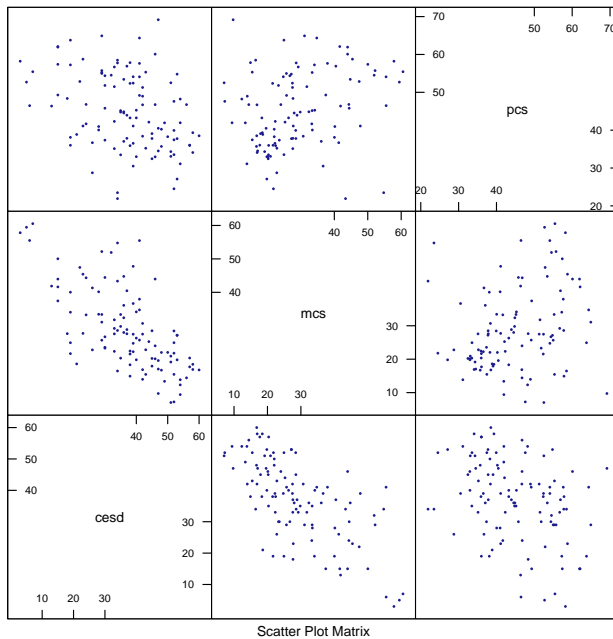
```
> cor(females$cesd, females$mcs, method="spearman", data=females)
```

```
[1] -0.6662
```

5.3 Pairs plots

Los pares de gráficos (matrices de gráficos de dispersión) pueden ser calculados para cada par de un set de variables.

```
> splom(smallHELP, cex=0.3)
```



El paquete **GGally** ofrece una opción para gráficos de pares más elaborados.

5.4 Regresión lineal simple

Los modelos de regresión lineal simple son descritos en detalle en *Start Modeling with R*. Estos usan con la misma notación de fórmula introducida con anterioridad para resúmenes numéricos y gráficos, especificando la respuesta y los predictores. Aquí consideraremos ajustar el modelo `cesdmcs ~`

```
> cesdmodel <- lm(cesd ~ mcs, data=females)
> coef(cesdmodel)
```

```
(Intercept)      mcs
    57.349      -0.707
```

Para simplificar la salida, desactivamos la opción de desplegar las estrellas de significancia.

Tenemos la tendencia a introducir la regresión lineal de manera pronta en nuestros cursos, como una técnica simplemente descriptiva.

Es importante escoger buenos nombres para objetos de tipo modelo. Aquí la salida de `lm()` es guardada como `cesdmodel`, que denota que es un modelo de regresión a las puntuaciones de síntomas de depresión


```

> options(show.signif.stars=FALSE)
> coef(cesdmodel)

(Intercept)      mcs
      57.349      -0.707

> msummary(cesdmodel)

              Estimate Std. Error t value Pr(>|t|)
(Intercept)  57.3485      2.3806   24.09 < 2e-16
mcs          -0.7070      0.0757   -9.34 1.8e-15

Residual standard error: 9.66 on 105 degrees of freedom
Multiple R-squared:  0.454,      Adjusted R-squared:  0.449
F-statistic: 87.3 on 1 and 105 DF,  p-value: 1.81e-15

> coef(summary(cesdmodel))

              Estimate Std. Error t value  Pr(>|t|)
(Intercept)   57.349      2.38062   24.090 1.425e-44
mcs           -0.707      0.07566   -9.344 1.813e-15

> confint(cesdmodel)

              2.5 % 97.5 %
(Intercept) 52.6282 62.069
mcs         -0.8571 -0.557

> rsquared(cesdmodel)

[1] 0.454

> class(cesdmodel)

[1] "lm"

```

La salida de un `lm()` es un objeto de modelo lineal. Hay unas cuantas funciones que pueden operar estos objetos, como se vio previamente con `coef()`. La función `residuals()` devuelve un vector de residuales

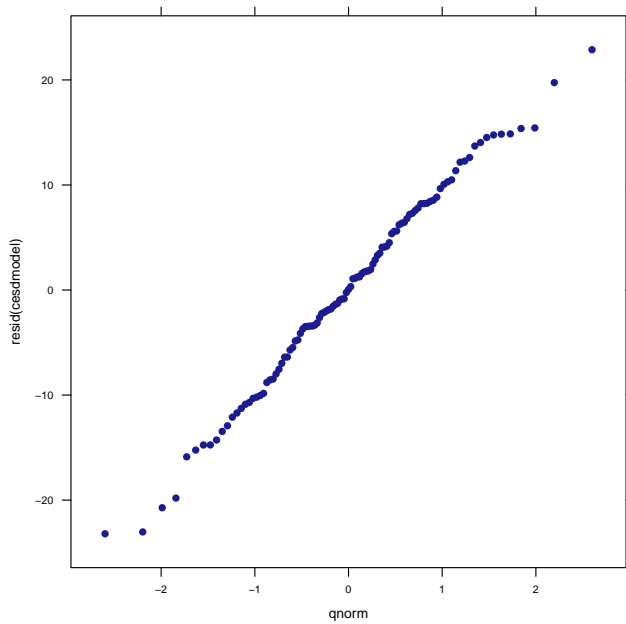
```

> histogram(~ residuals(cesdmodel), density=TRUE)

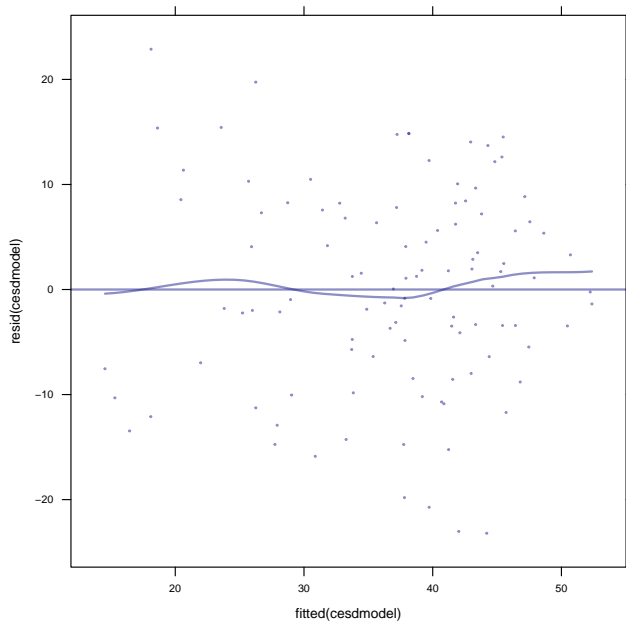
> qqmath(~ resid(cesdmodel))

```

La función `residuals()` puede ser abreviada `resid()`. Otra función útil es `fitted()`, que muestra un vector de los valores predichos.

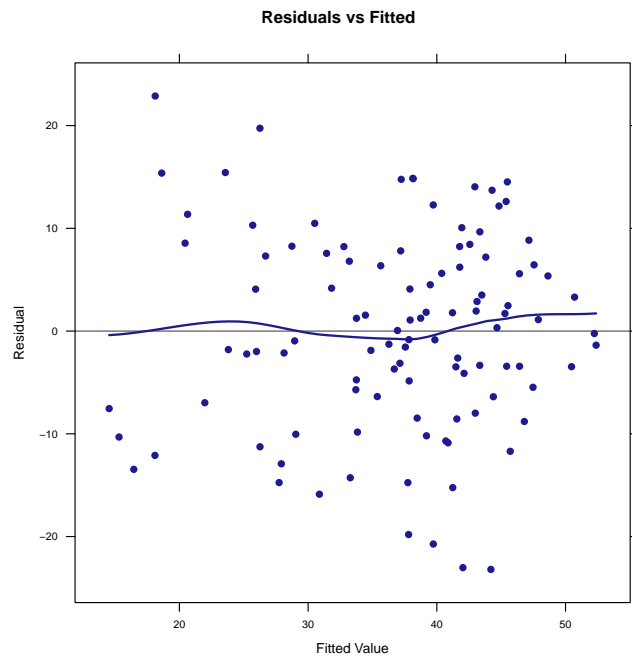


```
> xyplot(resid(cesdmodel) ~ fitted(cesdmodel),
  type=c("p", "smooth", "r"),
  alpha=0.5, cex=0.3, pch=20)
```



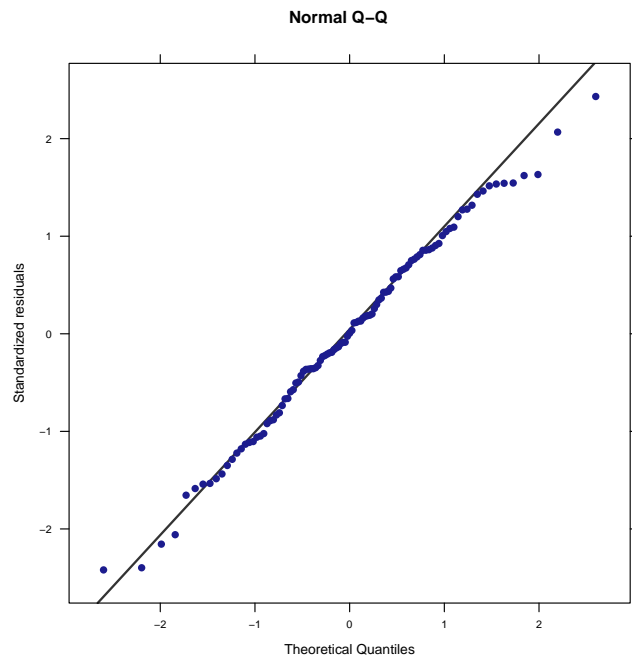
La función `mpplot()` puede facilitar crear una variedad de gráficos útiles, incluyendo los gráficos de dispersión de residuales y ajustados, especificando la opción (`which=1`).

```
> mplot(cesdmodel, which=1)
[[1]]
```



También puede generar un gráfico cuantilo-cuantilo para una distribución normal (`which=2`),

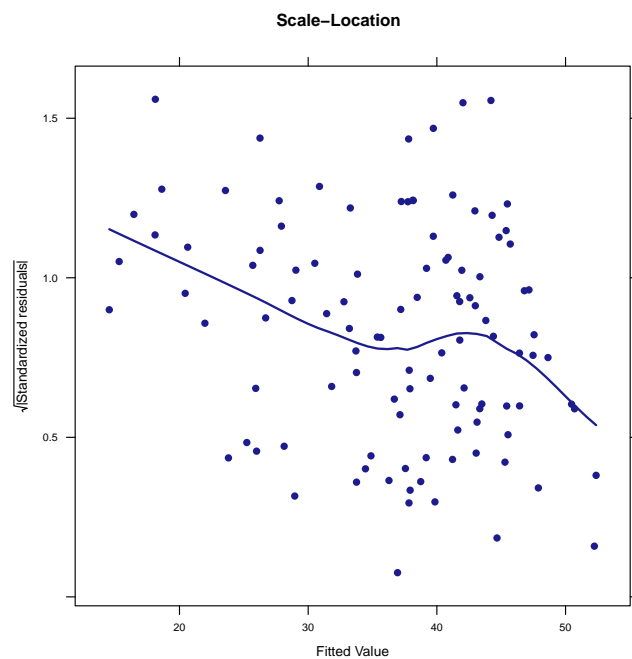
```
> mplot(cesdmodel, which=2)
[[1]]
```



Un gráfico de escala y locación,

```
> mplot(cesdmodel, which=3)
```

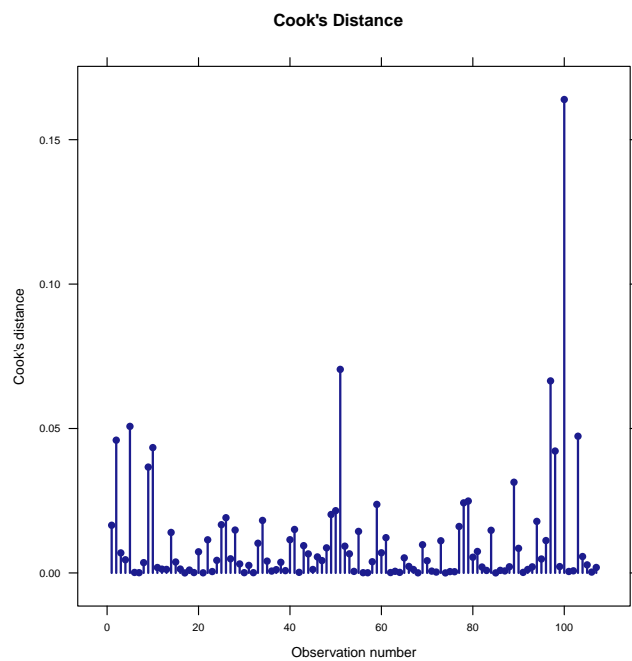
```
[[1]]
```



Distancia de Cook por número de observación,

```
> mplot(cesdmodel, which=4)
```

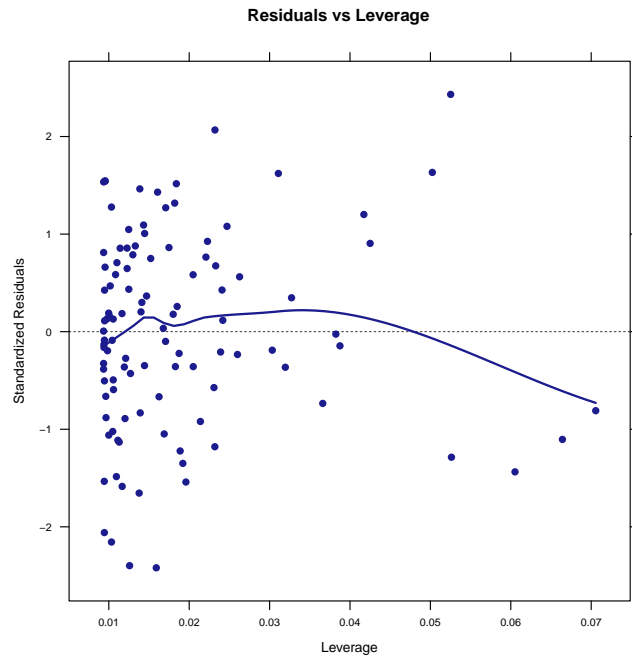
```
[[1]]
```



residuals vs.leverage,

```
> mplot(cesdmodel, which=5)
```

```
[[1]]
```



6

Dos variables categóricas

6.1 Tablas de clasificación cruzada (o de contingencia)

Las tablas de clasificaciones cruzadas (two-way o R por C) pueden ser construidas para una (o más) variables categóricas. Aquí consideramos la tabla de contingencia para el status de homeless (sin casa por una o más noches en los últimos 6 meses o con casa) y sexo.

```
> tally(~ homeless + sex, margins=FALSE, data=HELPrct)
```

	sex	
homeless	female	male
homeless	40	169
housed	67	177

También podemos calcular los porcentajes de columna.

```
> tally(~ sex | homeless, margins=TRUE, format="percent",  
  data=HELPrct)
```

	homeless	
sex	homeless	housed
female	19.14	27.46
male	80.86	72.54
Total	100.00	100.00

Podemos calcular el radio de propensión (odds ratio) directamente de la tabla:

```
> OR <- (40/169)/(67/177); OR
```

```
[1] 0.6253
```

El paquete `mosaic` tiene una función que calcula el ratio de propensión:

```
> oddsRatio(tally(~ (homeless=="housed") + sex, margins=FALSE,
  data=HELPrct))
```

```
[1] 0.6253
```

La función `CrossTable()` en el paquete `gmodels` también despliega una tabla de clasificación cruzada.

```
> require(gmodels)
> with(HELPrct, CrossTable(homeless, sex,
  prop.r=FALSE, prop.chisq=FALSE, prop.t=FALSE))
```

```
Cell Contents
|-----|
|               N |
|      N / Col Total |
|-----|
```

Total Observations in Table: 453

	sex		
homeless	female	male	Row Total
homeless	40	169	209
	0.374	0.488	
housed	67	177	244
	0.626	0.512	
Column Total	107	346	453
	0.236	0.764	

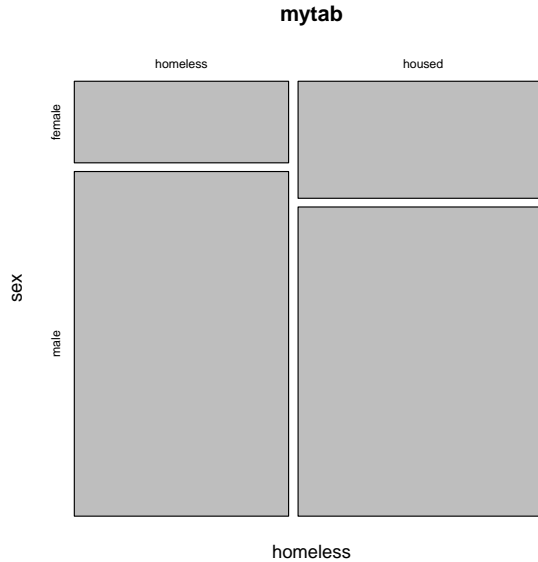
Los resúmenes gráficos de las tablas de clasificación cruzada pueden ser útiles para visualizar asociaciones. Los gráficos de `mosaic` son un ejemplo, donde el total (todas las observaciones) es proporcional a uno. Aquí, vemos como los hombres tienden a ser sobrepresentados entre los sujetos sin casa (como se representa en la línea horizontal, la cual es mayor para los sin casa que para los con casa)

PRECAUCIÓN!

El jurado está aún preocupado de la utilidad de los gráficos de `mosaic` (también conocidos como *eikosograms*) en cuanto a la presentación de gráficos con pocos datos. Hemos encontrado gran ayuda en ellos para mejorar el entendimiento de una tabla de contingencia de dos vías.

E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 2nd edition, 2001.


```
> mytab <- tally(~ homeless + sex, margins=FALSE,
  data=HELPrct)
> mosaicplot(mytab)
```



6.2 Creando tablas de resumen de estadísticas

Las tablas pueden ser creadas para estadísticas de resumen usando la función `do()`.

```
> HELPtable <- rbind(
  do(40) * data.frame(sex="female", homeless="homeless"),
  do(169) * data.frame(sex="male", homeless="homeless"),
  do(67) * data.frame(sex="female", homeless="housed"),
  do(177) * data.frame(sex="male", homeless="housed")
)
> tally(~ homeless + sex, data=HELPtable)
```

```
      sex
homeless female male
homeless    40  169
housed      67  177
```

6.3 Pruebas de ji-cuadrado

```
> chisq.test(tally(~ homeless + sex, margins=FALSE,
  data=HELPrct), correct=FALSE)
```

Pearson's Chi-squared test

```
data:  tally(~homeless + sex, margins = FALSE, data = HELPrct)
X-squared = 4.3, df = 1, p-value = 0.04
```

Encontramos una asociación estadística significativa: es poco verosímil que observáramos una asociación así de fuerte si el estado de vivir en una casa y el sexo fueran independientes en la población

Cuando los estudiantes descubren una asociación significativa, es importante que sean capaces de interpretar en el contexto del problema. La función `xchisq.test()` proporciona detalles adicionales (observados, esperados, contribución al estadístico y residual) para ayudar en este proceso.

```
> xchisq.test(tally(~homeless + sex, margins=FALSE,
  data=HELPrct), correct=FALSE)
```

x es de eXtra.

Pearson's Chi-squared test

```
data:  x
X-squared = 4.3, df = 1, p-value = 0.04
```

```
      40      169
( 49.37) (159.63)
 [1.78]  [0.55]
<-1.33>  < 0.74>
```

```
      67      177
( 57.63) (186.37)
 [1.52]  [0.47]
< 1.23>  <-0.69>
```

key:

```
  observed
 (expected)
 [contribution to X-squared]
 <Pearson residual>
```

Aquí observamos que hay menos mujeres indigentes, y más hombres de los que se esperaría.

6.4 *Prueba exacta de Fisher*

Una prueba exacta de Fisher puede también ser calculada. Este cálculo está dirigido a tablas de 2 por 2. Existen opciones para solventar el problema del límite de tamaño para tablas más grandes (vea `?fisher.test()`).

```
> fisher.test(tally(~homeless + sex, margins=FALSE,
  data=HELPrct))
```

Fisher's Exact Test for Count Data

```
data:  tally(~homeless + sex, margins = FALSE, data = HELPrct)
p-value = 0.05
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.3895 0.9968
sample estimates:
odds ratio
 0.6259
```

CAVANDO HONDO

Note la diferencia en el estimado del radio de propensión del visto en la sección 6.1. La función `fisher.test()` usa un estimador diferente (y un intervalo diferente en el perfil de la verosimilitud).

7

Respuesta cuantitativa, predictor categórico

7.1 Un predictor dicotómico: resúmenes gráficos y numéricos

Aquí vamos a comparar las distribuciones de las puntuaciones de CESD por sexo. La función `mean()` puede usarse para calcular la media del puntaje de CESD separado para hombres y mujeres.

```
> mean(cesd ~ sex, data=HELPrct)
```

```
female    male  
 36.89   31.60
```

La función `favstats()` puede mostrar más estadísticas por grupo.

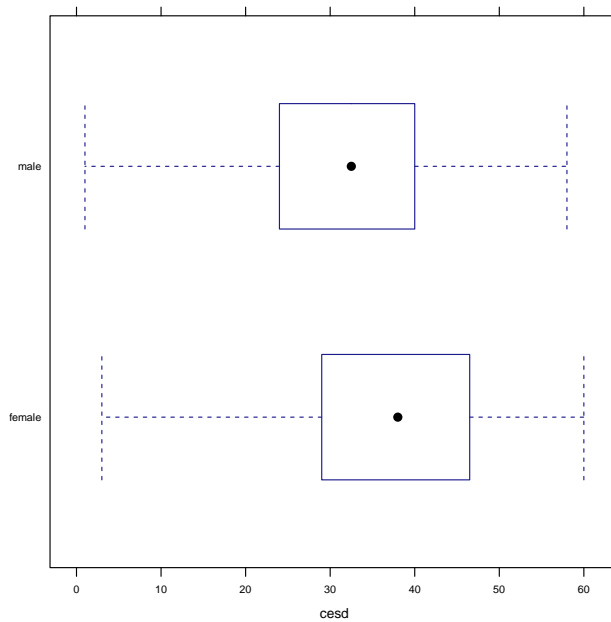
```
> favstats(cesd ~ sex, data=HELPrct)
```

	sex	min	Q1	median	Q3	max	mean	sd	n	missing
1	female	3	29	38.0	46.5	60	36.89	13.02	107	0
2	male	1	24	32.5	40.0	58	31.60	12.10	346	0

Los diagramas de cajas son una presentación gráfica particularmente útil para comparar distribuciones. La función `bwplot()` puede ser utilizada para desplegar los diagramas de cajas de los puntajes de CESD separados por sexo. Vemos de ambos resúmenes, numérico y gráfico, que las mujeres tienden a tener un puntaje un poco mayor en el CESD que los hombres.

Aunque usualmente ponemos variables explicatorias alrededor del eje horizontal, en ocasiones el ajuste de la página la envía a otra orientación preferible en estos gráficos

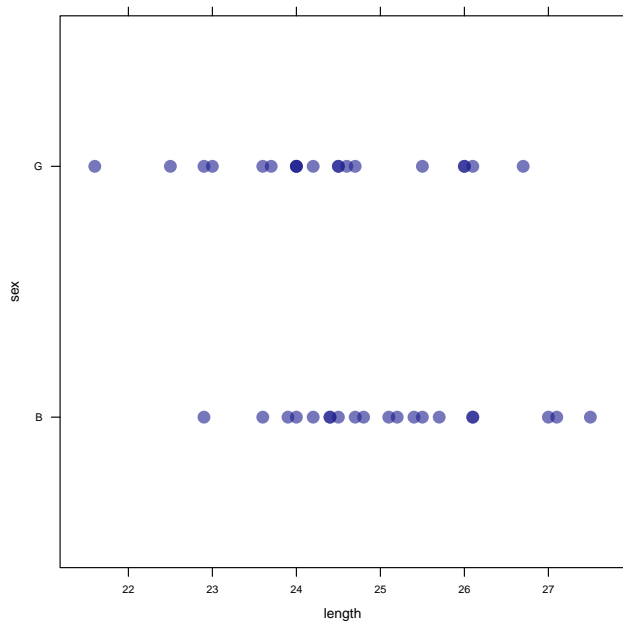
```
> bwplot(sex ~ cesd, data=HELPrct)
```



Cuando los tamaños de muestra son pequeños, no hay razón para resumirlos con un diagrama de cajas puesto que `xyplot()` puede manejar predictores categóricos. Incluso con 10-20 observaciones en un grupo, un gráfico de dispersión es bastante entendible. Ajustar un valor del nivel de alpha ayuda a detectar observaciones múltiples con el mismo valor.

Una vez, uno de nosotros vio a un biólogo presentar orgullosamente unos diagramas de caja de lado a lado. Pensando que esta era una victoria mayor, inocentemente se le preguntó cuántas observaciones había en cada grupo: cuatro respondió el biólogo.

```
> xyplot(sex ~ length, alpha=.6, cex=1.4, data=KidsFeet)
```



7.2 *Un predictor dicotómico: t de dos muestras*

La prueba de dos muestras de Student puede ser ejecutada sin (predeterminada) asumir o asumiendo una igualdad de varianza.

```
> t.test(cesd ~ sex, var.equal=FALSE, data=HELPrct)
```

Welch Two Sample t-test

```
data: cesd by sex
t = 3.7, df = 170, p-value = 3e-04
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 2.493 8.087
sample estimates:
mean in group female   mean in group male
          36.89             31.60
```

Podemos ver que hay una diferencia significativa entre dos grupos.

Podemos repetirla suponiendo igualdad de varianzas.

```
> t.test(cesd ~ sex, var.equal=TRUE, data=HELPrct)
```

Two Sample t-test

```
data:  cesd by sex
t = 3.9, df = 450, p-value = 1e-04
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 2.610 7.969
sample estimates:
mean in group female   mean in group male
          36.89             31.60
```

Los grupos pueden ser comparados también utilizando la función `lm()` (también bajo el supuesto de igualdad). El comando de mosaic `msummary()` ofrece una versión un poco más tersa de la salida típica de `summary()`

```
> msummary(lm(cesd ~ sex, data=HELPrct))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	36.89	1.19	30.96	< 2e-16
sexmale	-5.29	1.36	-3.88	0.00012

```
Residual standard error: 12.3 on 451 degrees of freedom
Multiple R-squared:  0.0323,      Adjusted R-squared:  0.0302
F-statistic: 15.1 on 1 and 451 DF,  p-value: 0.00012
```

7.3 Prueba de 2 grupos no paramétrica

La misma conclusión puede ser obtenida usando una prueba no paramétrica (suma de rangos de Wilcoxon).

```
> wilcox.test(cesd ~ sex, data=HELPrct)
```

```
Wilcoxon rank sum test with continuity correction
```

La función `lm()` es parte de una estructura más flexible de modelaje, mientras el `t.test()` es fundamentalmente una calle sin salida. `lm()` usa el supuesto de la igualdad de varianzas. Vea el libro, *Start Modeling in R* para más detalles. .

```
data:  cesd by sex
W = 23000, p-value = 1e-04
alternative hypothesis: true location shift is not equal to 0
```

7.4 La prueba con la permutación

Aquí, extendemos los métodos introducidos en la sección 3.6 para llevar a cabo una prueba de dos colas comparando

las edades por género. Primero, calculamos la diferencia observada en las medias

```
> mean(age ~ sex, data=HELPrct)

female  male
 36.25  35.47

> test.stat <- diffmean(age ~ sex, data=HELPrct)
> test.stat

diffmean
-0.7841
```

Podemos calcular el mismo estadístico después de mezclar los niveles de los grupos:

```
> do(1) * diffmean(age ~ shuffle(sex), data=HELPrct)

diffmean
1  0.2682

> do(1) * diffmean(age ~ shuffle(sex), data=HELPrct)

diffmean
1  0.03568

> do(3) * diffmean(age ~ shuffle(sex), data=HELPrct)

diffmean
1 -0.5149
2 -0.6128
3  0.3049
```

```
> rtest.stats <- do(500) * diffmean(age ~ shuffle(sex),
  data=HELPrct)
> head(rtest.stats, 3)

diffmean
1 -0.14786
2  0.31711
3  0.04792
```

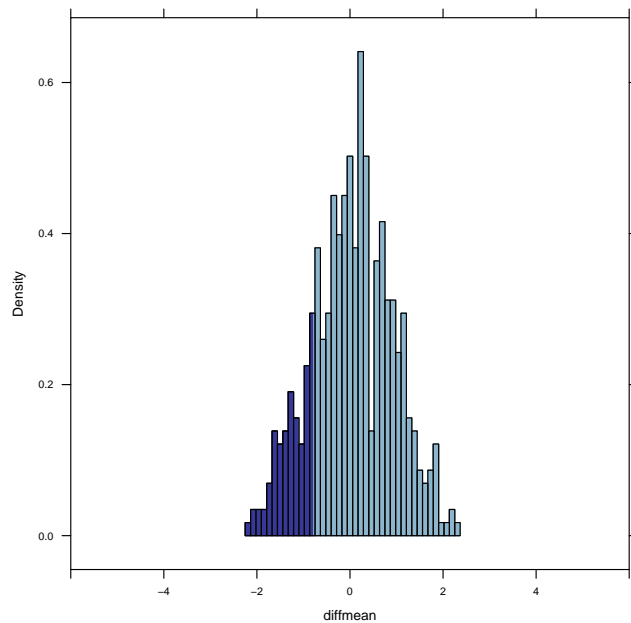
```
> favstats(~ diffmean, data=rtest.stats)

   min    Q1  median    Q3   max   mean    sd  n missing
-2.204 -0.5272 0.06015 0.6475 2.299 0.05017 0.8714 500      0
```

CAVANDO HONDO

Más detalles pueden ser encontrados en el paquete *mosaic*,
Resampling vignette

```
> histogram(~ diffmean, n=40, xlim=c(-6, 6),
  groups=diffmean >= test.stat, pch=16, cex=.8,
  data=rtest.stats)
> ladd(panel.abline(v=test.stat, lwd=3, col="red"))
```

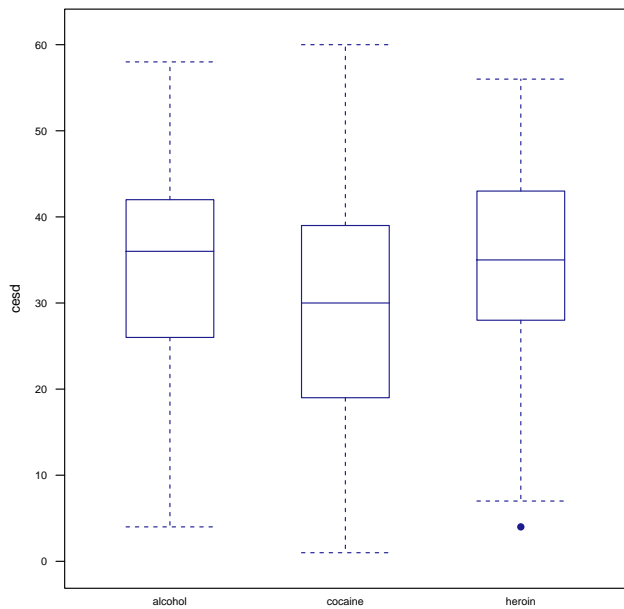


Aquí no vemos mucha evidencia que contradiga la hipótesis nula de que los hombres y las mujeres tienen la misma media de edad en la población.

7.5 *Análisis de varianza de una vía*

Las primeras comparaciones fueron entre dos grupos. Podemos también considerar probar las diferencias entre tres o más grupos usando un ANOVA de una vía. Aquí vamos a comparar los puntajes de CESD por sustancia primaria de abuso (heroína, cocaína o alcohol) con una línea en lugar de un punto para indicar la mediana.

```
> bwplot(cesd ~ substance, pch="|", data=HELPrct)
```



```
> mean(cesd ~ substance, data=HELPrct)
```

```
alcohol cocaine heroin
 34.37   29.42   34.87
```

```
> anovamod <- aov(cesd ~ substance, data=HELPrct)
> summary(anovamod)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
substance	2	2704	1352	8.94	0.00016
Residuals	450	68084	151		

Aunque las puntuaciones todavía altas (puntajes de 16 o más son generalmente considerados como síntomas "severos"), el grupo envuelto en cocaína tiende a tener los puntajes más bajos que aquellos cuya sustancia primaria es el alcohol o la heroína.

```
> modintercept <- lm(cesd ~ 1, data=HELPrct)
> modsubstance <- lm(cesd ~ substance, data=HELPrct)
```

El comando `anova()` puede resumir los modelos.

```
> anova(modsubstance)
```

Analysis of Variance Table

Response: cesd

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
substance	2	2704	1352	8.94	0.00016
Residuals	450	68084	151		

En este caso los resultados son idénticos (puesto que solo tenemos un predictor, con 2 grados de libertad)

La función `anova()` puede también ser utilizada para formalmente comparar dos modelos (anidados).

```
> anova(modintercept, modsubstance)
```

Analysis of Variance Table

Model 1: cesd ~ 1

Model 2: cesd ~ substance

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	452	70788				
2	450	68084	2	2704	8.94	0.00016

7.6 Las diferencias significativas honestas de Tukey

Hay una variedad de procedimientos de comparaciones múltiples que pueden ser utilizados después de ajustar el modelo ANOVA. Una de estas son las Diferencias significativas honestas de Tukey (HSD por las siglas en inglés). Otras opciones están disponibles dentro del paquete `multcomp`.

```
> favstats(cesd ~ substance, data=HELPrct)
```

	substance	min	Q1	median	Q3	max	mean	sd	n	missing
1	alcohol	4	26	36	42	58	34.37	12.05	177	0
2	cocaine	1	19	30	39	60	29.42	13.40	152	0
3	heroin	4	28	35	43	56	34.87	11.20	124	0

```
> HELPrct <- mutate(HELPrct, subgrp = factor(substance,
  levels=c("alcohol", "cocaine", "heroin"),
  labels=c("A", "C", "H")))
> mod <- lm(cesd ~ subgrp, data=HELPrct)
> HELPHSD <- TukeyHSD(mod, "subgrp")
> HELPHSD
```

Tukey multiple comparisons of means
95% family-wise confidence level

Fit: aov(formula = x)

\$subgrp

	diff	lwr	upr	p adj
C-A	-4.9518	-8.150	-1.753	0.0009
H-A	0.4981	-2.889	3.885	0.9362
H-C	5.4499	1.950	8.950	0.0008

> *mpplot(HELPHSD)*

De nuevo, podemos ver que el grupo de cocaína tiene significativamente menores puntajes de CESD que cualquiera de los otros grupos.

8

Respuesta categórica, predicto cuantitativo

8.1 Respuesta categórica, predictor cuantitativo

La regresión logística está disponible en R usando la función `glm()`, la cual permite una variedad de funciones de enlace y formas de distribuciones para modelos lineales generalizados, incluyendo la regresión logística.

La función `glm()` tiene el argumento `family`, el cual puede tomar la opción `link` (función de enlace). La función de enlace `logit` está predeterminada en `link` para la familia binomial, entonces no necesitamos especificarlo aquí. La forma más completa de uso sería `family=binomial(link=logit)`.

```
> logitmod <- glm(homeless ~ age + female, family=binomial,
  data=HELPrct)
> msummary(logitmod)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.8926	0.4537	1.97	0.049
age	-0.0239	0.0124	-1.92	0.055
female	0.4920	0.2282	2.16	0.031

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 625.28 on 452 degrees of freedom
 Residual deviance: 617.19 on 450 degrees of freedom
 AIC: 623.2

Number of Fisher Scoring iterations: 4

```
> exp(coef(logitmod))
```

	age	female
(Intercept)	2.4415	1.6355

```
> exp(confint(logitmod))
```

	2.5 %	97.5 %
(Intercept)	1.0081	5.988
age	0.9527	1.000
female	1.0501	2.574

Podemos comparar dos modelos (para una prueba de múltiples grados de libertad). Por ejemplo, podríamos estar interesados en la asociación del estado de indigencia (homeless) y la edad para cada uno de los tres grupos de sustancia.

```
> mymodsubage <- glm((homeless=="homeless") ~ age + substance,
  family=binomial, data=HELPrct)
> mymodage <- glm((homeless=="homeless") ~ age, family=binomial,
  data=HELPrct)
> msummary(mymodsubage)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.0509	0.5164	-0.10	0.9215
age	0.0100	0.0129	0.77	0.4399
substancecocaine	-0.7496	0.2303	-3.25	0.0011
substanceheroin	-0.7780	0.2469	-3.15	0.0016

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 625.28 on 452 degrees of freedom
 Residual deviance: 607.62 on 449 degrees of freedom
 AIC: 615.6

Number of Fisher Scoring iterations: 4

```
> exp(coef(mymodsubage))
```

(Intercept)	age	substancecocaine	substanceheroin
0.9504	1.0101	0.4725	0.4593

```
> anova(mymodage, mymodsubage, test="Chisq")
```

Analysis of Deviance Table

Model	1: (homeless == "homeless") ~ age	2: (homeless == "homeless") ~ age + substance
Resid. Df	451	449
Resid. Dev	622	608
Df		2
Deviance		14.3
Pr(>Chi)		0.00078

Observamos que los grupos que consumen cocaína y heroína son significativamente menos probables a estar sin casa que aquellos sujetos envueltos en el alcohol, después de controlar la edad. (Un resultado similar se ve considerando solamente el status de si tiene vivienda y la sustancia).

```
> tally(~ homeless / substance, format="percent", margins=TRUE, data=HELPrct)
```

	substance		
homeless	alcohol	cocaine	heroin
homeless	58.19	38.82	37.90
housed	41.81	61.18	62.10
Total	100.00	100.00	100.00

9

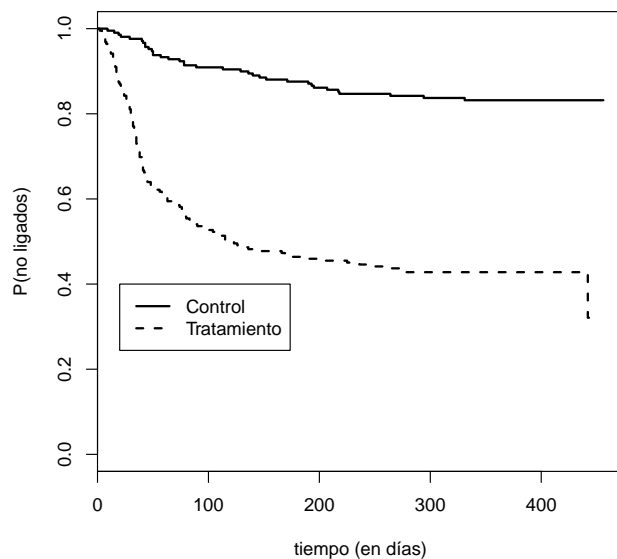
Resulados de análisis de supervivencia

Una extensiva cantidad de herramientas para análisis de supervivencia (tiempo a un evento) está disponible en el paquete `survival`

9.1 Diagrama de Kaplan-Meier

```
> require(survival)
> fit <- survfit(Surv(dayslink, linkstatus) ~ treat,
  data=HELPrct)
> plot(fit, conf.int=FALSE, lty=1:2, lwd=2,
  xlab="tiempo (en días)", ylab="P(no ligados)")
> legend(20, 0.4, legend=c("Control", "Tratamiento"),
  lty=c(1,2), lwd=2)
> title("Producto-Límite de los estimados de supervivencia(tiempo ligado)")
```

Producto-Límite de los estimados de sobrevivencia(tiempo liga



Vemos que los sujetos en el tratamiento (Health Evaluation and Linkage to Primary Care clinic) tuvieron significativamente más probabilidad de acoplarse al cuidado primario (menos probabilidad a "sobrevivir") que el grupo control (cuidado normal)

9.2 Modelo de riesgos proporcionales de Cox

```
> require(survival)
> summary(coxph(Surv(dayslink, linkstatus) ~ age + substance,
  data=HELPrct))
```

Call:

```
coxph(formula = Surv(dayslink, linkstatus) ~ age + substance,
  data = HELPrct)
```

```
n= 431, number of events= 163
(22 observations deleted due to missingness)
```

	coef	exp(coef)	se(coef)	z	Pr(> z)
age	0.00893	1.00897	0.01026	0.87	0.38
substancecocaine	0.18045	1.19775	0.18100	1.00	0.32
substanceheroin	-0.28970	0.74849	0.21725	-1.33	0.18

	exp(coef)	exp(-coef)	lower .95	upper .95
age	1.009	0.991	0.989	1.03
substancecocaine	1.198	0.835	0.840	1.71
substanceheroin	0.748	1.336	0.489	1.15

Concordance= 0.55 (se = 0.023)

Rsquare= 0.014 (max possible= 0.988)

Likelihood ratio test= 6.11 on 3 df, p=0.106

Wald test = 5.84 on 3 df, p=0.12

Score (logrank) test = 5.91 on 3 df, p=0.116

Ni la edad ni el grupo de sustancia fueron significativamente asociados a estar en cuidado primario.

10

Más de dos variables

10.1 Análisis de varianza (ANOVA) de dos (o más) vías

Podemos ajustar un modelo ANOVA de dos o más vías, con o sin interacción, usando la misma sintaxis de modelaje.

```
> HELPrct <- mutate(HELPrct, subgrp = factor(substance,
  levels=c("alcohol", "cocaine", "heroin"),
  labels=c("A", "C", "H")))
> median(cesd ~ substance | sex, data=HELPrct)
```

alcohol.female	cocaine.female	heroin.female	alcohol.male
40.0	35.0	39.0	33.0
cocaine.male	heroin.male	female	male
29.0	34.5	38.0	32.5

```
> bwplot(cesd ~ subgrp | sex, data=HELPrct)
> summary(aov(cesd ~ substance + sex, data=HELPrct))
```

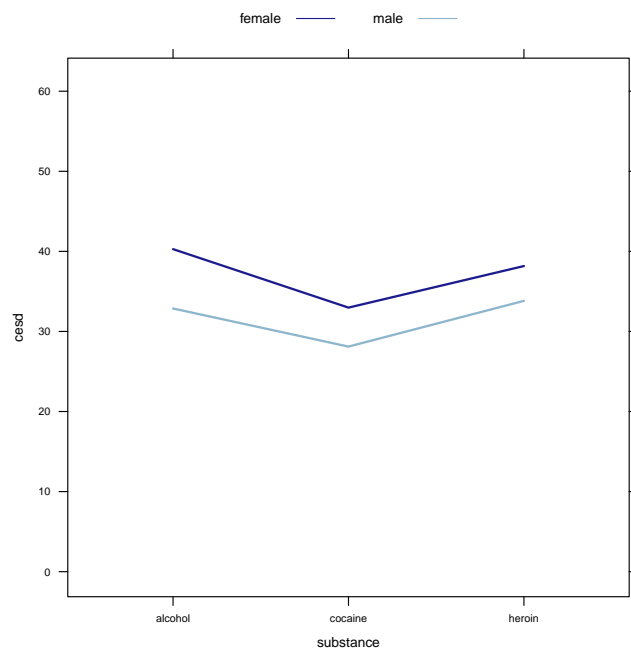
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
substance	2	2704	1352	9.27	0.00011
sex	1	2569	2569	17.61	3.3e-05
Residuals	449	65515	146		

```
> summary(aov(cesd ~ substance * sex, data=HELPrct))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
substance	2	2704	1352	9.25	0.00012
sex	1	2569	2569	17.57	3.3e-05
substance:sex	2	146	73	0.50	0.60752
Residuals	447	65369	146		

Hay poca evidencia de interacción, aunque hay efectos principales significativos en los términos de la sustancia(`substance`) y el sexo(`sex`).

```
> xyplot(cesd ~ substance, groups=sex, lwd=2,
  auto.key=list(columns=2, lines=TRUE, points=FALSE),
  type='a', data=HELPrct)
```



10.2 Regresión múltiple

La regresión múltiple es una extensión lógica de comandos previamente señalados, donde algunos predictores adicionales son agregados. Esto permite a los estudiantes intentar desenvolver relaciones multivariadas.

Aquí consideramos un modelo (de pendientes paralelas) para los síntomas depresivos como función del Puntaje del componente mental (MCS), edad (en años) y sexo del sujeto.

Tenemos la tendencia introducir la regresión lineal múltiple de forma temprana en nuestros cursos como una técnica puramente descriptiva, después regularmente volvemos a esta. La motivación para esto se encuentra descrita en detalle en el volumen acompañante *Start modelling with R*

```
> lmnointeract <- lm(cesd ~ mcs + age + sex, data=HELPrct)
> msummary(lmnointeract)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	53.8303	2.3617	22.79	<2e-16
mcs	-0.6548	0.0336	-19.50	<2e-16

age	0.0553	0.0556	1.00	0.3200
sexmale	-2.8993	1.0137	-2.86	0.0044

Residual standard error: 9.09 on 449 degrees of freedom

Multiple R-squared: 0.476, Adjusted R-squared: 0.473

F-statistic: 136 on 3 and 449 DF, p-value: <2e-16

También podemos ajustar un modelo que incluye una interacción entre MCS y sexo.

```
> lminteract <- lm(cesd ~ mcs + age + sex + mcs:sex, data=HELPrct)
> msummary(lminteract)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	55.3906	2.9903	18.52	<2e-16
mcs	-0.7082	0.0712	-9.95	<2e-16
age	0.0549	0.0556	0.99	0.324
sexmale	-4.9421	2.6055	-1.90	0.058
mcs:sexmale	0.0687	0.0807	0.85	0.395

Residual standard error: 9.09 on 448 degrees of freedom

Multiple R-squared: 0.477, Adjusted R-squared: 0.472

F-statistic: 102 on 4 and 448 DF, p-value: <2e-16

```
> anova(lminteract)
```

Analysis of Variance Table

Response: cesd

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
mcs	1	32918	32918	398.27	<2e-16
age	1	107	107	1.29	0.2563
sex	1	676	676	8.18	0.0044
mcs:sex	1	60	60	0.72	0.3952
Residuals	448	37028	83		

```
> anova(lmnointeract, lminteract)
```

Analysis of Variance Table

Model 1: cesd ~ mcs + age + sex

Model 2: cesd ~ mcs + age + sex + mcs:sex

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	449	37088				
2	448	37028	1	59.9	0.72	0.4

There is little evidence for an interaction effect, so we drop this from the model.

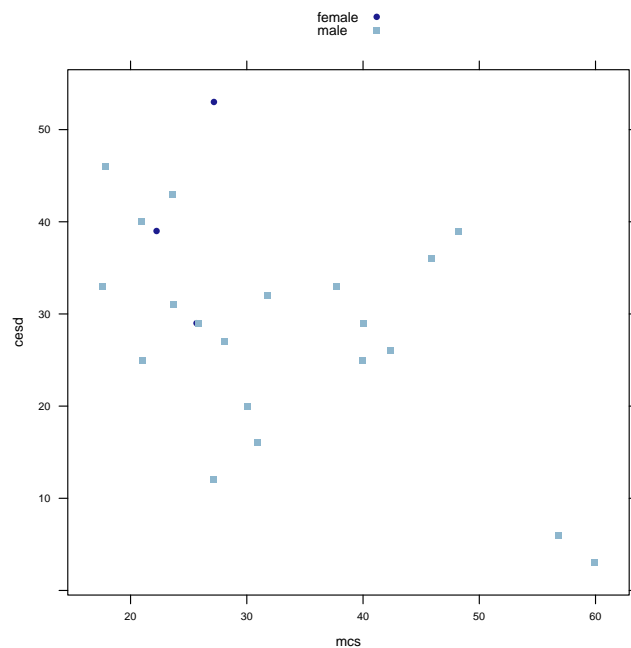
10.2.1 Visualizar los resultados de la regresión

Las funciones `makeFun()` y `plotFun()` del paquete `mosaic` pueden ser utilizadas para desplegar los valores predichos de un modelo de regresión. Por ejemplo, podríamos desplegar los valores de CESD predichos para un rango de valores de MCS (puntuación del componente mental) en un sujeto hipotético, hombre de 36 años y en un sujeto hipotético mujer con la misma edad en el cual se utiliza un modelo pendientes paralelas (sin interacción).

```
> lmfunction <- makeFun(lmnointeract)
```

Ahora podemos graficar los valores predichos separadamente para sujetos masculinos y femeninos sobre valores del rango de MCS (puntaje del componente mental), junto con los datos observados de aquellos de 36 años..

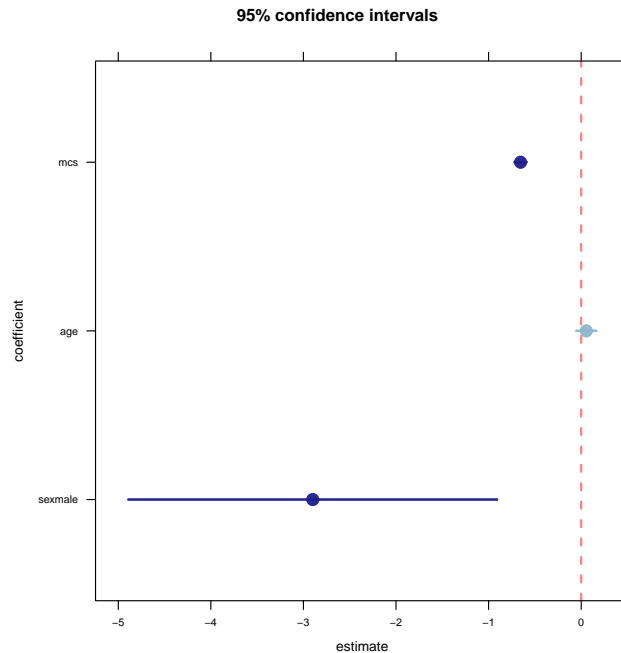
```
> xyplot(cesd ~ mcs, groups=sex, auto.key=TRUE,
  data=filter(HELPrct, age==36))
> plotFun(lmfunction(mcs, age=36, sex="male") ~ mcs,
  xlim=c(0, 60), lwd=2, ylab="predicted CESD", add=TRUE)
> plotFun(lmfunction(mcs, age=36, sex="female") ~ mcs,
  xlim=c(0, 60), lty=2, lwd=3, add=TRUE)
```



10.2.2 Gráficos de coeficientes

Es en ocasiones útil utilizar un gráfico de coeficientes para un modelo de regresión múltiple (junto con sus intervalos de confianza asociados)

```
> mplot(lmnointeract, rows=-1, which=7)
```



Los puntos más oscuros indican que los coeficientes de la regresión donde el intervalo de confianza al 95 % no incluye el valor de 0 de la hipótesis nula

PRECAUCIÓN!

Sea cuidadoso cuando ajusta la regresión con modelos con valores perdidos (ver también sección 13.11)

10.2.3 Diagnósticos de residuales

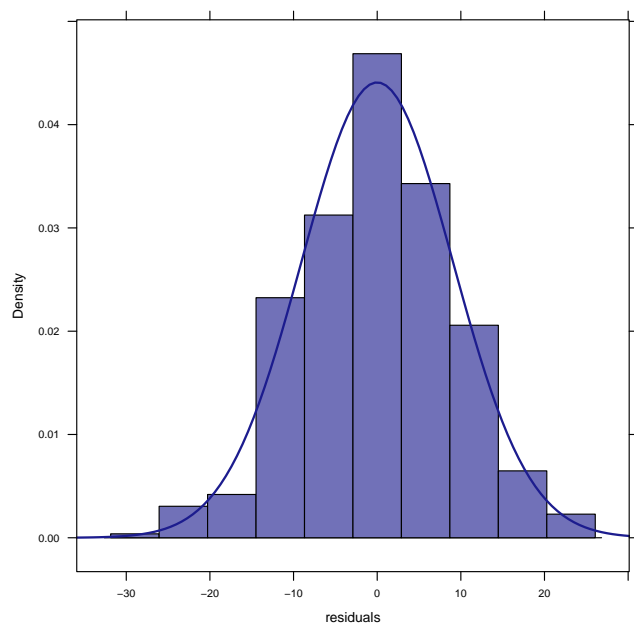
Es sencillo realizar algunos diagnósticos de residuales para este modelo. Empezamos por agregar los valores ajustados y los residuales al conjunto de datos.

```
> HELPrct <- mutate(HELPrct,
  residuals = resid(lmnointeract),
  pred = fitted(lmnointeract))

> histogram(~ residuals, xlab="residuals", fit="normal",
  data=HELPrct)
```

La función `mplot()` también puede ser utilizada para crear estos gráficos.

Aquí agregamos dos nuevas variables en un conjunto de datos existente. Es usualmente una buena práctica dar a los nuevos conjuntos de datos un nuevo nombre.



Podemos identificar que en el subconjunto de observaciones hay residuales extremadamente grandes.

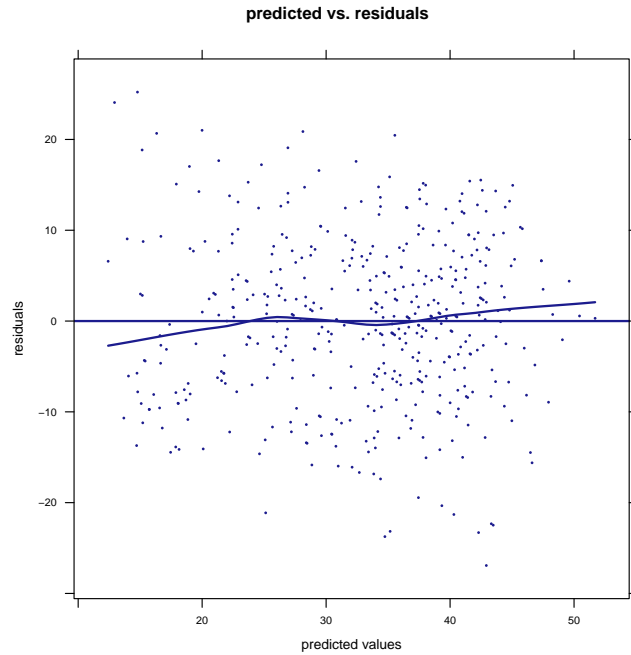
```
> filter(HELPrct, abs(residuals) > 25)
```

```

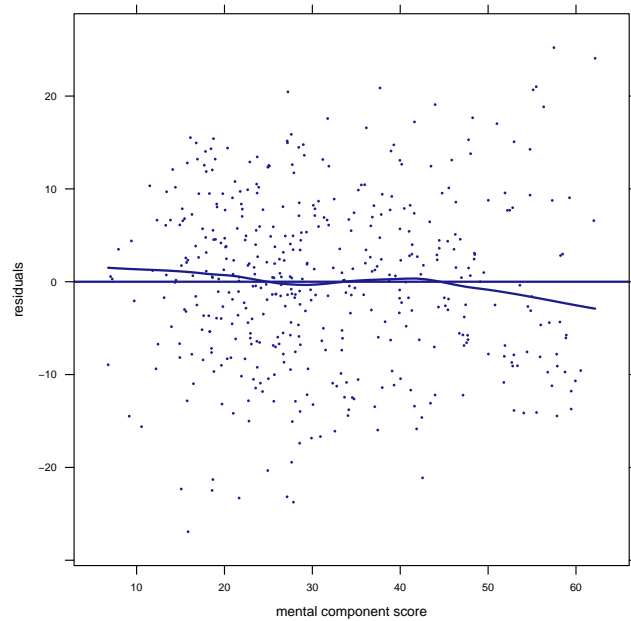
  age anysubststatus anysub cesd d1 daysanysub dayslink drugrisk e2b
1  43                0   no   16 15         191       414         0  NA
2  27                NA  <NA>  40  1          NA       365         3   2
  female sex g1b homeless i1 i2 id indtot linkstatus link  mcs
1      0 male  no homeless 24 36 44    41         0   no 15.86
2      0 male  no homeless 18 18 420   37         0   no 57.49
  pcs pss_fr racegrp satreat sexrisk substance treat subgrp
1 71.39     3  white    no      7  cocaine  yes    C
2 37.75     8  white   yes     3  heroin   no    H
  residuals  pred
1   -26.92 42.92
2    25.22 14.78

```

```
> xyplot(residuals ~ pred, ylab="residuals", cex=0.3,
  xlab="predicted values", main="predicted vs. residuals",
  type=c("p", "r", "smooth"), data=HELPrct)
```



```
> xyplot(residuals ~ mcs, xlab="mental component score",
  ylab="residuals", cex=0.3,
  type=c("p", "r", "smooth"), data=HELPrct)
```



El supuesto de normalidad, linealidad y homoscedasticidad parece razonable aquí.

11

Distribuciones de probabilidad & variables aleatorias

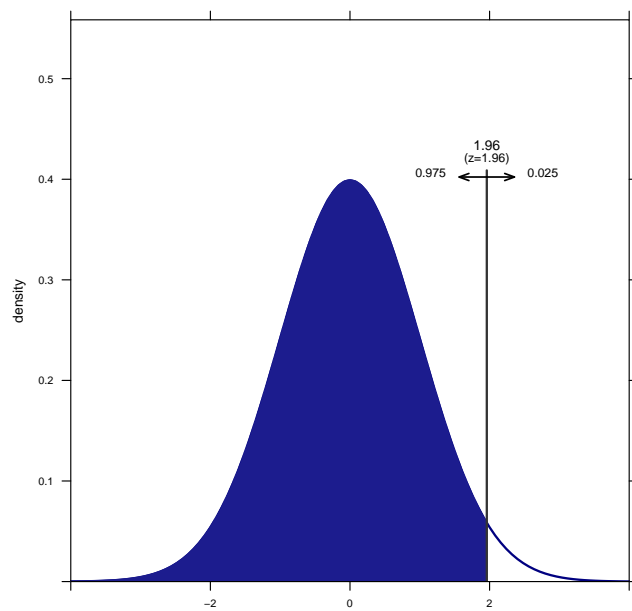
R puede calcular números relacionados a distribuciones de todo tipo. Está también dirigido a generar muestras aleatorias de estas distribuciones, que pueden ser utilizadas para simulación y exploración.

```
> xpnorm(1.96, mean=0, sd=1)    # P(Z < 1.96)
```

If $X \sim N(0, 1)$, then

$$\begin{aligned} P(X \leq 1.96) &= P(Z \leq 1.96) = 0.975 \\ P(X > 1.96) &= P(Z > 1.96) = 0.025 \end{aligned}$$

```
[1] 0.975
```



```
> # valor que satisface  $P(Z < z) = 0.975$ 
> qnorm(.975, mean=0, sd=1)
```

```
[1] 1.96
```

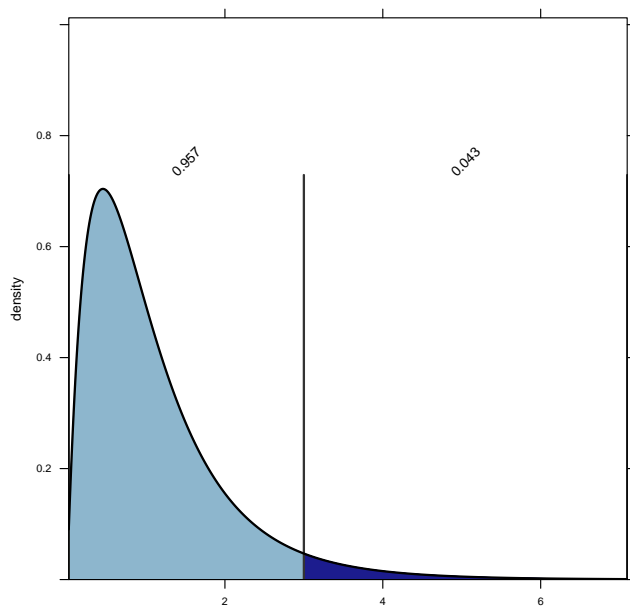
```
> integrate(dnorm, -Inf, 0) #  $P(Z < 0)$ 
```

```
0.5 with absolute error < 4.7e-05
```

Una exposición similar es disponible para la distribución F

```
> xpf(3, df1=4, df2=20)
```

```
[1] 0.9568
```

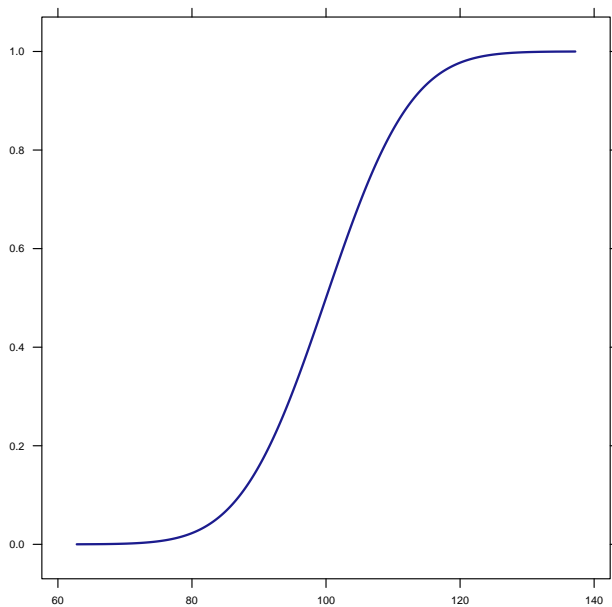



La siguiente tabla muestra los nombres de base para las distribuciones de probabilidad disponibles en la base de R. Estas funciones pueden tener el prefijo por **d** para encontrar la función de densidad de la distribución, **p** para encontrar la distribución acumulada, **q** para encontrar cuantiles y **r** para generar resultados aleatorios. Por ejemplo, para encontrar la función de densidad de una variable aleatoria, use el comando `dexp()`. La función `qDIST()` es el inverso de la función `pDIST()`, para un nombre base dado el nombre de tipo DIST.

Distribución	Nombre de base
Beta	beta
binomial	binom
Cauchy	cauchy
chi-square	chisq
exponential	exp
F	f
gamma	gamma
geometrica	geom
hypergeometrica	hyper
logistica	logis
lognormal	lnorm
binomial negativa	nbinom
normal	norm
Poisson	pois
T de Student	t
Uniforme	unif
Weibull	weibull

La función `plotDist()` puede ser utilizada para desplegar gráficos de varias maneras.

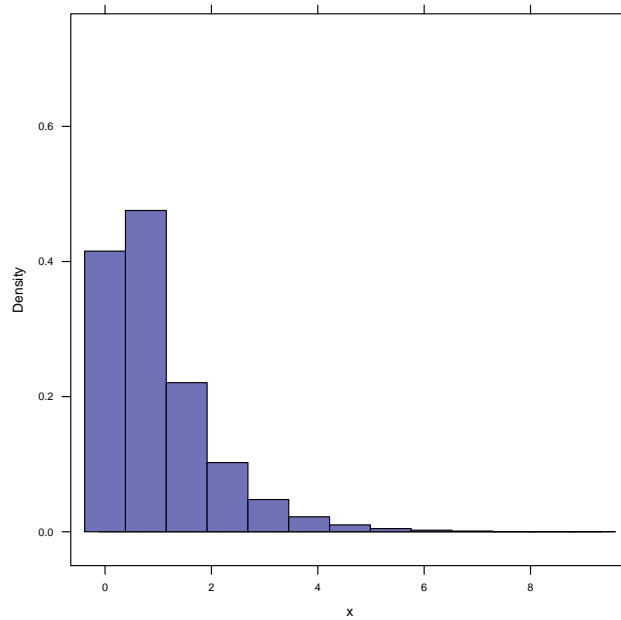
```
> plotDist('norm', mean=100, sd=10, kind='cdf')
```



```
> plotDist('exp', kind='histogram', xlab="x")
```

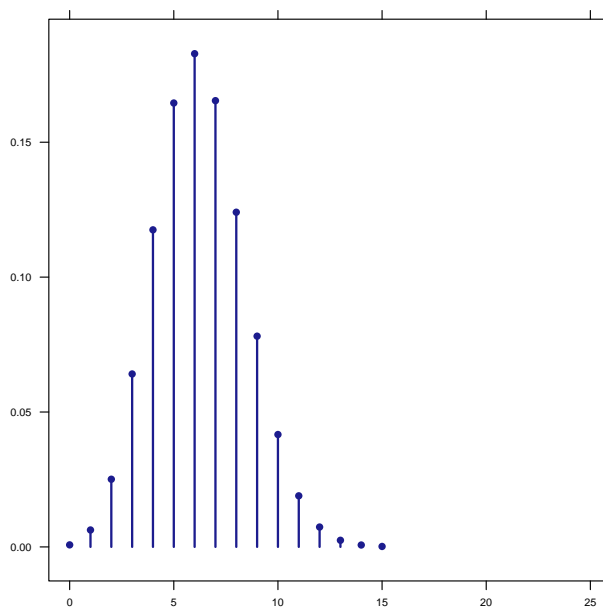
CAVANDO HONDO

El `fitdistr()` dentro del paquete **MASS** facilita la estimación de los parámetros para muchas distribuciones



Note que esto ajusta el parametro del rango a 1 y la siguiente función hace lo mismo:

```
> plotDist('exp', rate=1, kind='histogram', xlab="x")
> plotDist('binom', size=25, prob=0.25, xlim=c(-1,26))
```

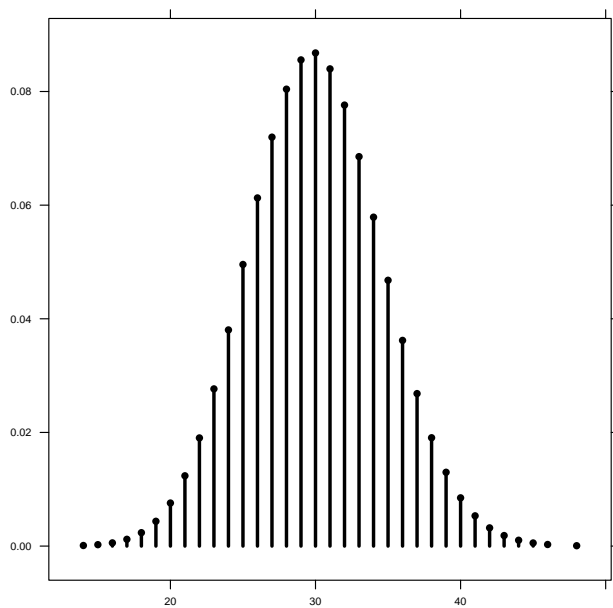


Distribuciones múltiples pueden ser incluidas en el mismo gráfico.

```

> plotDist("binom", size=100, prob=.3, col="black", lwd=3, pch=16)
> plotDist("norm", mean=30, sd=sqrt(100 * .3 * .7),
  groups = abs(x - 30) > 6 , type="h", under=TRUE)

```

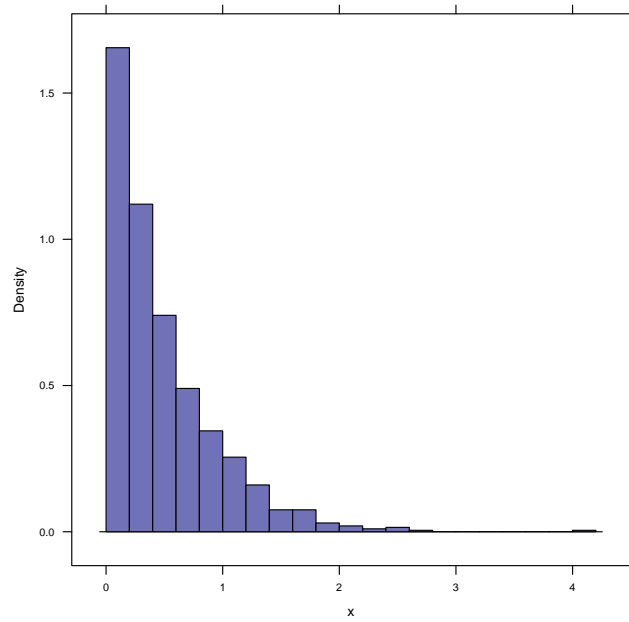


La función `plotFun()` puede ser utilizada para graficar una función arbitraria (en este caso una variable aleatoria exponencial).

```

> f <- makeFun(2*exp(-2*x) ~ x)  # exponential with rate parameter 2
> x <- rexp(1000, rate=2)
> histogram(~ x, width=0.2, center=0.1)
> plotFun(f(x) ~ x, col="red", lwd=2, add=TRUE)

```



12

Cálculos de Potencia

Aunque no es generalmente un tema de trato fundamental en los cursos introductorios, los cálculos de potencia y tamaño de muestra ayudan a reforzar ideas clave de estadística. En esta sección, podremos explorar cómo R puede ser utilizado para llevar a cabo cálculos de potencia, usando un abordaje analítico. Consideramos un problema simple con dos pruebas de hipótesis (la prueba t y la prueba de signos) con comparaciones de una cola.

Vamos a comparar la potencia de la prueba de signos y la potencia de la prueba basada en la teoría normal (prueba t de una cola con una muestra) asumiendo que σ es conocido. Sea X_1, \dots, X_{25} una variables aleatorias independientes distribuidas $N(0,3,1)$ (esto es un alterno para el que deseamos calcular la potencia). Considere probar la hipótesis nula $H_0 : \mu = 0$ contra $H_A : \mu > 0$, con una nivel de significancia de $\alpha = .05$.

12.1 Prueba de signos

Empezamos por calcular el error Tipo I para la prueba de signos. Aquí, queremos rechazar que el número de valores positivos es grande. Bajo la hipótesis nula, esto se distribuye como una variable aleatoria binomial, con $n=25$ intentos y $p=0.5$ probabilidad de valores positivos. Consideremos valores entre 15 y 19.

```
> xvals <- 15:19
> probs <- 1 - pbinom(xvals, size=25, prob=0.5)
> cbind(xvals, probs)

      xvals      probs
[1,]    15 0.114761
```

```
[2,]    16 0.053876
[3,]    17 0.021643
[4,]    18 0.007317
[5,]    19 0.002039
```

```
> qbinom(.95, size=25, prob=0.5)
```

```
[1] 17
```

Entonces, vemos que si decidimos rechazar la hipótesis cuando el número de valores positivos es de 17 o mayor, vamos a tener un nivel de α de `round(1-pbinom(16, 25, 0.5), 3)`, lo cual está cerca del valor nominal en el problema.

Calculamos la potencia de la prueba de signos como se muestra. La probabilidad de que $X_i > 0$, dado que H_A es cierta es dada por:

```
> 1 - pnorm(0, mean=0.3, sd=1)
```

```
[1] 0.6179
```

Esto lo podemos ver gráficamente usando el comando:

```
> xpnorm(0, mean=0.3, sd=1, lower.tail=FALSE)
```

If $X \sim N(0.3, 1)$, then

$$\begin{aligned} P(X \leq 0) &= P(Z \leq -0.3) = 0.3821 \\ P(X > 0) &= P(Z > -0.3) = 0.6179 \end{aligned}$$

```
[1] 0.6179
```

La potencia basada en que la hipótesis alternativa es igual a la probabilidad de conseguir 17 o más valores positivos, dado que $p = 0.6179$, es la siguiente

```
> 1 - pbinom(16, size=25, prob=0.6179)
```

```
[1] 0.3378
```

La potencia es modesta.

12.2 Prueba t

Después, calculamos la potencia de la prueba basada en la teoría normal. Para seguir la comparación justa, vamos a poner nuestro α en un nivel igual a 0.05388


```
> alpha <- 1-pbinom(16, size=25, prob=0.5); alpha
[1] 0.05388
```

Primero encontramos la región de rechazo.

```
> n <- 25; sigma <- 1 # given
> stderr <- sigma/sqrt(n)
> zstar <- qnorm(1-alpha, mean=0, sd=1)
> zstar
[1] 1.608
```

```
> crit <- zstar*stderr
> crit
[1] 0.3217
```

Entonces, rechazamos por las medias observadas, mayores a 0.322.

Para calcular la potencia de esta prueba de una cola, debemos encontrar la probabilidad de la hipótesis alternativa a su punto límite de la derecha

```
> power <- 1 - pnorm(crit, mean=0.3, sd=stderr)
> power
[1] 0.4568
```

La potencia de la prueba basada en la teoría normal es de 0.322. Para dar una revisión (o para futuros cálculos de este tipo) podemos usar la función `power.t.test()`

```
> power.t.test(n=25, delta=.3, sd=1, sig.level=alpha, alternative="one.sided",
  type="one.sample")$power
[1] 0.4408
```

Esta forma analítica (de aproximación basada en la fórmula) permite un estimado similar al valor que calculamos directamente.

Sobre todo, vemos que la prueba t tiene una mayor potencia que la prueba de signos, en especial si los datos son verdaderamente normales.

Calcular la potencia empíricamente demuestra el poder que tienen las simulaciones

13

Manejo de datos

El manejo de datos es una capacidad clave para permitir a los estudiantes (e instructores) a computar con los datos.^o como Diane Lambert de Google ha manifestado, "pensar con los datos". Tendemos a mantener el manejo de datos de los estudiantes en un mínimo durante la parte temprana de los cursos introductorios de estadística, después gradualmente introducimos temas como sean necesarios. Para cursos donde los estudiantes llevan a cabo proyectos sustantivos, el manejo de datos es más importante. Este capítulo describe algunas tareas clave en el manejo de datos.

13.1 Inspección de dataframes

La función `inspect()` puede ser útil describiendo las variables del dataframe (el nombre para un conjunto de datos en R).

```
> inspect(iris)
```

categorical variables:

	name	class	levels	n	missing
1	Species	factor	3	150	0

distribution

1 setosa (33.3%), versicolor (33.3%) ...

quantitative variables:

	name	class	min	Q1	median	Q3	max	mean	sd	n
1	Sepal.Length	numeric	4.3	5.1	5.80	6.4	7.9	5.843	0.8281	150
2	Sepal.Width	numeric	2.0	2.8	3.00	3.3	4.4	3.057	0.4359	150
3	Petal.Length	numeric	1.0	1.6	4.35	5.1	6.9	3.758	1.7653	150
4	Petal.Width	numeric	0.1	0.3	1.30	1.8	2.5	1.199	0.7622	150

missing

1 0

El libro *Empiece a enseñar con R* caracteriza una extensiva sección en manejo de datos, incluyendo el uso de la función `read.file()` para carga datos en R y RStudio

nombre para un conjunto de datos en R). Los paquetes `dplyr` y `tidyr` brindan una elegante herramienta para el manejo de datos y facilitan la habilidad de los estudiantes a computar con los datos. Hadley Wickham, autor de los paquetes, sugiere que hay seis expresiones (o verbos) implementados dentro de estos paquetes que permiten una larga lista de tareas sean realizadas. Filtrar (mantener filas con criterios de emparejamiento), seleccionar (elegir columnas por nombre), ordenar (reordenar filas), mutar (agregar nuevas variables), resumir (reducir las variables a valores) y agrupar por (colapsar grupos). Vea <http://www.amherst.edu/nhorner/predators> para más detalles

```
2      0
3      0
4      0
```

El dataframe `iris` incluye una variable categórica y cuatro variables cuantitativas

13.2 *Agregar nuevas variables a un dataframe*

Podemos agregar variables adicionales a un dataframe existente usando `mutate()`. Pero primero debemos crear una versión más pequeña del conjunto de datos `iris`.

```
> irisSmall <- select(iris, Species, Sepal.Length)

> # cortar pedazos de los datos en cajas
> irisSmall <- mutate(irisSmall,
  Length = cut(Sepal.Length, breaks=4:8))
```

Comandos múltiples pueden ser encadenados usando el operador `%>%` (pipa):

```
> irisSmall <- iris %>%
  select(Species, Sepal.Length) %>%
  mutate(Length = cut(Sepal.Length, breaks=4:8))
```

Note que en este uso el primer argumento de `select()` es la primera variable (como lo hereda de los datos de la pipa anterior).

```
> head(irisSmall)

  Species Sepal.Length Length
1  setosa         5.1   (5,6]
2  setosa         4.9   (4,5]
3  setosa         4.7   (4,5]
4  setosa         4.6   (4,5]
5  setosa         5.0   (4,5]
6  setosa         5.4   (5,6]
```

La función `cut()` tiene la opción de etiqueta que puede ser utilizada para especificar más nombres para describir los grupos

El conjunto de datos `CPS85` contiene datos de una Encuesta Actual de Población (Actual en 1985, eso es). Dos de las variables en este conjunto de datos son edad y educ.

Podemos estimar el número de años que el trabajador ha estado en la fuerza de trabajo desde que completó la educación, asumiendo que ha estado en la fuerza de trabajo desde que completó su educación y que su edad en la graduación es 6 años más que el número de años de educación obtenida. Podemos agregar esto como una nueva variable en el conjunto de datos usando `mutate()`

```
> CPS85 <- mutate(CPS85, workforce.years = age - 6 - educ)
> favstats(~ workforce.years, data=CPS85)
```

```
min Q1 median Q3 max mean sd n missing
-4 8 15 26 55 17.81 12.39 534 0
```

De hecho, esto es lo que se hizo para todos excepto uno de los casos para crear la variable `exper` que está ya en los datos de CPS85.

```
> tally(~ (exper - workforce.years), data=CPS85)

(exper - workforce.years)
0 4
533 1
```

13.3 Desechando variables

Puesto que ya tenemos la variable `exper`, no hay una verdadera razón para dejarnos nuestra nueva variable. Desechémosla. Note una utilización inteligente del signo de menos

```
> names(CPS85)

[1] "wage" "educ" "race"
[4] "sex" "hispanic" "south"
[7] "married" "exper" "union"
[10] "age" "sector" "workforce.years"

> CPS1 <- select(CPS85, select = -matches("workforce.years"))
> names(CPS1)

[1] "wage" "educ" "race" "sex" "hispanic" "south"
[7] "married" "exper" "union" "age" "sector"
```

Cualquier número de variables puede ser descartado o mantenido de una manera similar.

```
> CPS1 <- select(CPS85, select = -matches("workforce.years|exper"))
```

13.4 Renombrando variables

La columna de nombres (variables) de un conjunto de datos puede ser cambiada usando la función `rename()` en el paquete `dplyr`.

```
> names(CPS85)

[1] "wage"          "educ"          "race"
[4] "sex"           "hispanic"      "south"
[7] "married"       "exper"         "union"
[10] "age"           "sector"        "workforce.years"

> CPSnew = rename(CPS85, workforce=workforce.years)
> names(CPSnew)

[1] "wage"          "educ"          "race"          "sex"           "hispanic"
[6] "south"         "married"       "exper"         "union"         "age"
[11] "sector"        "workforce"
```

Los nombres de las filas pueden ser cambiados con una indicación simple, usando `row.names()`.

El conjunto de datos `faithful` (en el paquete `datasets`, que está siempre disponible) tiene nombres muy poco desafortunados.

```
> names(faithful)

[1] "eruptions" "waiting"
```

Es una buena idea establecer prácticas de elección de nombres de variables desde el primer día.

Las medidas son la duración de una erupción y el tiempo hasta la erupción subsecuente, entonces, demos a estar unos mejores nombres.

```
> faithful <- rename(faithful,
  duration = eruptions,
  time.til.next=waiting)
> names(faithful)

[1] "duration"      "time.til.next"

> xyplot(time.til.next ~ duration, alpha=0.5, data=faithful)
```

Si la variable que contiene el conjunto de datos es modificada o utilizada para almacenar un objeto diferente, los datos originales del paquete pueden ser recuperados usando `data()`.

```
> data(faithful)
> head(faithful, 3)

  eruptions waiting
1     3.600      79
2     1.800      54
3     3.333      74
```

13.5 Creando subconjuntos de datos

Podemos también usar `filter()` para reducir el tamaño del conjunto de datos seleccionando sólo algunas filas.

```
> data(faithful)
> names(faithful) <- c('duration', 'time.til.next')
> # any logical can be used to create subsets
> faithfulLong <- filter(faithful, duration > 3)
> xyplot( time.til.next ~ duration, data=faithfulLong )
```

13.6 Ordenar dataframes

Los conjuntos de datos pueden ser ordenados usando la función `arrange()`.

```
> head(faithful, 3)

  duration time.til.next
1     3.600          79
2     1.800          54
3     3.333          74

> sorted <- arrange(faithful, duration)
> head(sorted, 3)

  duration time.til.next
1     1.600          52
2     1.667          64
3     1.700          59
```

PRECAUCIÓN!
usualmente es mejor hacer conjuntos de datos nuevos en lugar de modificar el original

13.7 Uniendo conjuntos de datos

El conjunto de datos `fusion1` en el paquete `fastR` contiene información del genotipo para la SNP (Polimorfismo

de nucleotido simple) en el gen TCF7L2. El conjunto de datos pheno contiene fenotipos (incluyendo diabetes tipo 2 en estado caso/control) para un grupo de individuos con intersecciones. Podemos unir estos y explorar la asociación entre genotipos y fenotipos usando `merge()`.

```
> require(fastR)
> require(dplyr)
> fusion1 <- arrange(fusion1, id)
> head(fusion1, 3)
```

	id	marker	markerID	allele1	allele2	genotype	Adose	Cdose	Gdose	Tdose
1	1002	RS12255372	1	3	3	GG	0	0	2	0
2	1009	RS12255372	1	3	3	GG	0	0	2	0
3	1012	RS12255372	1	3	3	GG	0	0	2	0

```
> head(pheno, 3)
```

	id	t2d	bmi	sex	age	smoker	chol	waist	weight	height	whr	sbp	dbp
1	1002	case	32.86	F	70.76	former	4.57	112.0	85.6	161.4	0.9868	135	77
2	1009	case	27.39	F	53.92	never	7.32	93.5	77.4	168.1	0.9397	158	88
3	1012	control	30.47	M	53.86	former	5.02	104.0	94.6	176.2	0.9327	143	89

```
> require(tidyr)
> fusion1m <- inner_join(fusion1, pheno, by='id')
> head(fusion1m, 3)
```

	id	marker	markerID	allele1	allele2	genotype	Adose	Cdose	Gdose	Tdose	t2d	bmi
1	1002	RS12255372	1	3	3	GG	0	0	2	0	case	32.8
2	1009	RS12255372	1	3	3	GG	0	0	2	0	case	27.3
3	1012	RS12255372	1	3	3	GG	0	0	2	0	control	30.4

	sex	age	smoker	chol	waist	weight	height	whr	sbp	dbp
1	F	70.76	former	4.57	112.0	85.6	161.4	0.9868	135	77
2	F	53.92	never	7.32	93.5	77.4	168.1	0.9397	158	88
3	M	53.86	former	5.02	104.0	94.6	176.2	0.9327	143	89

Ahora estamos listos para iniciar nuestro análisis

```
> tally(~t2d + genotype, data=fusion1m)
```

	genotype		
t2d	GG	GT	TT
case	737	375	48
control	835	309	27

13.8 Tajando y cortando

The `tidyr` package provides a flexible way to change the arrangement of data. El paquete `tidyr` provee una forma flexible para cambiar el ordenamiento de los datos. Fue diseñado para convertir entre largas y amplias versiones de series de datos de series de tiempo y sus argumentos son nombrados con eso en mente.

Una situación común es cuando queremos convertir de una forma amplia a una forma larga por cambios en la perspectiva de lo que la unidad de observación es. Por ejemplo en los datos `traffic`, cada fila es un año y los se da en cada una los datos para diferentes estados.

Las viñetas que acompañan a los paquetes `tidyr` y `dplyr` se caracterizan por tener una cantidad de ejemplos útiles de manipulación de datos común

```
> traffic
```

	year	cn.deaths	ny	cn	ma	ri
1	1951	265	13.9	13.0	10.2	8.0
2	1952	230	13.8	10.8	10.0	8.5
3	1953	275	14.4	12.8	11.0	8.5
4	1954	240	13.0	10.8	10.5	7.5
5	1955	325	13.5	14.0	11.8	10.0
6	1956	280	13.4	12.1	11.0	8.2
7	1957	273	13.3	11.9	10.2	9.4
8	1958	248	13.0	10.1	11.8	8.6
9	1959	245	12.9	10.0	11.0	9.0

Podemos darle un nuevo formato a esto de tal forma que cada fila contenga una medida para un solo estado en un solo año

```
> longTraffic <- traffic %>%
  gather(state, deathRate, ny:ri)
> head(longTraffic)
```

	year	cn.deaths	state	deathRate
1	1951	265	ny	13.9
2	1952	230	ny	13.8
3	1953	275	ny	14.4
4	1954	240	ny	13.0
5	1955	325	ny	13.5
6	1956	280	ny	13.4

También podemos darle formato de otra forma, en esta ocasión, teniendo todos los datos para un estado en una fila del conjunto de datos.

```

> stateTraffic <- longTraffic %>%
  select(year, deathRate, state) %>%
  mutate(year=paste("deathRate.", year, sep="")) %>%
  spread(year, deathRate)
> stateTraffic

```

	state	deathRate.1951	deathRate.1952	deathRate.1953	deathRate.1954	deathRate.1955
1	cn	13.0	10.8	12.8	10.8	14.0
2	ma	10.2	10.0	11.0	10.5	11.8
3	ny	13.9	13.8	14.4	13.0	13.5
4	ri	8.0	8.5	8.5	7.5	10.0

	deathRate.1956	deathRate.1957	deathRate.1958	deathRate.1959
1	12.1	11.9	10.1	10.0
2	11.0	10.2	11.8	11.0
3	13.4	13.3	13.0	12.9
4	8.2	9.4	8.6	9.0

13.9 Creación de variables derivadas

Existen varias funciones que ayuda a facilitar la creación o recodificación de variables

13.9.1 Creando variables categóricas de una variable cuantitativa

De forma siguiente vamos a mostrar como crear una variable categórica de 3 niveles con cortes en 20 y 40 para la escala del CESD (que tiene un rango de 0 a 60)

```

> favstats(~ cesd, data=HELPrct)

min Q1 median Q3 max  mean    sd    n missing
1 25     34 41  60 32.85 12.51 453      0

> HELPrct <- mutate(HELPrct, cesdcut = cut(cesd,
  breaks=c(0, 20, 40, 60), include.lowest=TRUE))
> bwplot(cesd ~ cesdcut, data=HELPrct)

```

Puede ser preferible darle mejor nombre a las etiquetas.

```

> HELPrct <- mutate(HELPrct, cesdcut = cut(cesd,
  labels=c("low", "medium", "high"),
  breaks=c(0, 20, 40, 60), include.lowest=TRUE))
> bwplot(cesd ~ cesdcut, pch="|", data=HELPrct)

```

La función `ntiles()` puede automatizar la creación de grupos en este aspecto

La función `derivedFactor()` es incluso más general y puede ser utilizada para este propósito.

```
> HELPrct <- mutate(HELPrct,
  anothercut = derivedFactor(
    low = cesd >= 0 & cesd <= 20,
    medium = cesd > 20 & cesd <= 40,
    high = cesd > 40))
```

13.9.2 Reordenando factores

Como predeterminado R utiliza el primer nivel en el orden lexicográfico como grupo de referencia para el modelaje. Esto puede ser anulado usando la función `relevel()` (ver también `reorder()`).

```
> tally(~ substance, data=HELPrct)

substance
alcohol cocaine  heroin
    177      152      124

> coef(lm(cesd ~ substance, data=HELPrct))

(Intercept) substancecocaine substanceheroin
    34.3729         -4.9518         0.4981

> HELPrct <- mutate(HELPrct, subnew = relevel(substance,
  ref="heroin"))
> coef(lm(cesd ~ subnew, data=HELPrct))

(Intercept) subnewalcohol subnewcocaine
    34.8710         -0.4981        -5.4499
```

13.10 Estadísticas agrupadas

A veces puede ser útil calcular estadísticas de resumen por grupo, y agregar estas en un conjunto de datos. La función `group_by()` en el paquete `dplyr` facilita este proceso. Aquí demostramos cómo agregar una variable que contiene la media de la edad de los sujetos por grupo de sustancia.

```
> favstats(age ~ substance, data=HELPrct)
```

```

      substance min Q1 median    Q3 max  mean    sd    n missing
1   alcohol  20 33   38.0 43.00  58 38.20 7.652 177         0
2   cocaine  23 30   33.5 37.25  60 34.49 6.693 152         0
3   heroin   19 27   33.0 39.00  55 33.44 7.986 124         0

> ageGroup <- HELPrct %>%
  group_by(substance) %>%
  summarise(agebygroup = mean(age))
> ageGroup

# A tibble: 3 x 2
  substance agebygroup
  <fctr>      <dbl>
1   alcohol      38.20
2   cocaine      34.49
3   heroin       33.44

> nrow(ageGroup)

[1] 3

> nrow(HELPrct)

[1] 453

> HELPmerged <- left_join(ageGroup, HELPrct, by="substance")
> favstats(agebygroup ~ substance, data=HELPmerged)

      substance  min    Q1 median    Q3  max  mean sd    n missing
1   alcohol 38.20 38.20  38.20 38.20 38.20 38.20 0 177         0
2   cocaine 34.49 34.49  34.49 34.49 34.49 34.49 0 152         0
3   heroin  33.44 33.44  33.44 33.44 33.44 33.44 0 124         0

> nrow(HELPmerged)

[1] 453

```

13.11 Contabilizando datos perdidos

Los valores perdidos salen a flote en casi todas las investigaciones en el mundo real. Rutiliza el símbolo NA como un indicador de datos faltantes. El conjunto de datos `HELPmiss` dentro del paquete `mosaicData` incluye todos los $n=470$ sujetos ligados a la investigación de base (incluyendo el $n=17$ de sujetos con algunos datos faltantes que no fueron incluidos en `HELPrct`).

```

> smaller <- select(HELPmiss, cesd, drugrisk, indtot, mcs, pcs,
  substance)
> dim(smaller)

[1] 470    6

> summary(smaller)

      cesd      drugrisk      indtot      mcs      pcs
Min.   : 1.0   Min.   : 0.00   Min.   : 4.0   Min.   : 6.76   Min.   :14.1
1st Qu.:25.0   1st Qu.: 0.00   1st Qu.:32.0   1st Qu.:21.66   1st Qu.:40.4
Median :34.0   Median : 0.00   Median :37.5   Median :28.56   Median :48.9
Mean   :32.9   Mean   : 1.87   Mean   :35.7   Mean   :31.55   Mean   :48.1
3rd Qu.:41.0   3rd Qu.: 1.00   3rd Qu.:41.0   3rd Qu.:40.64   3rd Qu.:57.0
Max.   :60.0   Max.   :21.00   Max.   :45.0   Max.   :62.18   Max.   :74.8
      NA's      :2      NA's      :14      NA's      :2      NA's      :2

  substance
alcohol:185
cocaine:156
heroin :128
missing:  1

```

De los 470 sujetos en las 6 variables del conjunto de datos, solamente drugrisk, indtot, mcs y pcs tienen valores faltantes

```

> favstats(~ mcs, data=smaller)

  min   Q1 median   Q3   max  mean   sd  n missing
6.763 21.66  28.56 40.64 62.18 31.55 12.78 468      2

> with(smaller, sum(is.na(mcs)))

[1] 2

> nomiss <- na.omit(smaller)
> dim(nomiss)

[1] 453    6

> nrow(nomiss)

[1] 453

> ncol(nomiss)

[1] 6

```

```
> favstats(~ mcs, data=nomiss)

      min      Q1 median      Q3      max mean      sd      n missing
6.763 21.79   28.6 40.94 62.18 31.7 12.82 453          0
```

Alternativamente, podemos generar el mismo conjunto de datos usando condiciones lógicas.

```
> nomiss <- filter(smaller,
  (!is.na(mcs) & !is.na(indtot) & !is.na(drugrisk)))
> dim(nomiss)

[1] 453   6
```

Health Evaluation (HELP) Study

Muchos de los ejemplos en esta guía utilizaron datos del estudio HELP, una prueba clínica aleatorizada para pacientes en edad adulta, hospitalizados, reclutados de una unidad de desintoxicación. Los pacientes sin médico de cuidados primarios fueron aleatorizados para recibir evaluación multidisciplinaria y una pequeña intervención motivacional o cuidado usual, con la meta de unirlos al cuidado médico primario. Los fondos para el estudio HELP fueron previstos por el Instituto Nacional sobre Abuso de Alcohol y Alcoholicismo (R01-AA10870, Samet PI) y el Instituto Nacional sobre Abuso de Droga (R01-DA10019, Samet PI). Los detalles de la prueba aleatorizada así como los resultados de análisis adicionales han sido publicados¹.

Los sujetos elegibles eran adultos, que hablaran inglés o español, que reportaran el alcohol, la heroína o la cocaína como su droga de preferencia, que residieran en la proximidad de la clínica de cuidado primario a la cual sería referido, o si fuera el caso que no tuviera casa. Pacientes que tuvieran relaciones establecidas con el cuidado primario que planeaban continuar en estos programas, demencia significativa, planes específicos de dejar el área de Boston lo cual obstruiría la participación en la investigación, fallo en proveer información de contacto para rastreo o embarazo fueron excluidos.

Los sujetos fueron entrevistados en una línea de base durante su estadía en el programa de desintoxicación y se realizaron entrevistas de seguimiento cada 6 meses por 2 años. Variables continuas, conteos, variables discretas, predictores de tiempo de sobrevivencia y otros resultados fueron recolectados en cada una de estas cinco ocasiones. La Comisión Revisora Institucional del Centro Médico de la Universidad

¹ J. H. Samet, M. J. Larson, N. J. Horton, K. Doyle, M. Winter, and R. Saitz. Linking alcohol and drug dependent adults to primary medical care: A randomized controlled trial of a multidisciplinary health intervention in a detoxification unit. *Addiction*, 98(4):509–516, 2003; J. Liebschutz, J. B. Savetsky, R. Saitz, N. J. Horton, C. Lloyd-Travaglini, and J. H. Samet. The relationship between sexual and physical abuse and substance abuse consequences. *Journal of Substance Abuse Treatment*, 22(3):121–128, 2002; and S. G. Kertesz, N. J. Horton, P. D. Friedmann, R. Saitz, and J. H. Samet. Slowing the revolving door: stabilization programs reduce homeless persons' substance use after detoxification. *Journal of Substance Abuse Treatment*, 24(3):197–207, 2003

de Boston aprobó todos los aspectos del estudio, incluyendo la creación de conjuntos de datos sin identificación. Medidas de protección adicional fue asegurada por emisión de un Certificado de Confidencialidad del Departamento de Salud y Servicios Humanos.

El paquete **mosaicData** contiene algunas formas del conjunto de datos **HELP** sin identificación. Nos vamos a concentrar en **HELPrct** que contiene 27 variables de 453 sujetos con un mínimo de datos faltantes. Las variables incluidas en el conjunto de datos de **HELP** son descritas en la Tabla 14.1. Más información puede ser encontrada en: <http://www.amherst.edu/~nhorton/r2>. Una copia de los instrumentos de estudio puede ser encontrada en: <http://www.amherst.edu/~nhorton/help>

Cuadro 14.1: Descripción de las variables en el Conjunto de datos **HELPrct**

Variable	Descripción(valores)	Nota
age	Edad de referencia(en años) (rango 19–60)	
anysub	Uso de cualquier sustancia post desintoxicación	vea también day-sanysub
cesd	Escala del Centro de Estudios Epidemiológicos en Depresión (rango de 0–60, puntajes mayores indican mayores síntomas depresivos)	
d1	Cuántas veces ha sido hospitalizado por problemas médicos(en toda la vida) (rango 0-100)	
daysanysub	Tiempo (en días) después del primer uso de cualquier sustancia post-desintoxicación (rango 0–268)	vea también any-substatus
dayslink	Tiempo (en días) de asociación al cuidado primario(rango 0–456)	vea también linkstatus
drugrisk	Puntaje sobre riesgo a drogas del Risk-Assessment Battery (RAB) (rango 0–21)	vea también sex-risk
e2b	Número de veces en los últimos 6 meses que ha entrado a un programa de desintoxicación (rango 1–21)	

female	Género del entrevistado (0=hombre, 1=mujer)	
g1b	Experimentó pensamientos serios de suicidio (últimos 30 días, valores 0=no, 1=sí)	
homeless	Ha dormido 1 o más noches en la calle o un albergue en los últimos 6 meses(0=no, 1=sí)	
i1	Número promedio de bebidas (en unidades estándar, consumidas por día(en los últimos 30 días, rango 0-184)	vea también i2
i2	Máximo número de bebidas (en unidades estándar) consumidas por día (en los últimos 30 días, rango 0-184)	vea también i1
id	Identificador del sujeto (1-470)	
indtot	Puntaje total en el Inventario de Consecuencias del uso de drogas (rango 4-45)	
linkstatus	Asociación post-desintoxicación al cuidado primario (0=no, 1=sí)	vea también days-link
mcs	Puntaje del Componente Mental SF-36 (rango 14-75, puntajes altos son preferibles)	vea también pcs
pcs	Puntaje del Componente Físico SF-36(rango 14-75, puntajes altos son preferibles)	see also mcs
pss_fr	Apoyo social percibido (amigos, rango 0-14)	
racegrp	Raza/etnia (negro, blanco, hispano u otro)	
satreat	Cualquier tratamiento de abuso de sustancias de BSAS al inicio (0=no, 1=sí)	
sex	Sexo del entrevistado (hombre o mujer)	
sexrisk	Risk-Assessment Battery (RAB) sex risk score (range 0-21)	ver también dru-grisk
substance	Principal sustancia de abuso (alcohol, cocaína, heroína)	

treat	Aleatorización del grupo(Aleatorizado a una clínica HELP, no o sí)	
--------------	--	--

Notas: El rango se provee para las variables continuas.

15

Ejercicios y problemas

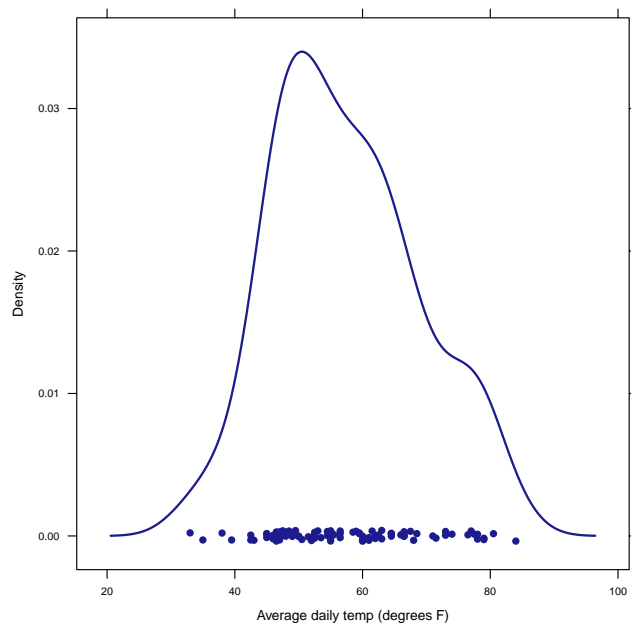
La primera parte del número del ejercicio dice de qué capítulo viene.

10.1. El conjunto de datos `RailTrail` dentro del paquete `mosaic`, incluye un conteo de cruces de un camino en Northampton, Massachusetts para 90 días en 2005. Las autoridades de la ciudad están interesadas en entender el uso de la red de caminos, y como cambia en función de la temperatura y el día de la semana. Describa la distribución de la variable `avgtemp` en terminos de su centro, dispersión y forma.

```
> favstats(~ avgtemp, data=RailTrail)
```

min	Q1	median	Q3	max	mean	sd	n	missing
33	48.62	55.25	64.5	84	57.43	11.33	90	0

```
> densityplot(~ avgtemp, xlab="Average daily temp (degrees F)",  
  data=RailTrail)
```



10.2. El conjunto de datos de `RailTrail` también incluye una variable llamada `cloudcover`. Describa la distribución de esta variable de la misma forma anterior.

10.3. El conjunto de datos `RailTrail` brinda datos sobre el conteo diario de cruces, llamado `volume`. Describa la distribución de la variable de la misma forma anterior..

10.4. El conjunto de datos `RailTrail` también contiene un indicador de si el día fue un día entre semana (`weekday==1`), o un fin de semana/día libre (`weekday==0`). Utilice `tally()` para describir la distribución de esta variable categorica. ¿Qué porcentaje de días son fines de semana/días libres?

10.5. Use diagramas de caja que se encuentren uno al lado del otro para comparar la distribución de `volume` por el tipo de día en el conjunto de datos `RailTrail`. Pista: va necesitar transformar la variable numéricos en un factor variable usando `as.factor()` o usar la opción `horizontal=FALSE`. ¿Qué concluye?

10.6. Use gráficos de densidad traslapados para comprar la distribución de `volume` por tipo de día en el conjunto de datos `RailTrail`. ¿Qué concluye?

10.7. Cree un gráfico de dispersión de `volume` en función de `avgtemp` usando el conjunto de datos de `RailTrail`,

junto con la línea de regresión y un suavizador de gráfico de dispersión (curva de regresión local). ¿Qué observa en esta relación?

10.8. Usando el conjunto de datos de `RailTrail`, ajuste un modelo de regresión múltiple para `volume` como función de `cloudcover`, `avgtemp`, `weekday` y la interacción entre el tipo de día y la temperatura promedio. ¿Hay evidencia para mantener la interacción con un nivel de significancia de $\alpha = 0,05$ `avgtemp`, `weekday` and the interaction

10.9. Use la función `makeFun()` para calcular el número predicho de cruces en un día de la semana con una temperatura promedio de 60 grados y sin nubes. Verifique este cálculo usando los coeficientes del modelo.

```
> fm <- lm(volume~cloudcover+avgtemp+weekday+weekday:avgtemp, data=RailTrail)
> coef(fm)
```

(Intercept)	cloudcover	avgtemp	weekday1
378.834	-17.198	2.313	-321.116
avgtemp:weekday1			
4.727			

10.10. Use `makeFun()` y `plotFun()` para desplegar los valores predichos para el número de cruces en días de la semana y en fines de semana/días libres para temperaturas promedio entre 30 y 80 grados en días nublados (`cloudcover=10`)

10.11. Usando un modelo de regresión múltiple, genere un histograma (con una curva de densidad normal sobre él) para evaluar la normalidad de los residuales.

10.12. Usando un modelo de regresión múltiple, genere un histograma (con una curva de densidad normal sobre él) para evaluar la normalidad de los residuales.

10.13. Usando el mismo modelo, genere un gráfico de dispersión de los residuales contra los valores predichos y comente la linealidad del modelo y el supuesto de igualdad de varianzas.

11.1. Genere una muestra de 1000 variables aleatorias exponenciales con el parámetro igual a 2, calcule la media de estas variables.

11.2. Encuentre la mediana de la variable aleatoria X , si esta se distribuye exponencialmente con un parámetro igual a 10.

12.1. Encuentre la potencia de una prueba t de dos colas con dos muestras donde ambas distribuciones son aproximadamente distribuidas normalmente con la misma desviación estándar, pero el grupo difiere en un 50 % de la desviación estándar. Asuma que hay 25 observaciones por grupo y un nivel de significancia de 0.0538760721683502

12.2. Encuentre el tamaño de muestra necesario para tener un 90 % de potencia para una prueba t de dos grupos donde la diferencia entre las medias es el 25 % de la desviación estándar en los grupos (con un α de 0.05).

13.1. Usando el conjunto de datos `faithful`, haga un gráfico de dispersión de la duración de la erupción contra el tiempo desde la erupción pasada.

13.2. En el conjunto de datos `fusion2` en el paquete `fastR` contiene genotipos para otro SNP. Una `fusion1`, `fusion2` y `pheno` en un solo conjunto de datos.

Note que `fusion1`, `fusion2` tienen las mismas columnas.

```
> names(fusion1)
```

```
[1] "id"      "marker"  "markerID" "allele1"  "allele2"  "genotype" "Adose"
[8] "Cdose"   "Gdose"   "Tdose"
```

```
> names(fusion2)
```

```
[1] "id"      "marker"  "markerID" "allele1"  "allele2"  "genotype" "Adose"
[8] "Cdose"   "Gdose"   "Tdose"
```

Usted tal vez quiere usar argumentos `suffixes` a `merge()` o renombrar las variables después de que las haya terminado de unirlos para hacer el conjunto de datos resultante más fácil de navegar. Ordene su conjunto de datos eliminando columnas que son redundantes o que no desea tener en su conjunto de datos final.

Bibliografía

- [BcRB⁺14] B.S. Baumer, M. Çetinkaya Rundel, A. Bray, L. Loi, and N. J. Horton. R Markdown: Integrating a reproducible analysis tool into introductory statistics. *Technology Innovations in Statistics Education*, 8(1):281–283, 2014.
- [HBW15] N.J. Horton, B.S. Baumer, and H. Wickham. Setting the stage for data science: integration of data management skills in introductory and second courses in statistics (<http://arxiv.org/abs/1401.3269>). *CHANCE*, 28(2):40–50, 2015.
- [KHF⁺03] S. G. Kertesz, N. J. Horton, P. D. Friedmann, R. Saitz, and J. H. Samet. Slowing the revolving door: stabilization programs reduce homeless persons’ substance use after detoxification. *Journal of Substance Abuse Treatment*, 24(3):197–207, 2003.
- [LSS⁺02] J. Liebschutz, J. B. Savetsky, R. Saitz, N. J. Horton, C. Lloyd-Travaglini, and J. H. Samet. The relationship between sexual and physical abuse and substance abuse consequences. *Journal of Substance Abuse Treatment*, 22(3):121–128, 2002.
- [MM07] D. S. Moore and G. P. McCabe. *Introduction to the Practice of Statistics*. W.H. Freeman and Company, 6th edition, 2007.
- [NT10] D. Nolan and D. Temple Lang. Computing

- in the statistics curriculum. *The American Statistician*, 64(2):97–107, 2010.
- [RS02] F. Ramsey and D. Schafer. *Statistical Sleuth: A Course in Methods of Data Analysis*. Cengage, 2nd edition, 2002.
- [SLH⁺03] J. H. Samet, M. J. Larson, N. J. Horton, K. Doyle, M. Winter, and R. Saitz. Linking alcohol and drug dependent adults to primary medical care: A randomized controlled trial of a multidisciplinary health intervention in a detoxification unit. *Addiction*, 98(4):509–516, 2003.
- [Tuf01] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 2nd edition, 2001.
- [Wor14] ASA Undergraduate Guidelines Workgroup. 2014 curriculum guidelines for undergraduate programs in statistical science. Technical report, American Statistical Association, November 2014. <http://www.amstat.org/education/curriculumguidelines.cfm>.

Índice alfabético

- `%>%`, 100
- `abs()`, 83, 84
- `add` option, 82, 92
- adding variables, 100
- `all.x` option, 104
- `alpha` option, 50, 62
- analysis of variance, 66
- `anova()`, 81
- `anova()`, 67, 72
- `aov()`, 67, 79
- `arrange()`, 103, 104
- `auto.key` option, 80, 82
-
- `binom.test()`, 39
- binomial test, 39
- bootstrapping, 36
- `breaks` option, 106
- `bwplot()`, 61, 66, 79
- `by.x` option, 104
-
- categorical variables, 39
- `center` option, 92
- `cex` option, 45, 62, 84
- `chisq.test()`, 42, 58
- `class()`, 49
- `coef()`, 48, 107
- coefficient plots, 83
- `col` option, 33
- `conf.int` option, 75
- `confint()`, 35, 40, 48
- contingency tables, 39, 55
-
- Cook's distance, 52
- `cor()`, 47
- `correct` option, 41
- correlation, 47
- Cox proportional hazards model, 76
- `coxph()`, 76
- CPS85 dataset, 100, 101
- creating subsets, 103
- cross classification tables, 55
- `CrossTable()`, 56
- `cut()`, 100, 106
-
- data management, 99
- `data()`, 102
- dataframe, 100
- dataframes
 - inspecting, 99
 - merging, 103
 - reshaping, 105
 - sorting, 103
 - subsetting, 103
- `density` option, 49
- `densityplot()`, 33
- derived variables, 106
- `derivedFactor()`, 107
- `diffmean()`, 65
- `dim()`, 108
- display first few rows, 100
- `dnorm()`, 88
- `do()`, 36
- `dotPlot()`, 32
-
- `dplyr` package, 29, 30
- dropping variables, 101
-
- `exp()`, 71
-
- factor reordering, 107
- `factor()`, 68, 79
- failure time analysis, 75
- faithful dataset, 102
- `family` option, 71
- `favstats()`, 28, 61, 107, 109
- `filter()`, 29, 101
- Fisher's exact test, 59
- `fisher.test()`, 59
- `fit` option, 30
- `fitted()`, 83
- `format` option, 29
- `freqpolygon()`, 34
- `function()`, 46
- fusion1 dataset, 104
-
- `gather()`, 105
- `glm()`, 71
- `grid.text()`, 33
- group-wise statistics, 107
- `group_by()`, 107
- `groups` option, 65, 82, 91
-
- `head()`, 100
- Health Evaluation and Linkage to Primary Care study, 111
- HELP study, 111

- HELPmiss dataset, 108
- HELPrct dataset, 27
- histogram(), 29, 92
- honest significant differences, 68
- include.lowest option, 106
- incomplete data, 108
- inspect(), 99
- inspecting dataframes, 99
- install.packages(), 14
- installing packages, 14
- integrate(), 88
- interactions, 81
- iris dataset, 100
- is.na(), 109
- Kaplan-Meier plot, 75
- knitr, 14
- labels option, 68
- ladd(), 33
- layout option, 31
- left_join(), 107
- levels option, 68
- leverage, 52
- linear regression, 48
- linearity, 45
- lm(), 48, 68
- loading packages, 14
- logistic regression, 71
- lowess, 45
- lty option, 33
- lwd option, 33, 45, 91
- makeFun(), 82, 92
- margins option, 39
- markdown, 14
- mean(), 27, 61
- median(), 28, 79
- merging dataframes, 103
- missing data, 108
- model comparison, 68
- Modeling with R*, 27
- mosaic package, 27
- mosaicplot(), 56
- mplot(), 50, 69, 83
- msummary(), 80
- msummary(), 48, 64, 71
- multiple comparisons, 68
- multiple regression, 80
- multivariate relationships, 80
- mutate(), 68, 79, 100, 101, 106, 107
- NA character, 108
- na.omit(), 109
- names(), 102
- ncol(), 109
- nint option, 31
- nrow(), 107, 109
- ntiles(), 106
- oddsRatio(), 56
- one-way ANOVA, 66
- options(), 27
- pairs plot, 48
- panel.abline(), 33
- panel.labels(), 46
- panel.mathdensity(), 33
- panel.text(), 46
- paste(), 106
- pbinom(), 96
- pch option, 45, 91
- pchisq(), 43
- Pearson correlation, 47
- permutation test, 64
- pipe operator, 100
- plotDist(), 43, 90
- plotFun(), 82, 92
- pnorm(), 88
- polygons, 34
- print(), 40
- prop.test(), 41
- proportional hazards model, 76
- pval(), 40
- qdata(), 36
- qnorm(), 88
- qqmath(), 49
- quantile(), 28
- quantiles, 28
- random variables, 87
- read.file(), 99
- regression, 48
- regression diagnostics, 83
- relevel(), 107
- rename(), 102
- renaming variables, 102
- reordering factors, 107
- reproducible analysis, 14
- require(), 14, 27
- resample(), 36
- resampling, 36, 64
- reshaping dataframes, 105
- resid(), 83
- residual diagnostics, 83
- residuals(), 49
- rexp(), 92
- rnorm(), 88
- row.names(), 102
- rownames(), 46
- rsquared(), 48
- RStudio.Version(), 19
- scale versus location, 51
- scatterplot matrix, 48
- scatterplots, 45
- sd(), 28
- select option, 101
- select(), 100, 106–108
- sessionInfo(), 19
- shuffle(), 65
- significance stars, 48
- smoothers, 45
- sorting dataframes, 103
- Spearman correlation, 47
- splom(), 48
- spread(), 106
- Start Modeling with R*, 27
- Start Teaching with R*, 27
- stem(), 29
- subsetting dataframes, 103

<code>sum()</code> , 43, 109	<code>tidyr</code> package, 30, 104	<code>which</code> option, 50
<code>summarise()</code> , 107	time to event analysis, 75	<code>width</code> option, 32, 92
<code>summary()</code> , 64	transforming dataframes, 105	<code>wilcox.test()</code> , 64
<code>Surv()</code> , 75	transposing dataframes, 105	<code>with()</code> , 27, 109
<code>survfit()</code> , 75	Tukey's HSD, 68	
survival analysis, 75	<code>TukeyHSD()</code> , 68	<code>xchisq.test()</code> , 43, 58
	<code>type</code> option, 45, 84, 91	<code>xlab</code> option, 75
<code>t.test()</code> , 35, 63		<code>xlim</code> option, 65
tables, 39, 55	<code>under</code> option, 91	<code>xpnorm()</code> , 88
<code>tally()</code> , 29, 39, 55, 107		<code>xyplot()</code> , 45, 62, 82
<i>Teaching with R</i> , 27	<code>var()</code> , 28	
<code>test</code> option, 72	<code>var.equal</code> option, 63	<code>ylab</code> option, 82
thinking with data, 99	vignettes, 13	